Robust Extraction of Secret Bits from Minutiae*

Ee-Chien Chang[1] and Sujoy Roy[2]

[1] School of Computing, National University of Singapore, Singapore
changec@comp.nus.edu.sg
[2] Institute for Infocomm Research, Singapore
sujoy@i2r.a-star.edu.sg,

Abstract. Our goal is to extract *consistent* bits from the same fingerprint in a noisy environment. Such bits can then be used as a secret key in several cryptographic applications. In order to correct inevitable noise during scanning and processing, a known approach extracts and publishes an additional information, known as secure sketch from the minutiae. During subsequent scanning, the sketch aids in correcting the noise to give the consistent bits. However, for minutiae (represented as 2D point set), known constructions produce sketches that are large, and are difficult to adapt to slight variations of the 2D point representation. Furthermore, even with simplified model on the noise and distribution of the minutiae, it is not clear what is the entropy of the bits extracted. To overcome the problems, we suggest using a locality preserving hash in sketch construction. We give a method that produces a small sketch and thus suitable for applications involving mobile devices. Since the sketch size is small, with a reasonable assumption, we can estimate the entropy of the secret bits extracted. In addition, we can incorporate statistical properties of the noise, and distribution of the minutiae in fine-tuning the method. Our method also includes registration of fingerprints. Experiments conducted on 4000 fingerprint images from the NIST 4 database show promising results. Assuming that an intermediate representation is uniformly distributed, with FNMR = 0.09 we are able to extract about 8 secret bits (by a conservative estimate) or 10 bits (with certain assumption on the underlying codebook).

Key words: Secure sketch, Fuzzy vault, Locality preserving hash, Cryptography.

1 INTRODUCTION

Fingerprints are probably one of the most widely used biometrics today. A typical fingerprint based authentication system, when given a template (obtained during enrollment) and a query (obtained during verification), decides whether the query is authentic by measuring its distance from the template. A less well studied application extracts a sequence of bits from the template, and such bits

* This is a revised version of the published paper in the ICB proceeding with these typo errors corrected: changing occurrences of "0.09%" to "0.09".

are then served as a secret key in other cryptographic operations. For example, one may use the secret bits to encrypt a file. During decryption of the file, the *exactly same* sequence needs to be extracted from the query. Note that when used as a key in cryptographic applications, the representative sequence has a stronger requirement, that their match must be *exact*, i.e., two similar minutiae must map to the exactly same bit sequence. This requirement is much more restrictive than the requirement of similarity (not necessarily exact match) between the minutiae, as in typical authentication systems. In this paper, our goal is to extract consistent bits from minutiae.

In order to achieve consistency, a technique, known as fuzzy commitment[9], secure sketch[7], shielding function[10] and helper data[13], extracts an additional information from the template during enrollment. We follow notations by Dodis et al. [7] and call the additional information a sketch. The sketch has to be made public. In the example on encryption, the sketch is to be stored in clear in the header information of the encrypted file. During verification, the sketch is employed to remove the noise, in a similar role as the parity check bits of an error correcting code. Since the sketch is published in clear, it has to be secure in the sense that it does not reveal too much about the original template.

Relationship with traditional authentication system. Note that a system which is able to extract consistent bits, can be adopted (although not preferable) to serve the purpose of a typical biometric authentication system. This is done by simply using the bits as the template: two templates are declared to be a match iff their extracted bits are exactly the same. Hence, if the systems is able to extract m consistent bits, the false match rate (FMR) of the corresponding authentication system is 2^{-m}. Since biometric authentication system is relatively more extensively studied, the performance of bit extraction system is unlikely to out-perform the state-of-the art biometric authentication system in terms of false match rate (FMR) and false non-match rate (FNMR). In other words, if the state-of-the-art authentication system achieves FMR of 0.1% at a particular FNMR, it is unlikely that we can extract more than 10 bits(with the same FNMR). Although 10 bits is too small for cryptographic operations and thus seems pessimistic, we could employ multi-modals to increase the bits size. Some previous works claim to extract significantly more bits from a single fingerprint. However, these claims may be based on simplified noise models, or the claims may not consider information leaked by the sketch and information leaked during alignment. The relationship between the FMR with the bits size is also discussed by Buhan et al. [1] from another perspective.

