Fingerprint template protection using minutia-pair spectral representations

Citation for published version (APA):

Document status and date:
Published: 05/04/2018

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
Fingerprint template protection using minutia-pair spectral representations

Taras Stanko¹, Bin Chen¹², Boris Škorić¹
¹ Eindhoven University of Technology  ² School of Computer and Information, Hefei University of Technology

Storage of biometric data requires some form of template protection in order to preserve the privacy of people enrolled in a biometric database. One approach is to use a Helper Data System. Here it is necessary to transform the raw biometric measurement into a fixed-length representation. In this paper we extend the spectral function approach of Stanko and Škorić [18], which provides such a fixed-length representation for fingerprints. First, we introduce a new spectral function that captures different information from the minutia orientations. It is complementary to the original spectral function, and we use both of them to extract information from the fingerprint image. Second, we construct a helper data system consisting of zero-leakage quantisation followed by the Code Offset Method. We show empirical data which demonstrates that applying our helper data system causes only a small performance penalty compared to fingerprint authentication based on the unprotected spectral functions.

Index Terms—Biometrics, fingerprint recognition, template protection, minutiae.

I. INTRODUCTION

A. Biometric template protection

Biometric authentication has become popular because of its convenience. Biometrics cannot be forgotten or left at home. Although biometric data is not exactly secret (we are leaving a trail of fingerprints, DNA etc.), it is important to protect biometric data for privacy reasons. Unprotected storage of biometric data could reveal medical conditions and would allow cross-matching of entries in different databases. Large-scale availability of unprotected biometric data would make it easier for malevolent parties to leave misleading traces at crime scenes (e.g. artificial fingerprints [13], synthesized DNA [8].) One of the easiest ways to properly protect a biometric database against breaches and insider attacks (scenarios where the attacker has access to decryption keys) is to store biometrics in hashed form, just like passwords. An error-correction step has to be added to get rid of the measurement noise. To prevent critical leakage from the error correction redundancy data, one uses a Helper Data System (HDS) [11], [5], [17], for instance a Fuzzy Extractor or a Secure Sketch [10], [6], [3]. The best known and simplest HDS scheme is the code-offset method (COM). The COM utilizes a linear binary error-correction code and thus requires a fixed-length representation of the biometric measurement. Such a representation is not straightforward when the measurement noise can cause features of the biometric to appear/disappear. For instance, some minutiae may not be detected in every image captured from the same finger.

A fixed-length representation called spectral minutiae was introduced by Xu et al. [24], [21], [22], [23]. For every detected minutia of sufficient quality, the method evaluates a Fourier-like spectral function on a fixed-size two-dimensional grid; the contributions from the different minutiae are added up. Disappearance of minutiae or appearance of new ones does not affect the size of this representation.

One of the drawbacks of Xu et al.’s construction is that phase information is discarded in order to obtain translation invariance. Nandakumar [14] proposed a variant which does not discard the phase information. However, it reveals personalised reliability data, which makes it difficult to use in a privacy-preserving scheme.

A minutia-pair based variant of Xu et al.’s technique was introduced in [18]. It has a more compact grid and reduced computation times. Minutia pairs (and even triplets) were used in [7], [9], but with a different attacker model that allows encryption keys to exist that are not accessible to the attacker.

B. Contributions and outline

First we extend the pair-based spectral minutiae method [18] by introducing a new spectral function that captures different information from the minutia orientations. Then we use the spectral functions as the basis for a template protection system. Our HDS consists of two stages. In the first stage, we discretise the analog spectral representation using a zero-leakage HDS [5], [17]. This first HDS reduces quantisation noise, and the helper data reveals no information about the quantised data. Discretisation of the spectral functions typically yields only one bit per grid point. We concatenate the discrete data from all the individual grid points into one long bitstring. In the second stage we apply the Code Offset Method. Our code of choice is a Polar code, because Polar code are low-complexity capacity-achieving codes with flexible rate. We present False Accept vs. False Reject tradeoffs at various stages of the data processing. We introduce the ‘superfinger’ enrollment method, in which we average the spectral functions from multiple enrollment images. By combining three enrollment images in this way, and constructing a polar code specifically tuned to the individual bit error rate of each bit position, we achieve an Equal Error Rate around 1% for a high-quality fingerprint database, and around 6% for a low-quality database.

