Practical Biometric Authentication with Template Protection

Pim Tuyls[1], Anton H.M. Akkermans[1], Tom A.M. Kevenaar[1], Geert-Jan Schrijen[1], Asker M. Bazen[2], and Raymond N.J. Veldhuis[2]

[1] Philips Research Laboratories
Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands
{pim.tuyls,ton.h.akkermans,tom.kevenaar,geert.jan.schrijen}@philips.com
[2] University of Twente - EEMCS - SAS
P.O. box 217, 7500 AE Enschede, The Netherlands
{a.m.bazen,r.n.j.veldhuis}@utwente.nl

Abstract. In this paper we show the feasibility of template protecting biometric authentication systems. In particular, we apply template protection schemes to fingerprint data. Therefore we first make a fixed length representation of the fingerprint data by applying Gabor filtering. Next we introduce the *reliable components* scheme. In order to make a binary representation of the fingerprint images we extract and then quantize during the enrollment phase the reliable components with the highest signal to noise ratio. Finally, error correction coding is applied to the binary representation. It is shown that the scheme achieves an EER of approximately 4.2% with secret length of 40 bits in experiments.

1 Introduction

Biometrics identify/authenticate people on what they are rather than on what they have (tokens) or what they know (passwords). Since biometric properties can not be lost or forgotten in contrast to tokens and passwords, they offer an attractive and convenient alternative to identify and authenticate people.

When the reference information, captured during the enrollment phase, is not properly protected some privacy problems arise. The main risks are given by: i) Biometrics contain sensitive information about people [Bolling, P65]. ii) Once compromised, the templates are compromised forever and can not be reissued [S99]. iii) Biometric data stored without protection can be used to perform cross-matching between databases and track peoples behaviour. iv) Many biometric identifiers can be forged based on template information [MMJ03]. This problem received recently a lot of attention [JS02, TG04, LT03, DRS04, JW99, Sou98].

Two equivalent approaches, Helper Data and Fuzzy Extractors, were proposed to solve this privacy problem [TG04, DRS04]. In these papers the theory of template protection has been developed and some algorithms were proposed. In [TVI04] these algorithms were applied to ear identification and a satisfactory performance (EER=3%, secret length=100) was achieved.

T. Kanade, A. Jain, and N.K. Ratha (Eds.): AVBPA 2005, LNCS 3546, pp. 436–446, 2005.

In this paper, we present an implementation of template protection for fingerprint based authentication. We present an algorithm based on helper data consisting of two parts. The first part identifies the *reliable* components with a high signal to noise ratio in the analog picture of a Gabor-filtered fingerprint. By applying quantization, a binary representation is made of the fingerprint. The second part of the helper data maps the binary representation onto a code word of an error-correcting code which is further used to correct the noise remaining after quantization.

2 Preliminaries

2.1 Biometric Verification

The biometric system that is considered in this paper is a *verification system*. As usual it consists of two phases. In the enrollment phase (executed at a Certification Authority (CA)), reference measurements are taken, the features are extracted, and the template is stored in e.g. a database in a properly protected way. During the verification phase, a live biometric measurement is compared to the template that is retrieved from the database using a claimed identity. Due to noise (caused by scratches, weather conditions, partial impressions, elastic deformations, etc.) the measurements taken during the enrollment and verification phase are different. This degrades the performance of a biometric verification system. In order to measure the performance, two different error rates are commonly used. The False Acceptance Rate (FAR) is the probability that an impostor is falsely accepted as a genuine user. The False Rejection Rate (FRR) is the probability that a genuine user is falsely rejected by the system. The Equal Error Rate (EER) is the error rate at the point of operation where FAR is equal to FRR.

2.2 Template Protection

Biometric data (and their extracted features) are modeled as k-dimensional random variables with entries in $\mathbb{R}$. The extracted features during the enrollment phase are denoted by $\boldsymbol{X}$ and those extracted during the verification phase by $\boldsymbol{X'}$. The data during the verification phase are modeled as a noisy version from those measured during the enrollment phase [TG04].

