Multibiometric Template Security Using Fuzzy Vault

Karthik Nandakumar and Anil K. Jain

Abstract— **Template security is a critical issue in biometric systems because biometric templates cannot be easily revoked and reissued. While multibiometric systems overcome limitations such as non-universality and high error rates that affect unibiometric systems, they require storage of multiple templates for the same user. Securing the different templates of a user separately is not optimal in terms of security. Hence, we propose a scheme for securing multiple templates of a user as a single entity. We derive a single multibiometric template from the individual templates and secure it using the fuzzy vault framework. We demonstrate that a multibiometric vault provides better recognition performance and higher security compared to a unibiometric vault. For example, our multibiometric vault based on fingerprint and iris achieves a GAR of 98.2% at a FAR of $\approx 0.01\%$, while the corresponding GAR values of the individual iris and fingerprint vaults are 88% and 78.8%, respectively. Further, we also show that the security of the system is only 41 bits when the iris and fingerprint vaults are stored separately. On the other hand, the multibiometric vault based on fingerprint and iris provides 49 bits of security.**

I. Introduction

Compared to traditional (uni)biometric authentication, multibiometric systems [1] offer several advantages such as better recognition accuracy, increased population coverage, greater security, flexibility and user convenience. However, a multibiometric system stores multiple templates for the same user corresponding to the different biometric sources. One of the main vulnerabilities of a biometric system is the exposure of a user's biometric template information. Access to a user's template can lead to (i) creation of physical spoofs (see [2]), (ii) replay attacks, and (iii) cross-matching across different databases to covertly track a person. Furthermore, unlike passwords or tokens, compromised biometric templates are not revocable. Due to these reasons, template security is essential to protect both the integrity of the biometric system and the privacy of the users. Although a number of approaches have been proposed to secure templates [3], most of these schemes have been designed primarily to secure a single template. While it is possible to apply these template protection schemes separately to each individual template in a multibiometric system, such an approach is not optimal in terms of security. Protecting the individual templates separately is analogous to having a system that requires multiple smaller passwords, which is less secure than a system that uses a single large password.

This research was supported by the Center for Identification Technology Research at West Virginia University.

Karthik Nandakumar is with Institute for Infocomm Research, A*STAR, Fusionopolis, Singapore, knandakumar@i2r.a-star.edu.sg

Anil K. Jain is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA, jain@cse.msu.edu

The problem of multibiometric template security has not been adequately addressed in the literature. To the best of our knowledge, the only reported work on this problem is the secure sketch approach proposed by Sutcu et al. [4]. Even this approach has not been evaluated on biometric databases with real intra-user variations. In this paper, we propose *a unified scheme to secure multiple templates of a user* in a multibiometric system by (i) transforming features from different biometric sources (e.g., fingerprint minutiae and iriscodes) into a common representation, (ii) performing feature-level fusion to derive a single multibiometric template, and (iii) securing the multibiometric template using a single fuzzy vault construct [5].

II. Fuzzy Vault Framework

Biometric template protection schemes can be broadly classified into *feature transformation* approach and *biometric cryptosystems* [3]. In the feature transform approach, the biometric features are modified using a transformation function, whose parameters are typically derived from a random key. Only the transformed template is stored and matching takes place directly in the transformed domain. The feature transform approach can be further categorized as (i) *salting* - the transform is invertible, so the security is based on the secrecy of the key, and (ii) *non-invertible transform* - a one-way function where it is computationally hard to invert a transformed template even if the key is known.

In a biometric cryptosystem [6]–[10], some public information about the biometric template (referred to as *helper data*) is stored. The helper data is usually obtained by binding a key K (that is independent of the biometric features) with the template T. Hence, such schemes are known as *key-binding biometric cryptosystems*. Matching is performed indirectly by recovering the key from the helper data using the query biometric features (Q) and verifying the validity of the recovered key. Error correction coding techniques are typically used to handle intra-user variations.

A well known example of biometric cryptosystem is the fuzzy vault framework [5], which is designed to secure biometric features that are represented as an unordered set. Let X denote a biometric template with r elements. The user selects a key K, encodes it in the form of a polynomial P of degree n and evaluates the polynomial P on all the elements in X. The points lying on P are hidden among a large number (denoted by s) of random chaff points that do not lie on P and the union of genuine and chaff point sets constitutes the helper data or vault V. In the absence of user's biometric data, it is computationally hard to identify the genuine points in V, and hence the template is secure.