The outline of the paper is as follows. In Section II we introduce notation briefly review helper data systems, the spectral minutiae representation, and polar codes. In Section III we introduce the new spectral function. In Section IV we explain our experimental approach and motivate certain design choices.
such as the number of discretisation intervals and the use of a Gaussian approximation. We introduce two methods for averaging enrollment images.

Section IV contains our results, mostly in the form of ROC curves. In Section V we discuss the results and identify topics for future work.

II. PRELIMINARIES

A. Notation and terminology

We use capitals to represent random variables, and lowercase for their realizations. Sets are denoted by calligraphic font. The set $S$ is defined as $S = \{0, \ldots, N - 1\}$. The mutual information (see e.g. [4]) between $X$ and $Y$ is $I(X;Y)$. The probability density function (pdf) of the random variable $X \in \mathbb{R}$ in written as $f(x)$ and its cumulative distribution function (cdf) as $F(x)$. We denote the number of minutiae found in a fingerprint by $Z$. The coordinates of the $j$'th minutia are $x_j = (x_j, y_j)$ and its orientation is $\theta_j$. We write $x = (x_j)_{j=1}^Z$ and $\theta = (\theta_j)_{j=1}^Z$. We will use the abbreviations $FRR = $ False Reject Rate, $FAR = $ False Accept Rate, $EER = $ Equal Error Rate, $ROC = $ Receiver Operating Characteristic. Bitwise xor of binary strings is denoted as $\oplus$.

B. Helper Data Systems

A HDS is a cryptographic primitive that allows one to reproducibly extract a secret from a noisy measurement. A HDS consist of two algorithms: Gen (generation) and Rec (reconstruction), see Fig. 1. The Gen algorithm takes a measurement $X$ as input and generates the secret $S$ and a helper data $W$. The Rec algorithm has as input a noisy measurement $Y$ and the helper data; it outputs an estimator $\hat{S}$. If $Y$ is sufficiently close to $X$ then $\hat{S} = S$. The helper data should not reveal much about $S$. Ideally it holds that $I(W;S) = 0$. This is known as Zero Leakage helper data.

$$X \xrightarrow{\text{Gen}} S \xrightarrow{W} Y \xrightarrow{\text{Rec}} \hat{S}$$

Fig. 1: Data flow in a generic Helper Data System.

C. Two-stage HDS template protection scheme

Fig. 2 shows the two-stage HDS architecture mentioned in Section I-B. The enrollment measurement $x$ is transformed to the spectral representation $(x_i)_{i=1}^M$ on $M$ grid points. The first-stage enrollment procedure Gen1 is applied to each $x_i$ individually, yielding short (mostly one-bit) secrets $s_i$ and zero-leakage helper data $w_i$. The $s_1 \ldots s_M$ are concatenatated into a string $k$. Residual noise in $k$ is dealt with by the second-stage HDS (Code Offset Method), whose Gen2 produces a secret $c$ and helper data $r$. A hash $h(c||z)$ is computed, where $z$ is salt. The hash and the salt are stored.

In the verification phase, the noisy $y$ is processed as shown in the bottom half of Fig. 2. The reconstructed secret $\hat{c}$ is hashed with the salt $z$; the resulting hash is compared to the stored hash.