Challenging issues in sketch construction. The design of a secure sketch heavily depends on the notion of similarity between templates, which is related to the associated underlying metric space. Although there are near optimal constructions for some metrics like Hamming distance for binary strings [9, 7], and set-difference for sets [8, 7, 2], extending such constructions to the more "complicated" metric for minutiae is not straightforward. This is because the sketch

has to handle both white noise, which corresponds to Hamming distance, and replacement noise, which corresponds to set-difference [6, 3].

Clancy et al. [6] proposed a sketch construction for fingerprint that uses *chaff points*. Essentially, a large set of random points (known as the chaff points) is first generated, and the union of the chaff points and the minutiae is the sketch. The chaff points-based approach is adopted in a few other schemes [12, 13, 3]. However, it is very difficult to analyze the security achieved by this approach. Without rigorous analysis, it is not clear how many secret bits can be extracted. Indeed, an attack is proposed by Chang et al. [4] which demonstrates that the information revealed by the sketch is more than what was previously estimated. Although there are constructions with provable bounds on entropy loss[3] (based on formulation by Dodis et. al.[7]), it is not clear how effective are those constructions in practice. Furthermore, although the original minutiae are hidden, specific important information is revealed. For example, a point that does not appear in the sketch is definitely not in the original.

Besides the difficulty in analyzing the security of chaff points-based approach, there are also a few practical issues. Firstly, the sketch should be small so as to be fitted into mobile devices like smart card. Chaff points-based methods give large sketch since the size has a tradeoff with the security. Secondly, since the template is not available during verification, it is not clear how to align the query. Earlier works typically assumed that the fingerprints are already aligned with the exception of some recent works by Uludag et al. [13], which employ orientation field flow curves for alignment. Thirdly, previous works assume simplified models of noise and distribution of the minutiae. It is not clear how to fine-tune chaff generation for different noise models, and different statistical models of the minutiae.

Main idea in our construction. To handle the above mentioned issues, we propose applying a locality-preserving transformation[3] on the minutiae to a vector space of real coefficients. The function is locality preserving in the sense that two close-by data in the original metric space remains close to each other in the transformed space. Besides being locality-preserving, the function should decorrelate the original and give mutually independent real coefficients. We decorrelate the original by applying Principle Component Analysis so that the coefficients are pair-wise independent.

Note that we do not propose a new representation for fingerprint. Instead, we apply a transformation that preserves distance information of a well-accepted representation for fingerprints, and then extract bits from the transformed data.

Below are some advantages of the proposed method.

1. The transformation provides a way to map the minutiae to a metric space, whereby sketch can be easily constructed. It is fairly easy to adapt our transformation to different variants of minutiae representation. For example, if the orientation of each minutia is to be included, it is easy to modify the transformation to accustom this variation.

[3] Also known as biometric embedding[7]

2. The transformation "diffuses" certain information on each individual minutia. In particular, some important specific information, for e.g. total number of minutiae, possible locations of the minutiae etc, are diffused and not leaked out.
3. Our method is able to handle a wider class of noise models. As we shall see later, we use a maximum likelihood decoding to search for the the secret bits. As long as the noise model is able to facilitate maximum likelihood decoding, it can be incorporated.
4. Our method can also handle a more general model of minutiae distribution. This is achieved by applying PCA in the transformed domain.
5. As opposed to the chaff points-based method, no randomness is injected during the sketch construction. As a result, the size of our secure sketch is small.

Our method includes registration of fingerprints. This is achieved by including singular (core/delta) points[5] in the final sketch. These points carry global information and are independent of the minutiae, and thus unlikely to reveal any information of the minutiae, whereby the secret bits are extracted.

Our experimental data set consists of 2000 pairs of fingerprints from the NIST 4 database, using 100 pairs as training data and the rest as test data. Experimental studies show that the method can extract 10 consistent bits with FNMR of 0.09 and the total sketch size is around 320 bits. Hence, if this bit extraction system is adopted to serve in the traditional authentication system, its FNMR is 0.09 and FMR is 0.1%, which is close to typical authentication systems.

2 Proposed Method

2.1 Enrollment/Sketch construction

Step 1. Extracting minutiae and singular points. Given a fingerprint, the set of minutiae $\mathbf{x}$ is extracted. Next, the singular (core and delta) points of the fingerprint are extracted using the complex filter based method proposed by Nilsson et al. [11].