The core algorithm of a template protecting biometric system extracts a *secret* from the biometric data. Generally speaking such an algorithm is built on a Secret Extraction Code [TG04] or equivalently a Fuzzy Extractor [DRS04]. For the sake of simplicity we describe the algorithm in terms of a shielding function [LT03], which generates a special set of secret extraction codes [TG04] but has all necessary properties. A shielding function $G : \mathbb{R}^k \times \{0,1\}^k \to \{0,1\}^K$ extracts a secret of length K from the biometric as follows. Given a randomly chosen secret $S \in \{0,1\}^K$ and a biometric $\boldsymbol{X} \in \mathbb{R}^k$, *helper data* $W \in \{0,1\}^k$ is computed such that $G(X,W) = S$ (equivalently the equation $G(X,W) = S$ is solved for W). A shielding function is called δ-contracting if for all $\boldsymbol{X'}$ that lie

within a ball of radius δ of $\boldsymbol{X}$ we have $G(\boldsymbol{X}', W) = G(\boldsymbol{X}, W) = S$. The function G is called ϵ-revealing if the helper data W leaks less than ϵ bits on S (in the information theoretic sense), i.e. $\mathbf{I}(W; S) \leq \epsilon$. It is the goal to design the system such that W leaks also a minimal amount of information on $\boldsymbol{X}$; i.e. $\mathbf{I}(W; \boldsymbol{X})$ has to be minimized. It was shown in [LT03] that for a shielding function G, $\mathbf{I}(W; \boldsymbol{X})$ can not be made equal to zero.

During the *enrollment* phase the features $\boldsymbol{X}$ of Alice's biometric are extracted, a secret S is randomly choosen and the helper data W is computed. Then, a one-way hash function H is applied to S and the data $(Alice, W, H(S))$ is stored in a database.

During the *verification* phase, (at the sensor) a noisy version $\boldsymbol{X}'$ of Alice's biometric $\boldsymbol{X}$ is measured. When Alice claims her identity the helper data W is passed onto the sensor. The sensor computes $S' = G(\boldsymbol{X}', W)$ and $H(S')$. At the database $H(S')$ is compared to $H(S)$. If both are equal access is granted and if they are unequal no access is granted. Note that in contrast to usual practice in biometrics ("fuzzy matching") an exact match is performed. We stress that the helper data is sent over a public channel, i.e. W can be captured by an attacker. The system is however designed such that the knowledge of W provides a minimal amount of information on $\boldsymbol{X}$ and S [LT03, TG04, DRS04]. For basic examples of template protecting biometric verification systems, we refer to [TG04, DRS04].

2.3 Fingerprint Feature Extraction

In this section we present a fixed length feature vector representation, of which the elements can be compared one by one directly. The selected feature vector describes the global shape of the fingerprint by means of the local orientations of the ridge lines.

In order to allow for direct comparison of the feature vectors, without requiring a registration stage during matching, the feature vectors have to be prealigned during feature extraction. For this purpose, the core point (i.e. the uppermost point of the innermost curving ridge) is used. These core points are automatically extracted using a likelihood ratio-based algorithm that is described in [Baz04].

To describe the shape of the fingerprint, we extract two types of feature vectors from the gray scale fingerprint images. The first feature vector is the squared directional field that is defined in [Baz02]. It is evaluated at a regular grid of 16 by 16 points with spacings of 8 pixels, which is centered at the core point. At each of the 256 positions, the squared directional field is coded in a vector of two elements, representing the x- and y-values, resulting in a 512-dimensional feature vector. An example fingerprint and its directional field are shown in Figures 1a and 1b respectively.

The second feature vector is the Gabor response of the fingerprint, which is discussed in [BV04]. After substraction of the spatial local mean, the fingerprint image is filtered by a set of four complex Gabor filters, which are given by:

$$h_{\text{Gabor}}(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \cdot \exp\left(j2\pi f \cdot (x \sin\theta + y \cos\theta)\right) \qquad (1)$$