During authentication, the user provides a biometric query denoted by $X^{'}$. If $X^{'}$ overlaps substantially with X, the user can identify many points in V that lie on the polynomial. If the number of discrepancies between X and $X^{'}$ is less than $(r-n)/2$, Reed-Solomon decoding can be applied to reconstruct P and the authentication is successful. On the other hand, if X and $X^{'}$ do not have sufficient overlap, it is infeasible to reconstruct P and the authentication is unsuccessful. While the fuzzy vault scheme is not a perfect template protection technique [11], [12], it is robust to intra-user variations in the biometric data and implementations of the vault for fingerprint, face, iris and signature modalities have been proposed. We propose a fuzzy vault implementation to secure multibiometric templates as a single entity.

III. Multibiometric Fuzzy Vault

In our multibiometric vault implementation, the biometric features are represented as elements in the Galois Field GF(2^{16}) and the key size is set to $16n$ bits, where n is the degree of the polynomial P. As in [13], we replace the Reed-Solomon polynomial decoding step by a combination of Lagrange interpolation and Cyclic Redundancy Check (CRC) based error detection. During authentication, the query biometric features are used to filter out the chaff points in the vault V resulting in an unlocking set $L^{'}$. Several candidate sets of size $(n+1)$ are generated from $L^{'}$ and polynomials are reconstructed using Lagrange interpolation. CRC based error detection is used to identify the correct polynomial and hence, decode the correct key. Though this method has a higher computational cost due to the large number of interpolations, it has better tolerance to errors.

The critical component of our multibiometric vault is the transformation of features from different biometric sources (e.g., fingerprint minutiae and iriscodes) into a common unordered set representation. We first describe how fingerprint minutiae and iriscodes can be individually encoded as elements in GF(2^{16}) and then show how the multibiometric template can be derived in the following three scenarios: (i) multiple impressions of the same finger, (ii) multiple instances of a biometric trait (e.g., left and right index fingers) and (iii) multiple traits (e.g., fingerprint and iris).

A. Fingerprint Minutiae Encoding

We follow the approach proposed by Nandakumar et al. [13] for fingerprint minutiae encoding. The location and orientation attributes of a minutia point are quantized and concatenated in order to obtain a 16-bit number, which is then considered as an element in GF(2^{16}). Only a fixed number (denoted by r) of minutiae are selected for vault construction based on their quality. To facilitate the alignment of query minutiae to the template, a set of high curvature points are extracted from the template image and stored along with the vault. The high curvature points do not reveal any information about the minutiae. During authentication, the aligned query minutiae are used to coarsely filter out the chaff points in the vault. A minutiae matcher is then applied to find correspondences between the query minutiae and the remaining points in the vault. Vault points having a matching minutia in the query constitute the unlocking set.

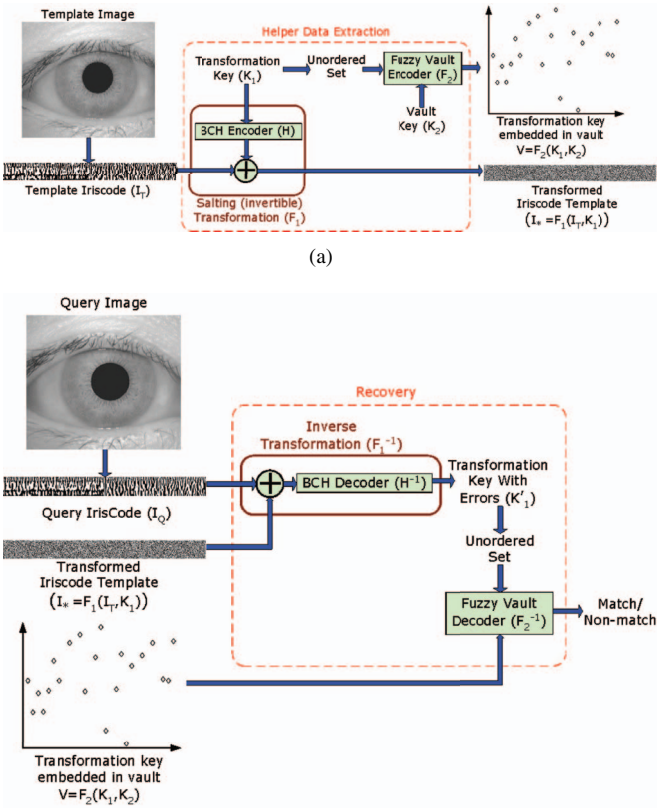

Fig. 1. Schematic diagram of the iris based fuzzy vault. (a) Helper data extraction and (b) authentication.