D. Minutia-pair spectral representation

Minutiae are special features in a fingerprint, e.g. ridge endings and bifurcations. We briefly describe the minutia-pair spectral representation introduced in [13]. For minutia indices $a, b \in \{1, \ldots, Z\}$ the distance and angle between these minutiae are given by $R_{ab} = |x_a - x_b|$ and $\tan \phi_{ab} = \frac{y_a - y_b}{x_a - x_b}$. The spectral function $M_{x\theta}$ is defined as

$$M_{x\theta}(q, R) = \sum_{a=1}^Z \sum_{b=\alpha+1}^Z e^{iq\phi_{ab}} e^{-\frac{(R_{ab} - R)^2}{2\sigma^2}} e^{i\theta_{ab} - \theta_a},$$

where $\sigma$ is a width parameter. The spectral function is evaluated on a discrete $(q, R)$ grid. The variable $q$ is integer and can be interpreted as the Fourier conjugate of an angular variable, i.e. a harmonic. The function $M_{x\theta}$ is invariant under translations of $x$. When a rotation of the whole fingerprint image is applied over an angle $\delta$, the spectral function transforms in a simple way,

$$M_{x\theta}(q, R) \rightarrow e^{iq\delta} M_{x\theta}(q, R).$$

E. Zero Leakage Helper Data Systems

We briefly review the ZLHDS developed in [5], [17] for quantisation of an enrollment measurement $X \in \mathbb{R}$. The density function of $X$ is $f$, and the cumulative distribution function is $F$. The verification measurement is $Y$. The $X$ and $Y$ are considered to be noisy versions of an underlying 'true' value. They have zero mean and variance $\sigma_X^2, \sigma_Y^2$, respectively. The correlation between $X$ and $Y$ can be characterised by writing $Y = \lambda X + V$, where $\lambda \in [0, 1]$ is the attenuation parameter and $V$ is zero-mean noise independent of $X$, with variance $\sigma_V^2$. It holds that $\sigma_Y^2 = \lambda^2 \sigma_X^2 + \sigma_V^2$. We consider the identical conditions case: the amount of noise is the same during enrollment and reconstruction. In this situation we have $\sigma_X^2 = \sigma_Y^2$ and $\lambda^2 = 1 - \frac{\sigma_V^2}{\sigma_X^2}$.

The real axis $\mathbb{R}$ is divided into $N$ intervals $\Omega_\alpha = (\Omega_\alpha, \Omega_{\alpha+1})$, with $\alpha \in S, S = \{0, \ldots, N-1\}$. Let $p_\alpha = \Pr[X \in \Omega_\alpha]$. The quantisation boundaries are given by $\Omega_\alpha = F^{-1}(\sum_{j=0}^{\alpha-1} p_j)$. The Gen algorithm produces the secret $s$ as $s = \max\{\alpha \in S : x \geq \Omega_\alpha\}$ and the helper data $w \in [0, 1]$ as $w = [F(x) - \sum_{j=0}^{s-1} p_j]/p_s$. The inverse relation, for computing $x$ as a function of $s$ and $w$, is given by $x_{s,w} = F^{-1}(\sum_{j=0}^{s-1} p_j + wp_s)$. The Rec algorithm computes the estimator $\hat{s}$ as the value in $S$ for which it holds that $y \in (\tau_{s,w}, \tau_{s+1,w})$, where the parameters $\tau$ are decision boundaries. In the case of Gaussian noise these boundaries are given by

$$\tau_{s,w} = \frac{\lambda x_{s-1,w} + \xi_{s,w}}{2} + \frac{\sigma_Y^2}{\lambda} \ln \frac{p_{s-1}}{p_s},$$

Here it is understood that $\xi_{-1,w} = -\infty$ and $\xi_{N,w} = \infty$, resulting in $\tau_{0,w} = -\infty, \tau_{N,w} = \infty$. The above scheme ensures that $I(W;S) = 0$ and that the reconstruction errors are minimized.
F. The Code Offset Method (COM)

We briefly describe how the COM is used as a Secure Sketch. Let $C$ be a linear binary error correcting code with message space $\{0,1\}^m$ and codewords in $\{0,1\}^n$. It has an encoding $\text{Enc}: \{0,1\}^m \rightarrow \{0,1\}^n$, a syndrome function $\text{Syn}: \{0,1\}^n \rightarrow \{0,1\}^{n-m}$ and a syndrome decoder $\text{SynDec}: \{0,1\}^{n-m} \rightarrow \{0,1\}^n$. In Fig. 2 the Gen2 computes the helper data $r$ as $r = \text{Syn} k$. The $c$ in Fig. 2 is equal to $k$. The Rep2 computes the reconstruction $\hat{k} = k' \oplus \text{SynDec}(r \oplus \text{Syn} k')$.