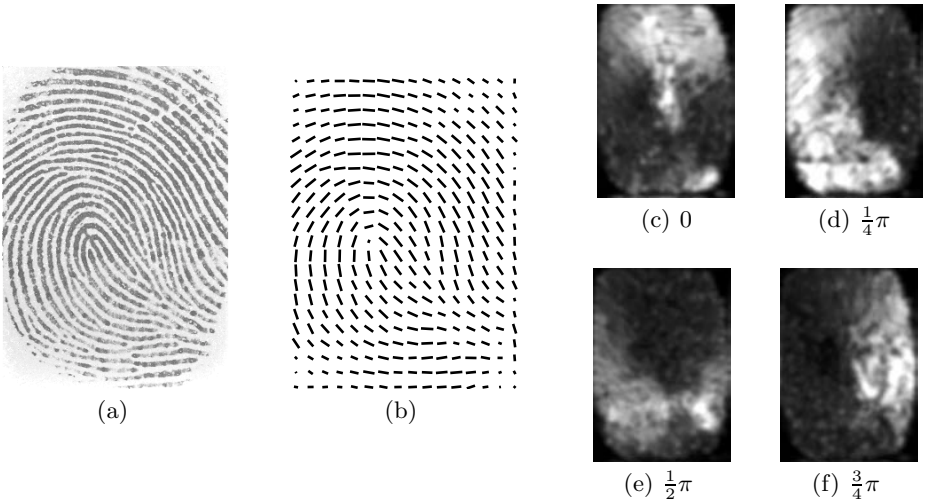

Fig. 1. (a) Fingerprint image, (b) its directional field and (c)-(f) the smoothened absolute values of Gabor responses for different orientations θ

The orientations θ are set to 0, $\pi/4$, $\pi/2$, and $3\pi/4$, the spatial frequency f is tuned to the average spatial ridge-valley frequency ($f = 0.11$), and the width of the filter σ is set such that the entire orientation range is covered ($\sigma = 3.5$). The absolute values of the output images are taken, which are subsequently filtered by a low-pass Gaussian window. The resulting images are shown in Figures 1c to 1f.

Again, samples are taken at a regular grid of 16 by 16 points with spacings of 8 pixels and centered at the core point. The resulting feature vector is of length 1024. This feature vector is inspired by FingerCode [Jai00], but it can be calculated more efficiently since a rectangular grid is used rather than a circular one, and it performs better.

The resulting feature vector that is used for matching is a concatenation of the squared directional field and the Gabor response. It describes the global shape of the fingerprint in 1536 elements.

3 Integration of Template Protection with Fingerprint Verification

From each user we use M measurements of his/her biometric for enrollment. The enrollment phase comprises five steps: *Feature Extraction*, *Statistical Analysis*, *Quantization*, *Selecting Reliable Components* and *Creating Helper Data*. These steps are described in detail in Section 3.1.

In the verification phase the biometric of a user is measured. Then, feature extraction and quantization are performed and using the helper data the noise is removed and the secret reconstructed. The details are explained in section 3.2. The complete scheme is shown in Fig. 2.