B. Encoding of Iriscode Features

The most common representation scheme used for matching iris images is the iriscode [14]. Since the iriscode is a fixed length binary vector, we cannot secure the iriscode directly using the fuzzy vault framework. To overcome this problem, we construct the iris cryptosystem in two steps (see Figure 1(a)). In the first step, we apply a salting (invertible) transform to the iriscode template based on a randomly generated transformation key. Since the transform is invertible, the security of the transformed iriscode template relies on the security of the key. In the second step, we represent the transformation key as an unordered set and secure it using the fuzzy vault construct. Both the transformed iriscode template and the vault that embeds the transformation key constitute the helper data in our iris cryptosystem.

The proposed iris cryptosystem has two main advantages. Firstly, the salting step can be considered as a feature transformation function that converts a fixed length binary vector into an unordered set. This enables us to secure diverse biometric templates such as fingerprint minutiae and iriscode as a single multibiometric fuzzy vault. Moreover, both the salting and fuzzy vault steps can account for intra-user variations in the iriscode template. Due to the presence of

two layers of error correction, the proposed iris cryptosystem allows larger intra-user variations in the iriscode template and hence, provides a high genuine accept rate.

The salting transform consists of two operations, namely, BCH encoding and an Exclusive-OR (XOR) operation. Let I_T be a iriscode template of length N_I bits that is to be secured using the fuzzy vault framework. First, we partition the template I_T into r non-overlapping components $\left[I_T^1, \cdots, I_T^r\right]$ such that each component contains exactly M_I bits. Next, we randomly generate r binary vectors $K^1, \cdots, K^r$ each of length M_K bits. These r random binary vectors together constitute the transformation key K_1 of length rM_K bits. The BCH encoder is applied individually to the binary vectors $K^1, \cdots, K^r$ to obtain codewords $H(K^1), \cdots, H(K^r)$. Finally, an XOR operation is performed between the r codewords generated by the BCH encoder and the corresponding components of the iriscode template to obtain the components of the transformed iriscode. The transformed iriscode template I_* can be represented as $\left[I_*^1, \cdots, I_*^r\right]$, where the j^{th} component I_*^j is given by $I_*^j = I_T^j \oplus H(K^j)$, $\oplus$ denotes the XOR operation. Hence, the complete salting transformation can be represented as a function F_1 that takes the iriscode template I_T and the transformation key K_1 as inputs and generates the transformed iriscode I_* such that $I_* = F_1(I_T, K_1)$. Since the value of M_K is set to 16 in our implementation, we can represent the r components of the transformation key as elements in $GF(2^{16})$. Hence, the components of the transformation key K_1 can be directly represented as an unordered set and secured using the fuzzy vault.

During authentication (see Figure 1(b)), the inverse salting transform is applied to the transformed iris code template I_* using the query iriscode I_Q. Let I_Q be the query iriscode of length N_I bits. We partition the query I_Q into r non-overlapping components $\left[I_Q^1, \cdots, I_Q^r\right]$ and an XOR operation is performed between the r components of the query iriscode and the *corresponding* components (the order of the iriscode bits is preserved) of the transformed iriscode to obtain the corrupted codewords. If the Hamming distance between a corrupted codeword and the corresponding original codeword is less than the error correcting capability of the BCH code, the corrupted codeword can be correctly decoded to obtain a component of the transformation key K_1. The components of the recovered transformation key are represented as an unordered set, which is then used for vault decoding. If the template and query iriscodes are sufficiently similar, the recovered key will be sufficiently similar to K_1 and hence, the vault can be successfully decoded.

