Fuzzy Extractors for Asymmetric Biometric Representations

Qiming Li
Institute for Infocomm Research, Singapore
qiming.li@ieee.org

Muchuan Guo
National University of Singapore
guomuchu@comp.nus.edu.sg

Ee-Chien Chang
National University of Singapore
changec@comp.nus.edu.sg

Abstract

Fuzzy extractors are recently proposed error-tolerant cryptographic primitives that are potentially useful to protect biometric templates. However, there are challenges in adopting these primitives. Firstly, fuzzy extractors require the data obtained during both enrollment and verification to be in the same feature representation. However, for better performance on ROC, multiple high quality samples can be obtained during enrollment, which result in an asymmetric setting whereby data obtained in enrollment and verification are stored in different representations. Secondly, fuzzy extractors only concern about the strength of the secret key extracted, and does not directly assure that privacy is preserved. In this paper, we consider a simplified asymmetric setting and propose a sketch scheme. We analyze the key strength measured by the number of secret bits that can be extracted, and the privacy measured by the information leakage on the user identities. We next apply and investigate the scheme on FVC fingerprint datasets.

1. Introduction

Recent developments in biometrics lead to many promising applications and wide spread adoptions of biometric systems. As such systems gaining popularity, the concerns on privacy are also getting more attention. Since biometric data cannot be easily revoked or replaced, it is essential to protect the data from being revealed while supporting applications such as verification and identification. Moreover, due to the strong bond between identities and biometrics, it is also often crucial to protect the privacy of users in these systems.

Security and privacy concerns of cryptographic systems utilizing biometric data have attracted the attention from both signal processing and cryptography/security communities. Various solutions have been proposed in recent years, ranging from ad-hoc transformations which are intended to be hard to invert, to rigorous theoretic studies on error-tolerant cryptography assuming simplified models of biometric data.

In this paper, we take a closer look at *fuzzy extractors* [8] that mainly utilize *sketches* to help in recovering original biometric templates, and *strong extractors* to extract secret keys from templates. In spite of many recent developments (Section 2.1), there are still challenges in applying fuzzy extractors to real biometric data. In particular, we observe that existing notions of fuzzy extractors

1. Do not directly support asymmetric settings, and
2. Do not necessary preserve privacy.

In the following, we are going to explain more clearly what these issues are and why they are challenging.

Asymmetric Biometric Representations The formulation of fuzzy extractors assumes that the biometric data obtained during enrollment and verification are in the same domain and a distance function could be defined on it. However, for better performance in terms of ROC, a biometric template can be a description of the distribution of (noisy) biometric data from an individual, which can be estimated during enrollment from multiple samples.

In this case, the representations of the biometric data are *asymmetric* in the sense that, a template computed during enrollment would contain much more information than the data that can be obtained during verification. But is it feasible (and if so, how) to recover the original template, which is a distribution, from a single sample obtained during verification? What would be the information leakage in this case? Such asymmetry in the amount of information certainly needs to be studied more carefully, and the constructions of secure sketches (and fuzzy extractors) in this case is non-trivial.

Privacy Preservation Initially, fuzzy extractors are designed for key extraction, and do not directly address privacy preservation of the templates. For example, although a secure fuzzy extractor may allow the extraction of a strong cryptographic key from the biometric data, if the sketches are compromised, they may still reveal identities of the users. The situation is made worse with asymmetric biometric representations, since it requires more information to be stored to successfully extract consistent keys from biometric data. This may lead to certain types of privacy attacks to be described in Section 3.

To illustrate the privacy concerns, let us consider an application where a fuzzy extractor is used on a mobile phone such that only the owner can unlock it using his/her biometric samples. A privacy sensitive user may require that even if the phone is stolen and the sketch stored in it is compromised, the attacker should not be able to find out his/her identity. However, this is not guaranteed by the fuzzy extractor no matter how strong the extracted key is.

In this paper, we study how to build secure sketches for asymmetric representations of biometric data. In particular, we take a look at a minutiae-based fingerprint authentication scheme similar to that in [6], and propose a secure sketch scheme. We analyze the security by investigating the information leakage of the template, which in turn determines the entropy of the secret key extracted. This is for applications that use the extracted secret bits for subsequent cryptographic operations. Furthermore, we examine the implications of these sketches on users' privacy by giving some attack algorithms. We found that, although asymmetric biometric representations may improve the ROC and increase the key strength, potentially it would leak information about the identities and thus it is not suitable in preserving privacy.