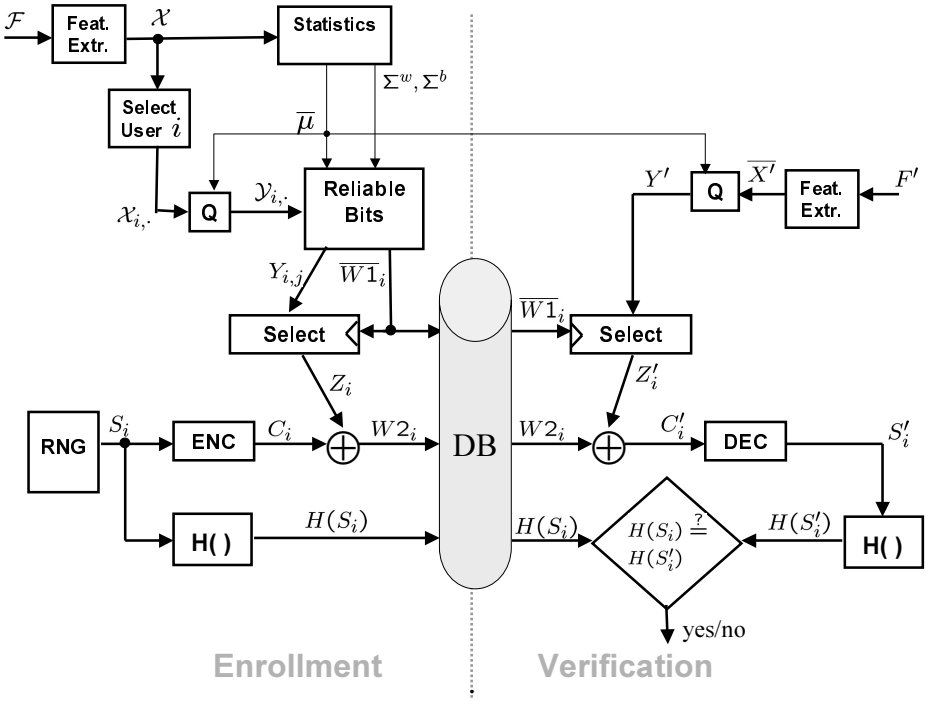

Fig. 2. Overview of the reliable components scheme

3.1 Enrollment

Feature Extraction. At the input of the scheme, we consider a set of biometric enrollment measurements $\mathcal{F} = \{F_{i,j}\}_{i=1..N,j=1..M}$ where a subscript i, j denotes the j-th enrollment measurement of the i-th user. Thus N is the number of users and M the number of enrollment measurements per user such that $\mathcal{F}$ consists of NM digital images of fingerprints. In the Feature Extraction block (depicted as 'Feat. Extr.' in Fig. 2) feature vectors $\boldsymbol{X}$ are extracted from these images, according to the method described in Section 2.3. The set of NM feature vectors is denoted as $\mathcal{X} = \{\boldsymbol{X}_{i,j}\}_{i=1..N,j=1..M}$, where $\boldsymbol{X}_{i,j} \in \mathbb{R}^k$ denotes the j-th feature vector of the i-th person with components $(\boldsymbol{X}_{i,j})_t$ where $t = 1 \ldots k$.

Statistical Analysis. Firstly, we compute the estimated mean feature vector $\boldsymbol{\mu}_i$ of person i and the mean $\boldsymbol{\mu}$ of all enrollment feature vectors as follows,

$$\boldsymbol{\mu}_i = \frac{1}{M}\sum_{j=1}^{M} \boldsymbol{X}_{i,j} \ , \quad \boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\mu}_i \quad . \tag{2}$$

Secondly, we compute estimates of the within-class covariance matrix Σ^w and the between-class covariance matrix Σ^b,

$$\Sigma^w = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(\boldsymbol{X}_{i,j}-\boldsymbol{\mu}_i)(\boldsymbol{X}_{i,j}-\boldsymbol{\mu}_i)^T \ , \ \ \Sigma^b = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{\mu}_i-\boldsymbol{\mu})(\boldsymbol{\mu}_i-\boldsymbol{\mu})^T. \tag{3}$$

Quantization. In Fig. 2, the quantization block is denoted by 'Q'. In this block a binary representation (bit string) is derived from the input feature vectors of person i denoted as $\mathcal{X}_i = \{\boldsymbol{X}_{i,j}\}_{j=1\ldots M}$. The 'Select User i' block in Fig. 2 selects these feature vectors from the total set $\mathcal{X}$. The quantization of $\mathcal{X}_i$ is based on the mean $\boldsymbol{\mu}$ [1] determined in the 'Statistical Analysis' block. A binary string $Q(\boldsymbol{X}_{i,j})$ is constructed from the feature vector $\boldsymbol{X}_{i,j}$ where each bit $(Q(\boldsymbol{X}_{i,j}))_t$ is defined as (for $t \in \{1, \ldots, k\}$)

$$(Q(\boldsymbol{X}_{i,j}))_t = \begin{cases} 0 & \text{if } (\boldsymbol{X}_{i,j})_t \leq (\boldsymbol{\mu})_t \\ 1 & \text{if } (\boldsymbol{X}_{i,j})_t > (\boldsymbol{\mu})_t \ . \end{cases} \tag{4}$$

Selecting Reliable Components. In this step we look for the reliable components in the M bit strings $Q(\mathcal{X}_i) = \{Q(\boldsymbol{X}_{i,j})\}_{j=1\ldots M}$ of user i. The block 'Reliable Bits' of Fig. 2 determines the K most reliable components (or bits) for user i and creates a first set of helper data $\boldsymbol{W1}_i$. K is a fixed parameter[2] that matches the length of the codewords that are going to be used in the 'Creating Helper Data' step. The reliable components are defined as follows.