C. Feature Level Fusion

When multiple fingerprint impressions of the same finger are available, we can apply a mosaicing technique [15] to combine the minutiae from the individual images into a single mosaiced template. When multiple instances of the same biometric trait are available for a user, we can obtain the multibiometric template by concatenating the different feature sets. For example, if $M_{F_1}^T$ and $M_{F_2}^T$ are the template minutiae sets derived from the right and left index fingers of a user, respectively, the combined minutiae set M_F^T can be obtained as the union of the sets $M_{F_1}^T$ and $M_{F_2}^T$. The fuzzy vault for the combined minutiae set M_F^T can be constructed using the same procedure described in section III-A. The high curvature points from both the fingers are stored separately along with the single multibiometric vault. During authentication, the query and template minutiae sets of the two fingers are aligned independently. The aligned query minutiae sets of the right and left index fingers are used to filter the chaff points from the vault to generate two unlocking sets $L_{F_1}^{'}$ and $L_{F_2}^{'}$. Either the union or the largest unlocking set can be considered as the final unlocking set that is used for polynomial reconstruction.

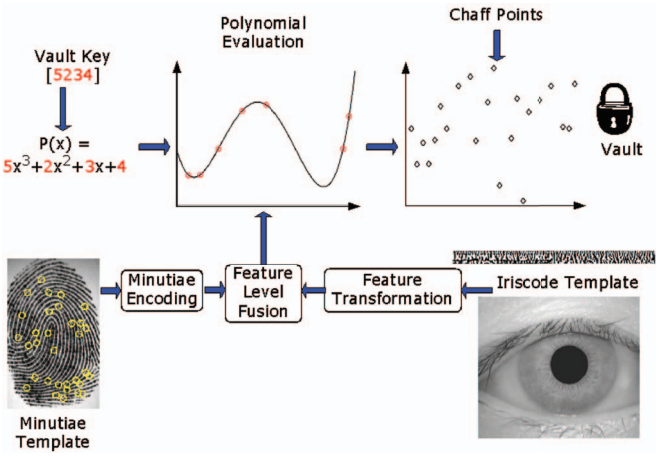

Fig. 2. Schematic diagram of a multimodal (fingerprint and iris) fuzzy vault.

Figure 2 shows the encoding phase of a multimodal fuzzy vault with fingerprint and iris modalities. In this scenario, a feature transformation function is applied to the iriscode template to convert it into an unordered set with the help of a transformation key. Let X_F and X_I be the set of feature points generated by the fingerprint and iris modalities, respectively. The union, X, of the two sets X_F and X_I is formed such that the Hamming distance between any two elements in the union is greater than or equal to 2. The high curvature points from the fingerprint and the transformed iriscode template are stored along with the vault as helper data. During authentication, the query iriscode is used to recover the transformation key from the transformed iriscode template. The aligned query minutiae set and the recovered iris transformation key are used to filter the chaff points from the vault and two unlocking sets $L_F^{'}$ and $L_I^{'}$ are generated. The union of the two unlocking sets is considered as the final unlocking set that is used for polynomial reconstruction.

IV. Experimental Results

The performance of the fingerprint fuzzy vault has already been evaluated in [13] on the FVC2002 and MSU-DBI fingerprint databases. For evaluating the vault on multi-finger data, we use only the MSU-DBI database because the public-

domain FVC databases do not have images from multiple fingers of a person. The MSU-DBI database contains two pairs of impressions for each of the 160 users from four different fingers (two index and two middle fingers). The two pairs of impressions were collected six weeks apart. Hence, this database is suitable to study the multiple finger and multiple impression scenarios in the fuzzy vault implementation. We use only the impressions from right and left index fingers in our experiments. For the single finger experiments, the number of genuine minutiae is between 24 and 36 and the number of chaff points in the vault is 300. The degree of the polynomial is set to 11 or 12. When the number of minutiae in the template and/or query fingerprint is less than the required number of genuine points, namely 24, we call it as failure to capture (FTC) error. In the case of the multifinger vault, 48 to 72 genuine points are used in the vault and the number of chaff points in the vault is 600.