Step 2. Locality Preserving Hash. Herein the set of minutiae $\mathbf{x}$ are mapped to a vector $\mathbf{v}$ in $\mathbb{R}^k$. This step consists of two transformations, M_1 and M_2, where M_1 maps the minutiae to a real vector, and M_2 de-correlates the vector and keeps only k coefficients. The final output is

$$\mathbf{h_x} = M_2(M_1(\mathbf{x})).$$

The transformations M_2, M_1 and the parameter k are chosen during the design stage. First, a set $\mathcal{L}$ of 2D lines are selected, where$|\mathcal{L}| = q$ is some integer greater than k. Given the set $\mathbf{x}$, and a line $\ell \in \mathcal{L}$, we can obtain an integer which is the difference of number of points on the two sides of the line. Since there

are q lines in $\mathcal{L}$, given $\mathbf{x}$, we obtain q integers. Let $\mathbf{v} = (v_1, v_2, \ldots, v_q)$ where v_i is the integer obtained corresponding to the i-th line in $\mathcal{L}$. Let M_1 denote this transformation, that is $\mathbf{v} = M_1(\mathbf{x})$.

Using knowledge of the statistical distribution of $\mathbf{v}$ (derived from a large collection of fingerprints), Principle Component Analysis (PCA) can be carried out to derive a linear transformation that de-correlate the coefficients in $\mathbf{v}$. By keeping only k decorrelated coefficients, we obtain the linear transformation M_2.

The q lines in $\mathcal{L}$ are randomly chosen during the design stage. One could choose the lines with certain properties, for example, equally spaced horizontal and vertical lines. One could also choose much more lines, i.e., using a larger q. Nevertheless, experiments suggest that the final performance is similar. The prior knowledge of the statistical distribution can be obtained from a database of samples. For example, the NIST fingerprint database [14] provides ample fingerprints to estimate the distribution of $M_1(\mathbf{x})$.

Step 3. Convert $\mathbf{h_x}$ *to bits.* The sequence $\mathbf{h} = (h_1, h_2, \ldots, h_k)$ is converted to k bits $\mathbf{b} = (b_1, b_2, \ldots, b_k)$, where $b_i = 0$ iff $h_i < 0$, for each i.

Note that the PCA ensures that coefficients of $\mathbf{h_x}$ are pairwise uncorrelated. Hence we assume that the k-bits $\mathbf{b}$ are uniformly distributed, and its entropy is k.

Step 4. Extract consistent bits and secure sketch. This step requires a codebook $\mathcal{C} = \{\mathbf{C}_i\}_{i=1}^{2^m}$ where each $\mathbf{C}_i$ is a k-bits string and m is a parameter to be decided during the design stage. The message associated to the codeword $\mathbf{C}_i$ is its index i. Given the k-bits sequence $\mathbf{b_x}$, its nearest codeword $\mathbf{c_r}$, with respect to Hamming distance, is determined. The message associated to $\mathbf{c}_r$ is output as the consistent bits. The sketch is the bit sequence

$$s_{\mathbf{x}} = \mathbf{b_x} \oplus \mathbf{c}_r,$$

where $\oplus$ is the xor operations.

The codebook is determined during the design stage, and m an important parameter needs to be determined. In our implementation, in order to facilitate experimental studies for different values of k and m, we use a random codebook. That is, the codewords are randomly chosen during the design stage. To improve the performance, for a particular k and m, a good error correcting code can be used as the codebook. The use of xor operation here seems abrupt. This is a common technique in sketch construction for binary strings with Hamming distance as the underlying metric[7, 9]

The message can be represented as a m-bits string. It is not necessary that the entropy of the message is m. We will discuss this further in Section 3. Alternatively, instead of taking the message as the secret, we can also use $\mathbf{c}_r$ as the secret bits. However, there is high redundancy in the k-bits $\mathbf{c}_r$.

Step 5. Publish the final sketch. The final sketch consists of the sketch $s_{\mathbf{x}}$, and the singular (core and delta) points information. The final sketch is then made public, for example, by storing it in the header of the encrypted file.

2.2 Verification

Now, given a query fingerprint and the final sketch (recall that the final sketch consists of the singular points, and the sketch $s_{\mathbf{x}}$, we want to extract the consistent bits.

Step 1. Alignment. Same as Step 1 in enrollment, the minutiae $\mathbf{y}$ and singular points of the query are extracted. Next, using the extracted singular points, and the singular points in the final sketch, alignment is carried out.

Step 2 & 3. Obtain $\mathbf{h_y}$ *and* $\mathbf{b_y}$. The same steps (Step 2 & 3) in the enrollment are carried out to obtain the k-bits $\mathbf{b_y}$.