The t-th component of $Q(\boldsymbol{X}_{i,j})$ for a fixed user $i = 1, \ldots, N$ is called *reliable*, if the values $(Q(\boldsymbol{X}_{i,j}))_t$ for $j = 1 \ldots M$ are *all equal*. The boolean vector $\boldsymbol{B}_i \in \{0,1\}^k$ denotes the reliable bits. Its t-th entry equals one if the t-th component of $Q(\boldsymbol{X}_{i,j})$ is reliable otherwise the t-th entry is zero. For user that have less than K reliable components, we additionally define *soft reliable* components. Define *p-soft reliable components* of user i as the values t for which $M - p$ of the values $(Q(\boldsymbol{X}_{i,j}))_t$ for $j = 1 \ldots M$ are equal. The boolean vector $\boldsymbol{B}_i^{(p)} \in \{0,1\}^k$ denotes these p-soft reliable bits.

Creating Helper Data. The helper data of our scheme consists of two parts. The first part, denoted by the vector $\boldsymbol{W1}$ is determined as follows. We define the Signal-to-Noise Ratio vector $\boldsymbol{\xi} \in \mathbb{R}^k$ by the following equation,

$$(\boldsymbol{\xi})_t = \frac{(\Sigma^b)_{t,t}}{(\Sigma^w)_{t,t}} \quad , \quad t \in \{1, \ldots, k\}. \tag{5}$$

1. For each user i we determine the K most reliable components with the highest Signal-to-Noise Ratio based on the vectors $\boldsymbol{\xi}$, $\boldsymbol{B_i}$ and $\boldsymbol{B_i^{(p)}}$: first the reliable components (indicated by $\boldsymbol{B_i}$) with the highest $\boldsymbol{\xi}_t$ value are chosen. If the chosen amount of components is less than K, the p-soft reliable components with the highest $\boldsymbol{\xi}_t$ value are added (for successively $p = 1, 2, \ldots$) until a total amount of K components is chosen. The positions of these chosen components are stored in the vector $\boldsymbol{W1}_i \in \mathbb{N}^K$.

[1] Instead of the mean, the median can be used too. This leads to the same results

[2] The value of K is chosen in such a way that the vast majority of users have more than K reliable components

2. For each user i, we select the bits indicated by helper data $\boldsymbol{W1}_i$ and combine these bits into a new vector Z_i. More precisely, $(Z_i)_t = (Q(\boldsymbol{X}_{i,j}))_{(\boldsymbol{W1}_i)_t}$. (This step corresponds to the 'Select' box in Fig. 2).
3. Let $\mathcal{C}$ be an ECC [3] with parameters (K, s, d) where K denotes the length of the code words, s the number of information symbols and d the number of errors that can be corrected. For each user i, a secret $S_i \in \{0,1\}^s$ is randomly choosen [4] and encoded into the codeword $C_i \in \mathcal{C}$. The second part of the helper data $W2_i$ is then given by $W2_i = C_i \oplus Z_i$ (where $\oplus$ stands for bitwise XOR).

Finally the secret S_i is hashed using a cryptographic (one-way) hash function H and the values $\boldsymbol{W1}_i$, $W2_i$ and $H(S_i)$ are stored in the database (indicated with 'DB' in Fig. 2), linked to user i. Note that the secret size equals the number of information symbols s in the (K, s, d) code $\mathcal{C}$.

3.2 Verification

During the verification phase a noisy biometric F_i' of user i is measured. On F_i' the following computations are performed. i) Features are extracted from F_i' and a feature vector $\boldsymbol{X'}_i$ is obtained. ii) In the quantization block a bit string is derived by comparing the value of each component $(\boldsymbol{X'}_i)_t$ with the mean value $(\boldsymbol{\mu})_t$ according to Eq. 4 (where $\boldsymbol{X}_{i,j}$ is replaced by $\boldsymbol{X'}_i$ and $Q(\boldsymbol{X}_{i,j})$ is replaced by $Q(\boldsymbol{X'}_i)$). iii) The first helper data vector $\boldsymbol{W1}_i$ from the database is used to select K components from $Q(\boldsymbol{X'}_i)$ which yields a bit string Z_i'. iv) Then, $Z_i' \oplus W2_i = C_i \oplus (Z_i \oplus Z_i')$ is computed and the errors are corrected such that C_i' is obtained. v) Finally S_i' is obtained by decoding C_i' and $H(S_i')$ is compared to $H(S_i)$ stored in the database. If both values match, user i is authenticated.