The performance of the iris cryptosystem has been evaluated on the CASIA iris image database ver 1.0 [16]. This database consists of images from 108 different eyes with 7 images per eye collected over two sessions and we use one image from each session for evaluation. We use the algorithms described in [17] for pre-processing, segmentation and extraction of iriscodes from the iris images. In our implementation, the Gabor phase responses are sampled at 48 different radii and 360 angles to generate a $(48\times360\times2)$-bit iriscode. Further, we partition the iriscode into $r(=48)$ components with each partition containing 1023 bits. We use a $(1023, 16)$ BCH coding scheme, which can correct up to 247 errors in a 1023-bit codeword. Thus, the BCH codes are capable of correcting approximately 25% of the errors in the query iriscode. The size of the transformation key K_1 used to secure the iriscode template is (48×16) bits. The transformation key itself is secured using the fuzzy vault framework by using a vault key K_2 of size $16n$ bits, where n is the degree of the polynomial used in vault encoding. We evaluate the performance of the iris cryptosystem at two different values of n (10 and 11), which provide a false accept rate of less than 0.02%. The number of chaff points used in the vault is set to 500.

Finally, a virtual multimodal (right index finger and iris) database, consisting of 108 users derived from the MSU-DBI fingerprint and CASIA iris databases is used to evaluate the performance of a multimodal fuzzy vault that simultaneously secures the minutiae template from the right index finger and the iriscode template. The number of genuine points in the multimodal vault is between 72 and 84 and the total number of points in the vault after adding the chaff points is 884.

A. Performance of Fingerprint Vault

The performance of the multi-impression and multi-finger vault is summarized in Table I. When a single impression from the right index finger is used for encoding, the failure to capture rate (FTCR) of the system is 5.6% (see the first row of Table I). When the key size is 176 bits (corresponds to $n = 11$), the genuine accept rate (GAR) and false accept rate (FAR) of the system are 82.5% and 0.02%, respectively. From row 3 of Table I, we observe that mosaicing (multiple impressions of the same finger) reduces the FTCR from 5.6% to 2.5% and also increases the GAR of the system for the both the key sizes. In the multi-finger scenario, when the largest of the two unlocking sets $L^{'}_{F_1}$ and $L^{'}_{F_2}$ is selected as the final unlocking set $L^{'}_{F}$, the GAR improves significantly to 90% at a FAR of 0.02% compared to the single finger case. However, in this case there is no change in the size of the vault key, K_2, that determines the security of the vault. On the other hand, using the union of the two unlocking sets leads to a significant improvement in the security but leads to only a marginal improvement in the GAR.

B. Performance of Iris and Multimodal Vault

The performance of the iris cryptosystem is shown in the first row of Table II. The GAR of the iris cryptosystem is 88% at a false accept rate of less than 0.02%. The GAR of the Hamming distance-based iris matcher [14] that uses the original template and query iriscodes is approximately 94% at a FAR of 0.02%. Thus, there is some degradation in the GAR of the iris modality due to the application of the proposed template protection scheme. The reason for this degradation is that the BCH coding scheme has a strict threshold on the number of errors that can be corrected. When the number of bit differences between the template and query iriscode components is greater than 247, the corresponding components of the transformation key cannot be recovered. In some cases, features could not be reliably extracted from a relatively large region in the iris pattern due to factors like occlusion. The Hamming distance-based iris matcher compensates for this problem by ignoring the occluded regions in the Hamming distance computation. However, the proposed system cannot effectively handle this problem resulting in more false rejects. The third row in Table II shows that the GAR of the multimodal vault is significantly better than the GAR of the individual modalities.

V. Security Analysis

Dodis et al. [8] defined the security of biometric cryptosystems in terms of the min-entropy of the helper data. We analyze the security of the fuzzy vault framework by measuring the average min-entropy of the biometric template given the vault (see Table III). Recall that a vault V is an unordered set of t points consisting of r genuine and s chaff points. The vault can be decoded only if we find a candidate set $L^{''}$ consisting of $(n+1)$ genuine points. If no additional information is available, an adversary would have to decode the vault by randomly selecting subsets of $(n + 1)$ points from V, which is known as brute-force attack.