Step 4. Maximum Likelihood Decoding. Together with the sketch $s_{\mathbf{x}}$, compute

$$\widetilde{\mathbf{b}} = \mathbf{b_y} \oplus s_{\mathbf{x}}.$$

Next, using maximum likelihood decoding, find the most likely codeword $\mathbf{c}_r$ that gives $\widetilde{\mathbf{b}}$ (the method is to be described below). The message corresponding to $\mathbf{c}_r$ is the consistent bits.

Given the 100 pairs of fingerprints (original $\mathbf{x}$ and noisy version $\mathbf{y}$) as training data, we compute the k-bits $\mathbf{b_x}$ and $\mathbf{b_y}$. Next, every pair is compared to note the corresponding bit flip for every bit. This gives an estimate of the probability p_j that the j-th bit flips under noise. During maximum likelihood decoding, for each codeword $\mathbf{C}_i = (c_1, c_2, \ldots, c_k)$, we estimate the probability P_i that $\widetilde{\mathbf{b}}_i = (\widetilde{b_1}, \widetilde{b_2}, \ldots, \widetilde{b_k})$ is a noisy version of the codeword.

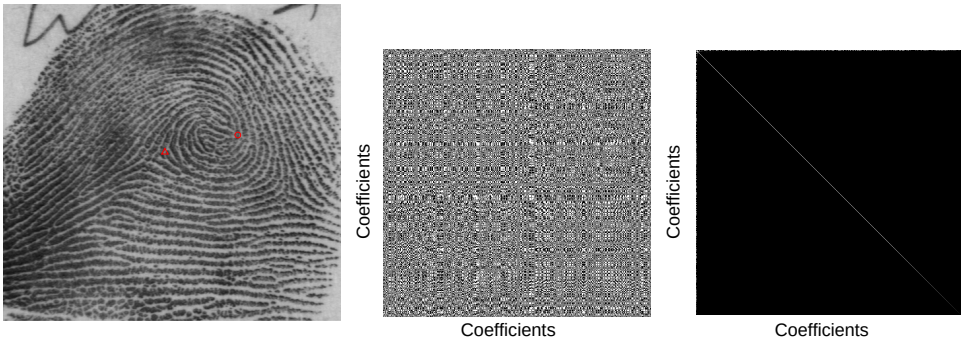
$$P_i = \prod_{j=1}^{k} p_j^{\widetilde{b}_j \oplus c_j} (1 - p_j)^{(1-\widetilde{b}_j \oplus c_j)}.$$

The codeword with largest P_i is chosen as the most likely codeword.

2.3 Remark

In sum, during the design stage, we determine a set of lines $\mathcal{L}$, and perform PCA to obtain two transformations M_1 and M_2. A codebook $\mathcal{C}$, and the probability of bit flip p_j is also determined during the design stage. There are two important parameters: k, the number of transformed bits, and $m = \log_2 |\mathcal{C}|$, where $|\mathcal{C}|$ is the number of codewords.

There is a subtle difference between enrollment and verification. During enrollment, the nearest word is determined with respect to the usual Hamming distance. However, during verification, we employ maximum likelihood decoding, which is essentially finding the nearest code with respect to a weighted Hamming distance.



(a) (b) (c)

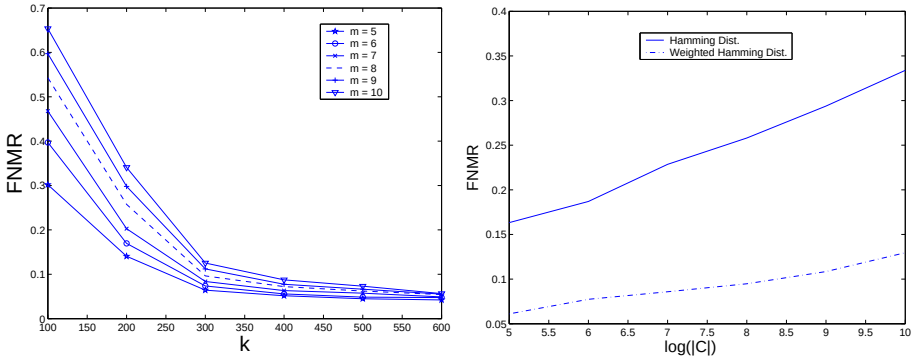
Fig. 1. (a) Singular point (core and delta) detection results for a fingerprint from the NIST database. (b) Covariance matrix of coefficient vectors (for 1900 fingerprint pairs of dimension 600) before applying PCA (c) Covariance matrix of coefficient vectors (for 1900 fingerprint pairs of dimension 600) after applying PCA.