We give a brief background on the related work in Section 2.1. We introduce our security model (Section 3) and give a generic scheme in simplified asymmetric setting (Section 4). An identity verification scheme based on fingerprint minutiae is presented in Section 5, followed by its security and privacy analysis in Section 6. We conclude in Section 8.

2. Background

2.1. Related Work

Perhaps the most well-known pioneer formal analysis on how to extract consistent keys from noisy data are the fuzzy commitment scheme [12], which handles Hamming distance, and the fuzzy vault scheme [11], which deals with set-difference. Dodis et al. [8] give a generalized and more formal framework called fuzzy extractors and prove that secure sketches imply fuzzy extractors. They also give various improvements and extensions to previous schemes.

Although technically sound, it is not easy to apply the above results to real biometric data. For example, the matching between a sample and a template may be complicated and may not define a metric space [5], and the original data may not even be discrete and/or need to be quantized [15, 22]. Some other (maybe less rigorous) attempts to apply fuzzy extractors on real biometric data include [24, 1, 6, 13].

A formal study of information leakage due to multiple sketches of the same individual is first discussed in [2], and later generalized in [3], which considered a few different attack models. It is known that with multiple sketches of the same fingerprint, the true minutiae may be revealed [21, 14].

Another line of research [16, 26, 25] handles continuous data with additional assumptions on the distributions of the biometrics, and uses mutual information as the security measure.

There have been a number of practical results including fingerprints [7, 27], iris patterns [9], voice features [17], hand grip patterns [4], and face features [23]. Other methods to protect biometric templates but not necessarily aiming at extracting a key include [19, 20]. It should be noted, however, that these results are less formal than well-established cryptographic techniques. For example, sometimes only brute-force attackers are considered.

It is worth to note that in some previous work such as [13, 4, 10], some extra data (namely, the location of the most reliable feature components) is stored and sometimes revealed for consistent extraction of the key. Such extra data can be viewed as descriptions of individual distributions, and our asymmetric model can be viewed as a generalized model for such cases.

2.2. Secure Sketch and Fuzzy Extractors

Informally, a secure sketch P is some information computed from noisy data X such that (1) P does not reveal too much information about X, and (2) given a Y that is similar to X according to some similarity measure, X can be reconstructed from Y and P.

A fuzzy extractor can be constructed from a secure sketch and used for verification of biometric data as the following. During enrollment, some biometric measurements are made, and a sketch P is computed from biometric data X obtained from the measurements. In addition, an *extractor* (such as pair-wise independent hash functions) can be applied on X to extract a key K. During verification, another measurement of the same biometrics is made to obtain data Y. If X and Y are sufficiently similar, according to the properties of secure sketches, X can be reconstructed from Y and P, hence the same key K can be extracted from X using the same extractor. In this way, the key K can be used in the same way as a cryptographic key or a password for verification. By carefully choosing the parameters, the key K can be made almost uniform and independent of P [8].

When a sketch P is compromised, an adversary may obtain partial information about X. Dodis et al. [8] formulate this as the *entropy loss*, which is the upper bound of the difference between the min-entropy of X and the length (strength) of the key K. Note that P does not leak information about K.

When the enrollment data describes distributions instead of just samples, it may contain too much entropy such that entropy loss appears to be very high or very difficult to analyze precisely (e.g., when it involves real numbers). Hence, in an asymmetric setting, it makes sense to only consider the entropy loss on the data X that is actually recovered during verification, and from which the key is extracted. Nevertheless, we need to ensure that other parts that are not considered in the computation of entropy loss are independent from the extracted key.

3. Security Models for Biometric Systems

Let us consider two applications described in the introduction. The first intends to extract a secret key K, and its security level is measured by the *key strength*, which is the length of K. The second application concerns about privacy protection. There are a number of different privacy attacks. In this paper, we only consider the scenarios in Table 1, and use *identity leakage* to measure the security, which is basically the amount of information about the identities leaked by the stored templates.

	KS	KB	Attack objective
S1	Many	One	Which sketch belongs to this user?
S2	One	Many	Which user enrolled this sketch?
S3	Many	Many	Match sketches and identities.