Suppose that the adversary has knowledge of p_i, where p_i is the probability that i^{th} point in the vault is a genuine point, $i = 1, \cdots, t$. Let us re-order the points in V such that $p_i \geq p_{i+1},\ \forall i = 1, \cdots, t-1$. Since there are $\binom{r}{n+1}$ combinations of candidate sets $L^{''}$ derived from V that can decode the vault and each $L^{''}$ can be ordered in $(n + 1)!$ ways, the min-entropy of a template M^T given V is

TABLE I

PERFORMANCE OF THE MULTI-IMPRESSION (MOSAICED TEMPLATE FROM TWO IMPRESSIONS) AND MULTI-FINGER (RIGHT AND LEFT INDEX FINGERS) FUZZY VAULT ON THE MSU-DBI FINGERPRINT DATABASE. THE FAILURE TO CAPTURE RATE (FTCR), GENUINE ACCEPT RATE (GAR) AND FALSE ACCEPT RATE (FAR) ARE EXPRESSED AS PERCENTAGES. THE KEY SIZE IS EXPRESSED IN BITS.

Scenario	FTCR	FAR = 0.02		FAR = 0	
		GAR	Vault Key Size	GAR	Vault Key Size
Right Index Finger	5.6	82.5	176	78.8	192
Left Index Finger	8.8	75.6	176	69.4	192
Right Index Finger (Mosaiced template)	2.5	83.1	176	81.2	192
Both Fingers (Largest of the two unlocking sets)	0	90	176	87.5	192
Both Fingers (Union of the two unlocking sets)	0	84.4	304	78.8	336

TABLE II

PERFORMANCE OF THE MULTIMODAL (RIGHT INDEX FINGER AND IRIS) FUZZY VAULT ON THE VIRTUAL MULTIMODAL DATABASE DERIVED FROM THE MSU-DBI FINGERPRINT AND CASIA IRIS DATABASES.

Scenario	FTCR	FAR = 0.02		FAR = 0	
		GAR	Vault Key Size	GAR	Vault Key Size
Iris	0	88	160	88	176
Right Index Finger	5.6	82.5	176	78.8	192
Right Index Finger + Iris (Union of the two unlocking sets)	0	98.2	208	98.2	224

TABLE III

SECURITY OF THE PROPOSED FUZZY VAULT IMPLEMENTATIONS. HERE, THE SECURITY IS MEASURED IN TERMS OF THE AVERAGE MIN-ENTROPY OF THE BIOMETRIC TEMPLATE GIVEN THE VAULT. THE PARAMETERS t, r AND n REPRESENT THE TOTAL NUMBER OF POINTS IN THE VAULT (GENUINE AND CHAFF), NUMBER OF GENUINE POINTS IN THE VAULT AND THE DEGREE OF THE POLYNOMIAL USED IN THE VAULT, RESPECTIVELY.

Modality	Assumptions	Parameters			Security (bits)
		t	r	n	
Fingerprint	Uniform distribution of minutiae	330	30	10	40
	Distribution of minutiae follows mixture model [18]	336	24-26	10	27
Iris	Iriscode has inherent entropy of 249 bits [19]; BCH code corrects up to 25% of the errors	548	48	10	40
Fingerprint + Iris	Uniform distribution of minutiae; iriscode has inherent entropy of 249 bits [19]; BCH code corrects up to 25% of the errors	884	84	13	49

$$H_\infty(M^T|V) \geq -log\left(\frac{\binom{r}{n+1}(n+1)!\prod_{i=1}^{n+1}p_i}{\prod_{i=1}^{n}\left(1-\sum_{k=1}^{i}p_k\right)}\right). \quad (1)$$

If we assume that the genuine and chaff points are uniformly distributed, $p_i = 1/t, i = 1, 2, \cdots, t$. This corresponds to the brute-force attack scenario and in this scenario, the min-entropy of M^T given V can be simplified as

$$H_\infty(M^T|V) = -log\left(\frac{\binom{r}{n+1}}{\binom{t}{n+1}}\right). \quad (2)$$

A. Fingerprint Vault

When the size of the vault key is 160 bits (which corresponds to $n = 10$ in our implementation), the number of genuine and chaff points in the vault are 30 and 300, respectively, and if we assume uniform distribution of minutiae, the min-entropy of a fingerprint vault is approximately 40 bits. While a security of 40 bits may be considered inadequate from the cryptographic point of view , it must be noted that the fuzzy vault framework eliminates the key management problem, which is a major issue in practical cryptosystems.