4 Results

4.1 Fingerprint Databases

To compare the performance of the matching algorithms with and without template protection, we applied the recognition algorithms to two fingerprint databases.

i) The first fingerprint database we used is the second FVC2000 [Mai00] database. This database contains 8 images of 110 different fingers. The 8-bit gray scale fingerprint images were captured using a capacitive sensor with a resolution of 500 dpi. The image size is 256 by 364 pixels. We use six fingerprints per person during enrollment, two fingerprints per person during verification.

ii) The second fingerprint database is collected at the University of Twente using an optical digitalPersona U.are.U sensor. This database contains 5 images of 500 different fingers. The resolution of the images is 500 dpi, the bit-depth is 8 bit, and the image size is 452 by 492 pixels. We used 4 fingerprints per person during enrollment, and one fingerprint per person for verification.

[3] ECC stands for Error Correcting Code

[4] This is indicated by the Random Number Generator (RNG) block in Fig. 2

4.2 Classification Without Template Protection

For comparison we implemented a likelihood ratio-based verification scheme. For the first database this yields an $EER = 1.4\%$ and for the second database an $EER = 1.6\%$ [5] The results are shown in Fig. 3.

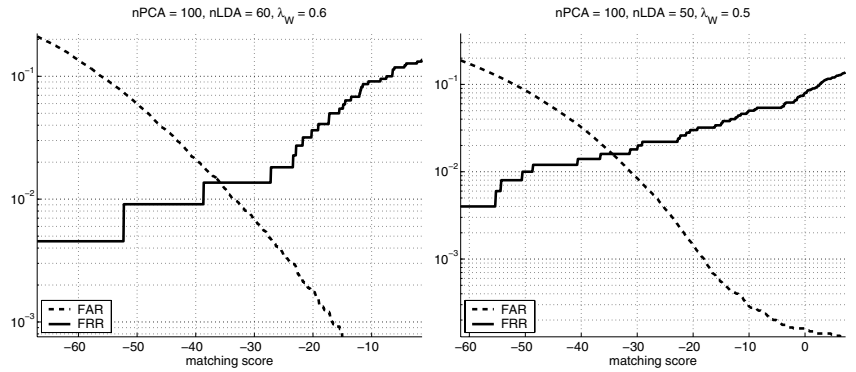

Fig. 3. Likelihood ratio-based results on databases 1 (left) and 2 (right)

4.3 Classification Results of the Reliable Component Scheme

In this section we give the results of the proposed Reliable Components Scheme for the databases described in Section 4.1.

The ECC we use is a binary BCH code described by the triplet (K, s, d). Since BCH codes do not exist for all triplets (K, s, d) we choose from the list of possible BCH codes the one that maximizes the performance. This choice is made as follows. For a set of test users, we investigate how the performance depends on the used BCH code. Fix K as explained in the enrollment procedure according to a valid BCH code. For our fingerprint databases, $K = 511$ is a good choice since the vast majority of users have more than 511 reliable components[6]. For this value of K consider the set of possible BCH codes $\mathcal{B}$ corresponding to all possible values of d (see also Figure 2).

i) For each possible value of d (according to K) choose a code $B(d)$ from $\mathcal{B}$.
ii) Perform enrollment i.e. determine $S, \boldsymbol{W}\mathbf{1}, W2$.
iii) Perform the verification phase and compute the FAR(d) and the FRR(d) for that value of d.

[5] For database 1, we used $n_{\mathrm{PCA}} = 100$, $n_{\mathrm{LDA}} = 60$, $\lambda_W = 0.5$ and threshold of -36. For database 2, we used $n_{\mathrm{PCA}} = 100$, $n_{\mathrm{LDA}} = 50$, $\lambda_w = 0.5$ and a threshold of -35. (n_{PCA},$n_{\mathrm{LDA}} = 50$ stand for the dimension after the PCA and LDA transformation respectively and λ_w is a regularization constant)

[6] On average a user has $\approx$ 800 reliable components. When selecting $K = 511$, on average 13 users will have less than 511 reliable bits (in both databases) and hence their helper data vector $\boldsymbol{W}\mathbf{1}_i$ also contains 1-soft reliable bits

As mentioned in section 4.1, we split the fingerprint database in a set of enrollment measurements and a set of verification measurements. The dependence of the FAR and the FRR on d is shown in Fig. 4 for one particular split. Clearly, when d is small the FAR will be small but the FRR will be rather high because the system is sensitive to noise. The results that we present here, are calculated by averaging over all possible splits: $\binom{8}{6} = 28$ different splits for database 1 and $\binom{5}{1} = 5$ splits for database 2. On average, the EER is achieved for $d \approx 86$ and $d \approx 102$ for database 1 and 2 respectively.