Table 1. Privacy Attacks

In this table, “KS” stands for “knowledge of sketches”, and “KB” stands for “knowledge of biometric samples”. In Scenario S1, the attacker knows all the sketches in a database and obtains one sample from a user. The attacker’s goal is to determine which sketch belongs to this user. In Scenario S2, the attacker knows only one sketch in a database but has the access to biometric samples of many users. The goal of the attacker in this case is to identify which user this sketch belongs to. Scenario S3 is a combination of these attacks.

It is worth noting that a scheme can achieve high key strength, yet suffer from high identity leakage. This is potentially so in an asymmetric setting, where a relatively large amount of information is required to describe each individual’s distribution, and such description itself could be used to identify the individuals.

There are naturally many variations of the scenarios in Table 1. For example, instead of finding out which sketch corresponds to a given identity, we could ask if an individual is enrolled in the database at all. Furthermore, we could consider two databases with sketches, and see if it is feasible to find out if some individuals have enrolled in both databases. Or we can try to find more about the key from multiple sketches (i.e., correlation attack [14]). Nevertheless, in this paper we only examine the above 3 scenarios and leave other variations as further work.

4. A Fuzzy Extractor Scheme

Here we give a simplified model for asymmetric biometric representations and propose a fuzzy extractor scheme. Despite its simplicity, our model is general, and is potentially applicable to various types of biometric data. As an example of application, we give a concrete construction on fingerprints in Section 5.

Feature Representations We assume that each biometric sample S is represented as a real feature vector of length n. That is, $S = (s_1, \cdots, s_n)$, where $s_i \in \mathbb{R}$. We further assume that each component s_i in the vector is independent, but is associated with a different weight w_i that represents its “importance”. The locations of those important components may be different for different individuals.

During enrollment, m samples are obtained from each individual and feature vectors computed. Whereas during verification, only one sample is acquired from an individual.

Component Grouping We compute the weights $(w_1, \cdots, w_n)$ from the m feature vectors obtained during enrollment. According to these weights, we divide the components into q groups. The grouping is done such that the total weight for each group are approximately the same. After that, the components in each group is combined into a single combined component that represents the group. Let $G = (G_1, \cdots, G_q)$, where each G_i describes the indices of the components in the i-th group, and how they are combined. Furthermore, for each combined component, we quantize it to r bits, where r is a pre-determined parameter. Let X be the resulting binary string of length qr.

The template for an individual consists of the binary string X, and the grouping information G.

Construction of Sketch Given a biometric template (X, G), an encoder $\mathcal{E}$ computes a sketch P_X for X using a known symmetric secure sketch scheme, and output the tuple $P = (P_X, G)$ as the final sketch. Given another binary string Y, the recovery algorithm $\mathcal{R}$ computes an X' from P_X and Y, and we will have $X = X'$ if Y and X are similar enough. The actual sketch scheme depends on r, q, and the similarity measure between X and Y.

This asymmetric sketch is different from a symmetric sketch in a few ways. First of all, part of the sketch P (i.e., the grouping information G) is only used to compute the binary string Y from a new sample, and is not involved in the recovery of X from Y. Secondly, although G may reveal information about the original data obtained during enrollment, it is tricky to include G in the computation of entropy loss, since G may be of very high entropy, and the entropy loss can be difficult to bound. Nevertheless if G can be made independent from X, we can conclude that G

would not reveal any information about the extracted key, and hence the entropies can be computed on X and P_X only, as if we only consider a symmetric case. However, G may indeed reveal information about the identities, as in the case of our scheme for fingerprints (Section 5).

Reconstruction During verification, only a single scan is obtained. The same feature vector of n components is computed, and the grouping G is applied to obtain a binary string Y of length q. The sketch P_X is then used with Y to recover the original X.

5. A Minutiae-Based Verification Scheme

We consider locations of minutiae extracted from fingerprint images as biometric features. We assume that the samples are pre-aligned, for example, using the scheme proposed by Nandakumar et al. [18]. The scheme follows the generic scheme presented in Section 4 with the following necessary details specific to the fingerprint data.

Feature Vector Generation The minutiae of a fingerprint are transformed such that they are represented as a vector of independent components. We use a transformation T similar to the method in [6] to obtain n components $w_1, w_2, \ldots, w_n$ for each set of minutiae.