Given a database of fingerprints, we can estimate the minutiae distribution using the mixture models proposed by Zhu et al. [18]. Based on these estimated distributions, we can compute p_i for all the points in V and thereby the average min-entropy for the database. For the MSU-DBI database, the average min-entropy when r is between 24 and 36, $n = 10$ and $t = 336$ is approximately 27 bits. This large entropy loss (from 40 bits in the brute force case to 27 bits) is because we generate the chaff points from a uniform distribution. Therefore, the chaff points do not follow the true minutiae distribution, making it easier to separate the chaff points from the genuine points. One way to improve the vault security is to estimate the distribution of minutiae during vault encoding and sample chaff points from this distribution.

B. Iris Cryptosystem

In the proposed iris cryptosystem, the helper data consists of two components, namely, the transformed iriscode template I_* and the vault V that secures the transformation key K_1 used to obtain I_*. Since the transformation key K_1

is independent of the template iriscode, it can be generated from a uniform distribution. Therefore, the min-entropy of K_1 given V ($H_\infty(K_1|V)$) can be computed using equation (2). In our implementation, $r = 48$, $n = 10$ and $t = 548$. Hence, $H_\infty(K_1|V)$ is approximately 40 bits.

Since we use a single XOR operation to obtain I_*, the min-entropy of template iriscode I_T given I_* ($H_\infty(I_T|I_*)$) depends only on the redundancy added to the key K_1 by the BCH encoder. Hao et al. [19] have estimated that in the worst case of an adversary having perfect knowledge of the correlation between the iriscode bits, the inherent uncertainty in a iriscode template is approximately 249 bits. They also showed that if a coding scheme can correct up to w bits in the iriscode template, the entropy of the iriscode template given the transformed template is approximately $\log\left(2^{249}/\binom{249}{w}\right)$ bits. In our implementation, the BCH coding scheme can correct up to 25% of the errors, which corresponds to approximately $w = 62$ bits (out of 249). Therefore, entropy of the I_T given I_* ($H_\infty(I_T|I_*)$) is approximately 52 bits. The overall security of the iris cryptosystem is given by $min\left(H_\infty(K_1|V), H_\infty(I_T|I_*)\right) \approx min(40, 52) \approx 40$ bits. The security of a comparable key-binding cryptosystem proposed by Hao et al. [19] is approximately 44 bits.

C. Multimodal Vault

In the case of the multimodal vault, both the minutiae template M^T and the transformation key K_1 used in the iris cryptosystem are secured using a single vault V. Therefore, decoding the vault reveals both the M^T and K_1 (and consequently I_T). Hence, the overall security of the system is given by $min\left(H_\infty(M^T, K_1|V), H_\infty(I_T|I_*)\right)$. In the multimodal vault, $t = 884$, $n = 13$, and the number of genuine points, r, is 84 (36 from the fingerprint modality and 48 from the iris modality). If we assume that the minutiae are uniformly distributed, $H_\infty(M^T, K_1|V)$ is approximately 49 bits. Hence, the overall security of the multimodal vault is approximately $min(49, 52) = 49$ bits.

Suppose that we construct two separate vaults V_F and V_I for the fingerprint and iris modalities, respectively. In this scenario, the overall security of the system will be $min\left(\log\left(2^{H_\infty(M^T,|V_F)} + 2^{H_\infty(K^1,|V_I)}\right), H_\infty(I_T|I_*)\right)$, which is approximately 41 bits for the same number of chaff points (300 for fingerprint and 500 for iris). Thus, the multimodal vault proposed here provides a significantly higher security (49 bits) compared to storing the individual templates using separate vaults.

VI. Summary

We have proposed a framework for securing multiple biometric templates of a user in a multibiometric system as a single entity. This is achieved by generating a single multibiometric template using feature level fusion and securing the multibiometric template using the fuzzy vault construct. We have also implemented a fully automatic fuzzy vault system for securing the fingerprint minutiae and iriscode templates. While we use an existing fingerprint fuzzy vault implementation to secure fingerprint minutiae, we propose a new vault implementation for securing iriscodes. A salting transformation based on a transformation key is used to indirectly convert the fixed-length binary vector representation of iriscode into an unordered set representation that can be secured using the fuzzy vault. We have shown that the multibiometric vault can secure templates from different biometric sources such as multiple fingerprint impressions, multiple fingers and multiple modalities such as fingerprint and iris. We have also demonstrated that the multibiometric vault provides better recognition performance as well as higher security compared to the unibiometric vaults.