3 Experiments and Analysis

Parameters The performance of our method is evaluated based on the tradeoff that can be achieved by varying the two parameters. (1) k, the number of PCA coefficients retained, and (2) $|\mathcal{C}|$ the size of the codebook needed. For each set of parameters, we measure (1) the entropy of the sketch, denoted by $|s|$ and (2) FNMR of the system.

Recall that we assume that the k-bits $\mathbf{b}$ are uniformly distributed. If the entropy of the sketch is $|s|$, then the entropy of the consistent bits will be $k-|s|$. However, it is not easy to estimate the sketch entropy. In this paper, we use a conservative estimate (i.e. a upper bound). Thus $k-|s|$ could underestimate the number of consistent bits. From another perspective, the sketch is essentially the syndrome with respect to the error correcting code. If we further assume that the syndrome is independent from the message, then $\log_2 |\mathcal{C}|$ is the number of consistent bits.

Data Set and Experimental setup. Experiments were conducted on a database of 4000 fingerprints from the NIST fingerprint database[14], which consists of two scans of 2000 fingerprints. 100 fingerprint pairs were taken as training data for modeling the noise and distribution of minutiae. The remaining 1900 pairs were left as test data. Minutiae were extracted from the image using *mindtct*, a minutia extraction software provided with the NIST package. The extracted minutiae information consists of the 2D coordinates, the orientation and the quality of the minutiae. Only high quality minutiae were selected based on a threshold. The final set consisted of around 50 to 60 minutiae, and it would require at least 800 bits to represent the minutiae.



(a) (b)

Fig. 2. (a) FNMR vs. k, for different values of $log_2(|\mathcal{C}|)$. (b) FNMR vs. k under hamming and weighted hamming distance. Illustrates the efficacy of using knowledge of noise is designing a distance function.

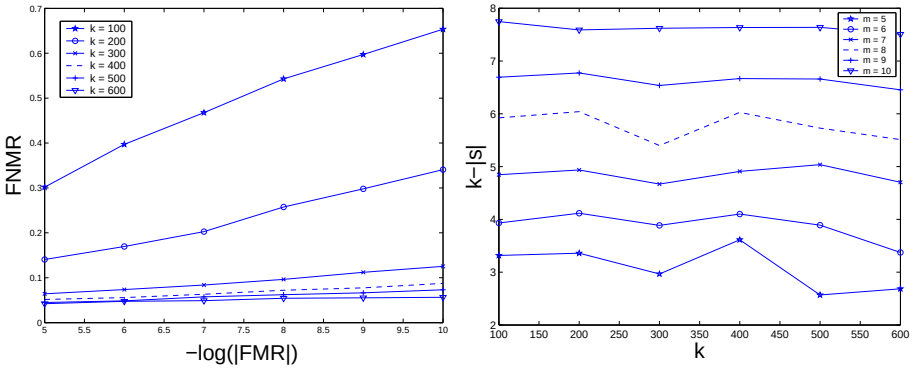
The singular points for alignment is extracted using complex filters filters[11]. Figure 1(a) depicts the singular points detected in one of the fingerprint scans.

The sketch size $|s|$ is estimated by first counting the number of 1's among the sketches $s_{\mathbf{x}}$ for all fingerprints in the test data. Now, we can estimate the probability that a bit in the sketch is 1, which in turn give the entropy of the k-bit $s_{\mathbf{x}}$.

Effect of PCA and the noise model. A total of $q = 600$ lines were chosen to perform locality preserving hash and then PCA is applied to generate 600 coefficients. The PCA attempts to decorrelate the coefficients and make them pairwise independent so that the bits in $\mathbf{b_x}$ would be uncorrelated. Figure 1(b)-(c) depicts an image of the covariance matrix of the coefficients for 1900×2 test fingerprints, before and after applying PCA. The high intensity along the diagonal of the covariance matrix image in Figure 1(c) is indicative of the statistical independence of the coefficients. The low intensity of the off-diagonal elements indicate low correlation.

Figure 2(b) illustrates the affect of weighted hamming distance and hamming distance on FNMR. Clearly, from the graph, using the noise model derived from the training data improves the FNMR.