In particular, We randomly draw n straight lines in the 2-D space, and for each line, we record the difference between the number of minutiae on the "left" of the line and that on the "right", where the left and right are defined arbitrarily but consistent for all fingerprints. Next, from a library of scanned fingerprints (we use FVC 2006 datasets in our experiments), we extract the n components for each fingerprint and treat them as the training data. Using PCA, we obtain a linear transformation whereby n pairwise independent components can be obtained from each fingerprint. The linear transformation is then set as a global parameter.

Computation of Weights For the i-th component, the standard deviation s_i and the mean m_i are estimated from the m samples. We treat the ratio $w_i = m_i/s_i$ to be the SNR for the component.

Grouping and Combination For the i-th group, let the components be $d_{i,1}, \cdots, d_{i,g}$, and we combine the components by computing $x_i = \sum_{j=1}^{g} c_j d_j$, where $c_j \in \{-1, 1\}$. The coefficients c_i's are chosen such that the absolute value of x_i is maximized, yet the sign of x_i is uniformly randomized. For example, we can randomly choose x_i to be negative first, and then set c_j to be -1 for all positive d_j, and vice versa. We assign one bit for each group (i.e., $r = 1$), such that for the i-th group, the assigned bit $b_i = 1$ *iff* $x_i > 0$. The grouping information G_i hence contains (1) the indices of all components in the i-th group, and (2) the coefficients c_i's used in the combination.

Sketch Construction To compute a sketch P_X for X, we first determine a threshold t on the number of bit errors to tolerate, which can be computed from the data acquired during enrollment and the desired FAR and/or FRR.

Next, to compute P_X, we use a "code-offset" scheme [12, 8]. In particular, we construct a (q, k, t) binary error-correcting code $\mathcal{C}$ where the minimum distance between codewords is at least $2t + 1$. After that, we randomly choose a codeword $c \in \mathcal{C}$, and then compute the sketch $P_X = X - c$. This can be achieved by randomly choose a q-bit string and decode it using $\mathcal{C}$. During verification, after obtaining a string Y, we use the sketch P_X to recover X by first computing $w = Y - P$, followed by decoding w using $\mathcal{C}$ to get c, and finally adding P_X back to c.

6. Security Analysis

Entropy Loss due to P It is shown in [8] that the entropy loss of the code-offset scheme is bounded by $2t + 1$ if the code is perfect, where t is the number of bit flips we want to tolerate. Hence, for the scheme as in Section 5 and, it is not difficult to show that if the codebook is perfect, the entropy loss of P_X in the asymmetric setting is at most $2t + 1$.

Moreover, we can see from Section 4 that the grouping information G is indeed independent from the secret bits extracted, since it only reveals the relative signs among the components within each group, and the final bits are actually randomly chosen at the time of sketch construction.

Furthermore, an upper bound of the entropy of the extracted key that is independent of the sketch is given by the logarithm of the Varshamov-Gilbert Bound. In actual applications, a random binary codebook would suffice in most cases.

If we assume the independence of the bits in X, we can then estimate the strength of the extracted key. Some numerical analysis on an actual fingerprint database will be given in Section 7.

Privacy Attacks using G It is not difficult to show that, although allowing consistent extraction of reasonable number of secret bits, our scheme does not preserve privacy under the attack scenarios in Table 1.

For example, in Scenario S1, we can attack the scheme described in Section 5 as the following. For a given biometric sample, we perform the transformation T and examine the grouping information G of each user, and see how likely the sample belongs to that user. For moderate number of users in question, we can assess the likelihood for all of them and choose the most likely one.

In particular, we keep track of a score for each sketch-sample pair. For each group of components to be combined, the sketch contains their *relative* signs. If the signs of the corresponding components are consistent with the information in the sketch, we increase the score for this pair. Likewise, for those groups with only one component, we check if the value of the component is large enough and increase the score accordingly. In the end, we choose the identity with the highest score as the result.

Clearly, this attack does not need to access the extracted secret key. In Section 7, we will evaluate the effectiveness of such attacks on privacy.

7. Evaluation

To evaluate our scheme, we apply our scheme on two databases DB2 and DB4 from Fingerprint Verification Competition 2006 (http://bias.csr.unibo.it/fvc2006/) which are of relatively good quality. Each database contains 12 imprints for 150 fingers. We employ the free minutiae extraction program from NIST Biometric Image Software (http://fingerprint.nist.gov/NBIS/) to obtain minutiae from the databases.

For each finger, we use the first 11 imprints as the training data to build our templates and use the last one as the testing data. As we mentioned in Section 5, all 12 samples for each finger are pre-aligned.