References

[1] A. Ross, K. Nandakumar, and A. K. Jain, *Handbook of Multibiometrics*. Springer, 2006.

[2] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni, "Fingerprint Image Reconstruction From Standard Templates," *IEEE Trans. on PAMI*, vol. 29, no. 9, pp. 1489–1503, 2007.

[3] A. Vetro and N. Memon, "Biometric System Security," Tutorial presented at Second International Conference on Biometrics, Seoul, South Korea, August 2007.

[4] Y. Sutcu, Q. Li, and N. Memon, "Secure Biometric Templates from Fingerprint-Face Features," in *Proceedings of CVPR Workshop on Biometrics*, Minneapolis, USA, June 2007.

[5] A. Juels and M. Sudan, "A Fuzzy Vault Scheme," in *Proc. of IEEE Intl. Symp. on Info. Theory*, Lausanne, Switzerland, 2002, p. 408.

[6] P. Tuyls, A. H. M. Akkermans, T. A. M. Kevenaar, G.-J. Schrijen, A. M. Bazen, and R. N. J. Veldhuis, "Practical Biometric Authentication with Template Protection," in *Proceedings of Fifth Intl. Conf. on AVBPA*, Rye Town, USA, July 2005, pp. 436–446.

[7] S. C. Draper, A. Khisti, E. Martinian, A. Vetro, and J. S. Yedidia, "Using Distributed Source Coding to Secure Fingerprint Biometrics," in *Proceedings of ICASSP*, vol. 2, Hawaii, April 2007, pp. 129–132.

[8] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," Cryptology ePrint Archive, Tech. Rep. 235, February 2006, A preliminary version of this work appeared in EUROCRYPT 2004.

[9] Y. Sutcu, Q. Li, and N. Memon, "Protecting Biometric Templates with Sketch: Theory and Practice," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 503–512, September 2007.

[10] I. R. Buhan, J. M. Doumen, P. H. Hartel, and R. N. J. Veldhuis, "Fuzzy Extractors for Continuous Distributions," in *Proceedings of ACM Symposium on Information, Computer and Communications Security*, Singapore, March 2007, pp. 353–355.

[11] W. J. Scheirer and T. E. Boult, "Cracking Fuzzy Vaults and Biometric Encryption," in *Proc. of Biometrics Symposium*, September 2007.

[12] A. Kholmatov and B. Yanikoglu, "Realization of Correlation Attack Against the Fuzzy Vault Scheme," in *Proc. of SPIE Symposium on Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, vol. 6819, San Jose, USA, January 2008.

[13] K. Nandakumar, A. K. Jain, and S. Pankanti, "Fingerprint-based Fuzzy Vault: Implementation and Performance," *IEEE Trans. on Info. Forensics and Security*, vol. 2, no. 4, pp. 744–757, December 2007.

[14] J. Daugman, "How Iris Recognition Works?" *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 21–30, 2004.

[15] A. Ross, S. Shah, and J. Shah, "Image Versus Feature Mosaicing: A Case Study in Fingerprints," in *Proceedings of SPIE Conference on Biometric Technology for Human Identification*, vol. 6202, Orlando, USA, April 2006, pp. 1–12.

[16] L. Ma, T. Tan, Y. Wang, and D. Zhang, "Personal Identification Based on Iris Texture Analysis," *IEEE Trans. on PAMI*, vol. 25, no. 12, pp. 1519–1533, December 2003.

[17] S. Shah, "Enhanced Iris Recognition: Algorithms for Segmentation, Matching and Synthesis," Master's thesis, Department of Computer Science and Electrical Engineering, West Virginia University, 2006.

[18] Y. Zhu, S. C. Dass, and Jain, "Statistical Models for Assessing the Individuality of Fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 391–401, September 2007.

[19] F. Hao, R. Anderson, and J. Daugman, "Combining Crypto with Biometrics Effectively," *IEEE Transactions on Computers*, vol. 55, no. 9, pp. 1081–1088, September 2006.