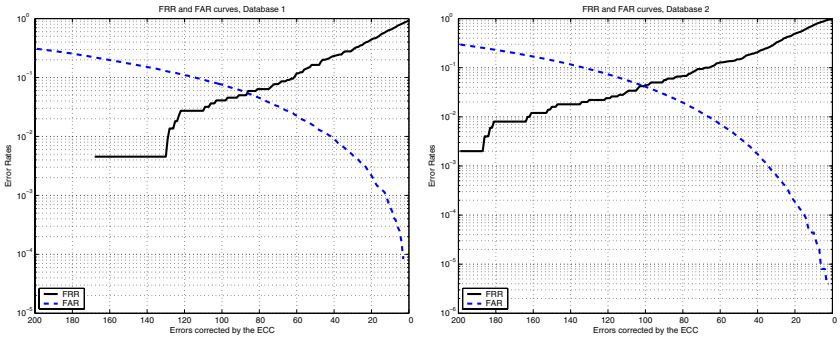

Fig. 4. Left: FAR and FRR as a function of d for database 1, where the training set consists of measurements $\{2, 3, 4, 5, 6, 7\}$ and the verification set of measurements $\{1, 8\}$. Right: FAR and FRR as a function of d for database 2, where the training set consists of measurements $\{1, 2, 3, 4\}$ and the verification set of measurement $\{5\}$

The BCH code that is closest to our (on average) required error correcting capability has parameters $(511, 76, 85)$ for database 1 and parameters $(511, 40, 95)$ for database 2. Fig. 5 summarizes the resulting FRR and FAR that can be achieved using these codes in columns 3 and 4. Furthermore the results for a few other codes (with error correcting capability close to the required average) are displayed. The figure shows that the average EER that can be achieved for database 1 is close to 5.3% and for database 2 around 4.5%. It turns out that many false acceptances and false rejections occur for certain people that have some low quality pictures in the original fingerprint database. For example, some

	ECC (K,s,d)	**FRR**	**FAR**	**FRR***	**FAR***
Database 1	(511,85,63)	0.099	0.025	0.069	0.029
	(511,76,85)	0.054	0.052	0.035	0.058
	(511,67,87)	0.052	0.055	0.034	0.061
Database 2	(511,49,93)	0.054	0.032	0.048	0.033
	(511,40,95)	0.054	0.035	0.048	0.036
	(511,31,109)	0.041	0.055	0.035	0.056

Fig. 5. Summary of the results for the two databases, for several selections of ECCs. (*): The columns FRR* and FAR* show the results if badly enrolled users are not taken into account

users have measurements where the core is on the edge of the picture or where no ridges can be distinguished. Users with such pictures amongst their enrollment data, will often have less than 511 reliable bits (and soft reliable bits are added). In a practical situation, these low quality pictures can easily be avoided during enrollment by visually checking the image quality of each enrollment measurement and repeating a measurement if the quality is too low. We tested this idea by leaving out users for which $\boldsymbol{W1}_i$ contains also *soft reliable* bits (see section 3.1). The results in terms of FAR and FRR are printed in the last two columns of Fig. 5. The performance for database 1 has improved, achieving an EER of about 4.5%. For database 2 the result is only slightly better with an EER of about 4.2%.

It follows from the results that the Reliable Components Scheme degrades the classification performance when compared to the likelihood ratio based scheme but performance is still of the same order (from a security point of view).

5 Security Analysis

The helper data consists of two parts ($\boldsymbol{W1}$ and $W2$) which are used for reliable feature extraction and noise correction on discrete data respectively, we discuss the information leakage by both parts. We present the analysis under the assumption that the quantized strings $Q(\boldsymbol{X})$ are randomly distributed over $\{0,1\}^{511}$ [7]. It follows from results in [TG04], that $\mathsf{H}(S|W2) = \mathsf{H}(S)$, i.e. $W2$ leaks no information on S. It follows from the assumption on the distribution of $Q(\boldsymbol{X})$ that $\boldsymbol{W1}$ does not provide information on S. Hence, the scheme is 0-revealing. It follows from the results in [TG04] that for the discrete case $\mathsf{H}(Q(\boldsymbol{X})|W) \geq \mathsf{H}(Q(\boldsymbol{X})) - (K - s)$ when a (K, s, d) BCH code is used. For database 1 using a (511,76,85) code, this implies that the helper data $W2$ reveals 435 bits and for database 2 using a (511,40,95) code it reveals 471 bits. We note however that given the helper data $W2$, the space of quantized fingerprints $Q(\boldsymbol{X})$ is still sufficiently large (2^{76} and 2^{40} respectively) to make an attack exploiting the helper data infeasible. Again from the assumption on the distribution of $Q(\boldsymbol{X})$ it follows that $W1$ does not increase the information leakage substantially.