We follow the algorithm in Section 5, and use $n = 600$ random lines to transform each set of minutiae to a feature vector with 600 components, which is then normalized. After PCA, we divide the resulting 600 components to $q = 86$ groups according to their SNR. The first 36 group of one component each are used as is. The next 78 groups have two components each. There are 40 groups with four components each, and the last 1 group has 8 components. The components in every group are combined as in Section 5. In the end, we obtained 86 combined components for each fingerprint, from which we obtain an 86-bit string X.

For each of the 12-th samples, we follow the same steps and obtain a binary string Y that is then matched with X. Based on the histogram of the number of bits matched, we can set the threshold t on the number of bit errors to tolerate and construct the sketch accordingly. Furthermore, we can estimate the key strength as $q - (2t + 1)$, which is actually the lower bound of the key strength if the all the bits in X are independent. Figure 1 shows the trade-off between the bound on key strength versus the FRR on the corresponding threshold t. We can see that the grouping step we performed always helps to extract more bits with the same FRR, although in general we cannot hope to extract a lot of bits from these fingerprints.

An ROC curve on the false accept rate and false reject rate is shown in Figure 2. The false reject curve is obtained by varying the threshold t, whereas the false accept rate is computed as the probability that a uniform random binary string is considered as a match with any given X. The false accept rate is computed in this way because of the lack of data, the fact that the components are normalized to zero-mean, and the assumption that each bit is independent.

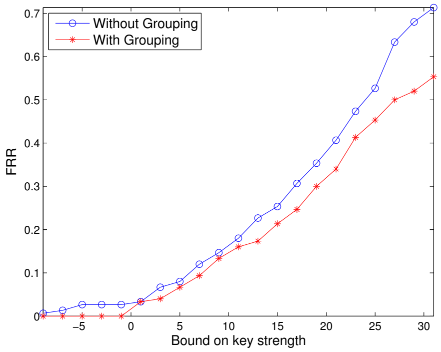

Figure 1. Key strengths

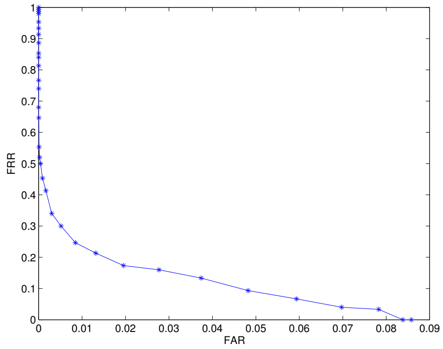

Figure 2. Performance of the scheme (ROC).

From Figures 2 and 1 we can explore various trade-offs of the scheme. For example, if we require that FAR is just below 1%, from Figure 2 we see that FRR is slightly above 20%, which corresponds to 15 secret bits in Figure 1.

To evaluate our privacy attacks as presented in Section 6, we apply our simple attack algorithm on the templates we generated, and found that in both attack Scenarios S1 and S2 in Table 1, we can identify the identity in question with an accuracy much better than random guessing.

In particular, our experiments show that for Scenario S1, we can correctly link a sample with a sketch with an accuracy of about 19% for 150 subjects, and in Scenario S2, the accuracy is about 20%. Compared with 1/150 for random guess, the leakage on identities is about $-\log 1/150 - (-\log 0.2) \approx 4.9$ bits in both scenarios. Scenario S3 is a simple combination with a similar accuracy.

This implies that the template would reveal important information on the identities that allows attackers to link sketches with identities, even when the secret keys are safe.

8. Conclusions

Key strength and privacy issues are crucial in secure biometric systems based on fuzzy extractors. We identify several challenging problems in applying fuzzy extractors under an asymmetric setting, where an enrolled biometric template has a different representation than its matching samples, which is in contrast with existing fuzzy extractor framework where both are assumed to be from the same domain. We propose a general scheme in a simple asymmetric setting. We give an scheme on fingerprints as an example of the general scheme, and evaluate it using public fingerprint datasets.

We further examine the privacy properties of the scheme by looking at how much information would be leaked about the identity by the sketches computed from the templates. We found that although the sketches are secure in the classical sense (i.e., the scheme gives reasonable key strength), they do reveal some information about the identities that allows certain types of privacy attacks.

Hence, asymmetric representations can be employed when the main concern is the key strength, but are undesirable when strong protection is required on the identities.