Relationship among different parameters. Fig. 2 (a) shows the FNMR for different k and $|\mathcal{C}|$. As expected, FNMR reduces with larger k. This is expected because using more bits should preserve more distance information. FNMR also reduces with lesser codewords in $\mathcal{C}$. This is because with lesser codewords, the distances among them increase and thus more tolerance to noise. A line drawn parallel to the x-axis gives the number of coefficients k, that need to be retained for a given requirement on FNMR and $log_2|\mathcal{C}|$.



(a) (b)

Fig. 3. (a) FNMR vs. $log_2(|\text{FMR}|)$ for different values of k. (b) $k - |s|$ vs. k. Since $|s|$ may be overestimated, this gives a lower bound on the number of consistent bits.

Figure 3(a) depicts the change in FNMR with $log_2(|\text{FMR}|)$ for different sizes of k. Note that for a 10 bit consistent key (i.e., codebook of size 2^{10}) for $k = 300$ bits, FNMR=0.09 (~ 0.1), which is very close to the FNMR obtained in typical biometric authentication systems. Figure 3 (b) depicts the change in sketch size $|s|$ for different k and $log_2|\mathcal{C}|$. An straight line draw parallel to the y-axis gives the sketch size for a given constraint on k and $|\mathcal{C}|$.

An observation is that to extract more bits, the size of the sketch $|s|$ has to be higher otherwise it leads to high FNMR. When k is around 300 bits we expect to be able to extract 10 bits with FNMR ~ 0.1, which is around the FNMR of typical authentication systems. For larger values of k, the FNMR does not improve significantly, hence 300 is a good tradeoff.

4 Conclusion

In this paper, a method for generating consistent bits from minutiae is proposed. Such bit sequences can be used as secret keys in cryptographic operations that require the exact sequence from different scans. Compared to known bit extraction methods, our proposed method allows for registration of the fingerprints using information (singular points) that is independent of the secret and thus does not leak any information about the minutiae. A locality preserving transformation followed by a PCA is performed on the minutiae to generate a binary sequence. Although some information is thrown away during the transformation, some of the obvious advantages of the method are that information is diffused and the binary sequence generated is robust. Unlike existing techniques, no randomness is injected during sketch construction and thus the sketch size is small. We show the use of a maximum-likelihood based distance measure for decoding that can

incorporate different noise models. Experimental results verify the efficacy of our proposed method.

References

1. I. Buhan, J. Doumen, P. H. Hartel, and R. N. J. Veldhuis. Fuzzy extractors for continuous distributions. In *ASIACCS*, pages 353–355, 2007.
2. E.-C. Chang, V. Fedyukovych, and Q. Li. Secure sketch for multi-sets. *Cryptology ePrint Archive*, Report 2006/090, 2006.
3. Ee-Chien Chang and Qiming Li. Hiding secret points amongst chaff. In *Eurocrypt*, 2006.
4. Ee-Chien Chang, Ren Shen, and Weijian Teo. Finding the original point set hideen among chaff. In *Proc. ACM Sym on Information, Computer and Communications Security*, 2006.
5. S. Chikkerur and N. Ratha. Impact of singular point detection on fingerprint matching performance. In *IEEE AUTOID*, 2005.
6. T. Charles Clancy, Negar Kiyavash, and Dennis J. Lin. Secure smartcardbased fingerprint authentication. In *ACM SIGMM workshop on Biometrics methods and applications*, pages 45–52, 2003.
7. Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Eurocrypt'04*, pages 523–540, 2004.
8. Ari Juels and Madhu Sudan. A fuzzy vault scheme. In *IEEE Intl. Symp. on Information Theory*, 2002.
9. Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *ACM Conf. on Computer and Communications Security*, pages 28–36, 1999.
10. 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.
11. K. Nilsson. *Localization of corresponding points in fingerprints by complex filtering*. PhD thesis, Chalmers University of Technology, Sweden, 2005.
12. I.M. Verbauwhede S. Yang. Secure fuzzy vault based fingerprint verification system. In *38th Asilomar Conf. on Signals, Systems, and Computers*, volume 1, pages 577–581, 2004.
13. U. Uludag and A. Jain. Securing fingerprint template: fuzzy vault with helper data. In *Proc. IEEE Workshop on Privacy Research In Vision*, June, 2006.
14. C. Watson, M. Garris, E. Tabassi, C. Wilson, R. McCabe, and S. Janet. NIST fingerprint image software 2. *NIST Special Database*, 2006.