6 Conclusions

We showed in this paper, that template protecting biometric authentication techniques can be efficiently implemented with a performance of EER $\approx 4.2\%$ and secret size ≈ 40 bits on fingerprints. The main idea consist of splitting the helper data in two parts, one part determines the reliable components and the other part allows for noise correction on the quantized representations.

[7] We can not prove this at the moment and need more data to compute the distribution of the strings $Q(\boldsymbol{X})$. The presented analysis gives however a good idea of how the security of the system has to be analyzed

References

[Baz02] A.M. Bazen and S.H. Gerez, Systematic Methods for the Computation of the Directional Field and Singular Points of Fingerprints *IEEE Trans. PAMI*,2002, 24, 7, 905-919.

[Baz04] A.M. Bazen and R.N.J. Veldhuis, Detection of cores in fingerprints with improved dimension reduction *Proc. SPS 2004*, 41-44, Hilvarenbeek, The Netherlands,

[BV04] A.M. Bazen and R.N.J. Veldhuis, Likelihood Ratio-Based Biometric Verification, *IEEE Trans. Circuits and Systems for Video Technology*, 2004, 14, 1, 86-94.

[Bolling] J. Bolling, A window to your health, In *Jacksonville Medicine*, 51, Special Issue: Retinal deseases.

[DRS04] Y. Dodis, L. Reyzin, A. Smith, Fuzzy Extractors: How to generate strong secret keys from biometrics and other noisy data, In *Advances in Cryptology - Eurocrypt'04*, LNCS 3027, 523–540, 2004.

[Jai00] A.K. Jain and S. Prabhakar and L. Hong and S. Pankanti, Filterbank-Based Fingerprint Matching, *IEEE Trans. Image Processing*, 2000, 9, 5, 846-859.

[JPP04] U. Uludag, S. Pankanti, S. Prabhakar, and A.K. Jain, Biometric Cryptosystems: Issues and Challenges, In *Proceedings of the IEEE*, Vol. 92, 6, June 2004.

[JS02] A. Juels, M. Sudan, A Fuzzy Vault Scheme *Proceedings of the 2002 International Symposium on Information Theory (ISIT 2002)*, Lausanne.

[JW99] A. Juels and M. Wattenberg, A fuzzy commitment scheme, *6th ACM Conference on Computer and Communication Security*, p. 28-36, 1999.

[LT03] J.-P. Linnartz and P. Tuyls, New shielding functions to enhance privacy and prevent misuse of biometric templates, *4th International Conference on Audio- and Video-Based Biometric Person Authentication*, 2003.

[Mai00] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman and A.K. Jain, FVC2000: Fingerprint Verification Competition, *IEEE Trans. PAMI*, 2002, 24, 3, 402-412.

[MMJ03] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer-Verlag New-York 2003.

[P65] L. Penrose, Dermatoglyphic topology, *Nature*, 205, (1965) 545-546.

[S99] B. Schneier, Inside risks: The uses and abuses of Biometrics, *Communications of the ACM*, 42, p136, 1999.

[Sou98] C. Soutar, D. Roberge, S.A. Stojanov, R. Gilroy and B.V.K. Vijaya Kumar, Biometric Encryption-Enrollment and Verification Procedures, Proc. of SPIE, Vol. 3386, 24-35, April 1998.

[TG04] P. Tuyls and J. Goseling, Capacity and Examples of Template Protecting Biometric Authentication Systems, *Biometric Authentication Workshop (BioAW 2004)*, LNCS 3087, 158–170, Prague, 2004.

[TVI04] P. Tuyls, E. Verbitskiy, T. Ignatenko, D. Schobben and T.H. Akkermans Privacy Protected Biometric Templates: Ear Identification Proceedings of SPIE, Vol. 5404, 176-182, April 2004.