References

[1] A. Arakala, J. Jeffers, and K. J. Horadam. Fuzzy extractors for minutiae-based fingerprint authentication. In *International Conference on Biometrics*, 2007.

[2] X. Boyen. Reusable cryptographic fuzzy extractors. In *Proceedings of the 11th ACM conference on Computer and Communications Security*, pages 82–91. ACM Press, 2004.

[3] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith. Secure remote authentication using biometric data. In *Eurocrypt*, 2005.

[4] I. Buhan, J. Doumen, P. Hartel, and R. Veldhuis. Feeling is believing: a secure template exchange protocol. In *International Conference on Biometrics*, volume 4642 of *LNCS*, pages 897–906, 2007.

[5] E.-C. Chang and Q. Li. Hiding secret points amidst chaff. In *Eurocrypt*, volume 4004 of *LNCS*, pages 59–72, St. Petersburg, Russia, May 2006. Springer Verlag.

[6] E.-C. Chang and S. Roy. Robust extraction of secrets bits from minutiae. In *International Conference on Biometrics*, 2007.

[7] T. Clancy, N. Kiyavash, and D. Lin. Secure smartcard-based fingerprint authentication. In *ACM Workshop on Biometric Methods and Applications*, 2003.

[8] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Eurocrypt*, volume 3027 of *LNCS*, pages 523–540. Springer-Verlag, 2004.

[9] F. Hao, R. Anderson, and J. Daugman. Combining cryptography with biometrics effectively. Technical Report UCAM-CL-TR-640, University of Cambridge, 2005.

[10] F. Hao, R. Anderson, and J. Daugman. Combining crypto with biometrics effectively. *IEEE Transactions on Computers*, 55(9):1081–1088, 2006.

[11] A. Juels and M. Sudan. A fuzzy vault scheme. In *IEEE Intl. Symp. on Information Theory*, 2002.

[12] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *Proc. ACM Conf. on Computer and Communications Security*, pages 28–36, 1999.

[13] E. Kelkboom, B. Gökberk, T. Kevenaar, A. Akkermans, and M. van der Veen. "3D Face": Biometric template protection for 3D face recognition. In *International Conference on Biometrics*, volume 4642 of *LNCS*, pages 566–573, 2007.

[14] A. Kholmatov and B. Yanikoglu. Realization of correlation attack against fuzzy vault scheme. In *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819 of *Proceedings of SPIE*, 2008.

[15] Q. Li, Y. Sutcu, and N. Memon. Secure sketch for biometric templates. In *Asiacrypt*, Shanghai, China, December 2006.

[16] J.-P. M. G. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *AVBPA 2003*, pages 393–402, 2003.

[17] F. Monrose, M. Reiter, Q. Li, and S. Wetzel. Cryptographic key generation from voice. In *IEEE Symp. on Security and Privacy*, 2001.

[18] K. Nandakumar, A. K. Jain, and S. C. Pankanti. Fingerprint-based fuzzy vault: Implementation and performance. *IEEE Transactions on Information Forensics and Security*, 2(4):744–757, 2007.

[19] N. Ratha, J. Connell, and R. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001.

[20] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):561–572, 2007.

[21] W. J. Scheirer and T. E. Boult. Cracking fuzzy vaults and biometric encryption. In *Biometrics Symposium*, 2007.

[22] Y. Sutcu, Q. Li, and N. Memon. Protecting biometric templates with sketch: Theory and practice. *IEEE Transactions on Information Forensics and Security*, 2(3):503–512, 2007.

[23] A. Teoh, A. Gho, and D. Ngo. Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1892–1901, 2006.

[24] V. V. T. Tong, H. Sibert, J. Lecoeur, and M. Girault. Biometric fuzzy extractors made practical: a proposal based on fingercodes. In *International Conference on Biometrics*, 2007.

[25] P. Tuyls, A. Akkermans, T. Kevenaar, G. Schrijen, A. Bazen, and R. Veldhuis. Practical biometric authentication with template protection. In *AVBPA*, pages 436–446, 2005.

[26] P. Tuyls and J. Goseling. Capacity and examples of template-protecting biometric authentication systems. In *ECCV Workshop BioAW*, pages 158–170, 2004.

[27] S. Yang and I. Verbauwhede. Automatic secure fingerprint verification system based on fuzzy vault scheme. In *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 609–612, 2005.