\[
M_{x\beta}(q, R) = \sum_{a=1}^{Z} \sum_{b=a+1}^{Z} e^{i \phi_{ab}(q-2)} e^{-\frac{(R-R_{ab})^2}{2\sigma^2}} e^{i(\theta_a + \theta_b)}. \tag{4}
\]

We will use $M_{x,0}$, $M_{x,\beta}$ and their fusion.

IV. EXPERIMENTAL APPROACH

A. Databases

We use the MCYT, FVC2000, and FVC2002 database. The MCYT database [15] contains good-quality images from 100 individuals: 10 fingers per individual and 12 images per finger. FVC2000 and FVC2002 contain low-quality images (only index and middle fingers [12]). Each FVC database contains 100 fingers, 8 images per finger. In FVC2002, images number 3, 4, 5, and 6 have an exceptionally large angular displacement, so they are omitted from the experiments.

We extract the minutia position and orientation $(x_j, y_j, \theta_j)$ by using VeriFinger software [11]. For MCYT we evaluate the spectral functions on the same grid as [18], namely $R \in \{16, 22, 28, \ldots, 130\}$ and $q \in \{1, 2, \ldots, 16\}$ and we maintain $\sigma = 2.3$ pixels. For the FVC databases we use the same grid, and $\sigma = 3.2$ pixels turns out to be a good choice. The average number of minutiae that can be reliably found is $Z = 35$.

III. A NEW SPECTRAL FUNCTION

Consider Fig. 3 (modified from [20]). The invariant angle $\beta_a$ is defined as the angle from the orientation of minutia $a$ to the connecting line $ab$, taken in the positive direction. (The $\beta_b$ is defined analogously). Modulo $2\pi$ it holds that $\theta_a + \beta_a = \phi_{ab}$ and $\theta_b + \beta_a = \phi_{ab} + \pi$. The spectral function (1) uses only the invariant angle $\beta_a - \beta_b + \pi = \theta_b - \theta_a$. The second invariant angle, which can be written e.g. as $\pi - \beta_a - \beta_b = \theta_a + \theta_b - 2\phi_{ab}$, is not used. We therefore now introduce a new spectral function, denoted as $M_{x,\beta}$, which incorporates the invariant angle $\pi - \beta_a - \beta_b$.

![Fig. 3: The relevant angles in a minutia pair.](image)
B. No image rotation

As mentioned in [18], during the reconstruction procedure one can try different rotations of the verification image, but it results only in a minor improvement of the EER. For this reason we do not apply image rotation.

C. Quantization methods

Before quantization all spectral functions are normalized to zero mean and unit variance, where the variance is taken of the real and imaginary part together. We quantize the real and imaginary part of the spectral functions separately. We study two methods: ‘hard thresholding’ (without helper data) and the Zero Leakage quantisation of Section II-B. The hard thresholding gives a bit value ‘1’ if \( \text{Re} M > 0 \) and ‘0’ otherwise. We will show results for this method mainly to demonstrate the advantages of Zero Leakage quantisation.

D. Gaussian probability distributions

When using the ZLHDS formulas we will assume that the spectral functions are Gaussian-distributed. Figs. 4 and 5 illustrate that this assumption is not far away from the truth.

E. Zero leakage quantization

1) Signal to noise ratio; setting \( N \)

In the ZL HDS of Section II-E the optimal choice of the parameter \( N \) (number of quantization intervals) depends on the signal to noise ratio. Fig. 6 shows a comparison between \( N = 2 \) and \( N = 3 \). At low noise it is obvious that \( N = 3 \) extracts more information from the source than \( N = 2 \). At \( \sigma_V / \sigma_X \) larger than approximately 0.3, there is a regime where \( N = 3 \) can extract more in theory, but is hindered in practice by the high bit error rate. At \( \sigma_V / \sigma_X > 0.55 \) the \( N = 2 \) ‘wins’ in all respects.

For our data set, we define a \( \sigma^2_{V}(q,R) \) for every grid point \( (q,R) \) as the variance of \( M(q,R) \) over all images in the database. The noise \( \sigma^2_{V}(q,R) \) is the variance over all available images of the same finger, averaged over all fingers. Figs. 7 and 8 show the noise-to-signal ratio. Note the large amount of noise; even the best grid points have \( \sigma_V / \sigma_X > 0.45 \). Fig. 6 tells us that setting \( N = 2 \) is the best option, and this is the choice we make. At \( N = 2 \) we extract two bits per grid point from each spectral function (one from \( \text{Re} M \), one from \( \text{Im} M \)). Hence our bit string string \( k \) (see Fig. 2) derived from \( M_{x\theta} \) has length 640. When we apply fusion of \( M_{x\theta} \) and \( M_{x\beta} \) this becomes 1280.

For \( N = 2 \) the formulas in Section II-E simplify to \( A_0 = (-\infty,0), A_1 = [0,\infty), p_0 = p_1 = \frac{1}{2}, \xi_{0,w} = F_{\text{inv}}(w), \xi_{1,w} = F_{\text{inv}}(\frac{w}{2}), \tau_{w} = \frac{1}{2}(\xi_{0,w} + \xi_{1,w}). \) Since we work with Gaussian distributions, \( F \) is the Gaussian cdf (‘probability function’).

2) Enrollment and reconstruction

We have experimented with three different enrollment methods:

E1. A single image is used.

E2: We take the first \( t \) images of a finger and calculate the average spectral function. We call this the ‘superfinger’
method. In the ZLHDS calculations the signal-to-noise ratio of the average spectral function is used.

E3: For each of $t$ images we calculate an enrollment string $k$. We apply bitwise majority voting on these strings. (This requires odd $t$.) The reconstruction boundaries are calculated based on the superfinger method, i.e. as in E2.

**Reconstruction:**

We study fingerprint authentication with genuine pairs and impostor combinations (all images of all impostor fingers), resulting in $O(M^2 L^2)$ data points. For the MCYT database, which is larger, we take only one random image per impostor finger, resulting in $O(M L^2)$ data points.

E2+E3: For genuine pairs we compare the superfinger to the remaining $M-t$ images. Thus we have $(M-t)L$ data points. Impostor pairs are generated as for E1.

Note: The VeriFinger software was not able to extract information for every image.

**V. Experimental results**

**A. FAR/FRR rates before error correction**

For each the data processing steps/options before application of the Code Offset method, we investigate the False Accept rates and False Reject rates. We identify a number of trends.

- Figs. 9 and 10 show ROC curves. All the non-analog curves were made under the implicit assumption that for each decision threshold (number of bit flips) an error-correcting code can be constructed that enforces that threshold, i.e. decoding succeeds only if the number of bit flips is below the threshold. Unsurprisingly, we see in the figures that quantisation causes a performance penalty. Furthermore the penalty is clearly less severe when the ZLHDS is used. Finally, it is advantageous to discard some grid points that have bad signal-to-noise ratio. For the curves labeled ‘ZLHDS+reliable components’ only the least noisy $512$ bits of $k$ were kept (1024 in the case of fusion). Our choice for the number $512$ is not entirely arbitrary: it fits error-correcting codes. Note in Fig. 10 that ZLHDS with reliable component selection performs better than analog spectral functions without reliable component selection. (But not better than analog with selection.)

- The E2 and E3 enrollment methods perform better than E1. Furthermore, performance increases with $t$. A typical example is shown in Fig. 11.

- The spectral functions $M_{xθ}$ and $M_{xβ}$ individually have roughly the same performance. Fusion yields a noticeable improvement. An example is shown in Fig. 12. (We implemented fusion in the analog domain as addition of the two similarity scores.)

- Tables I to V show Equal Error Rates and Bit Error Rates. We see that enrollment methods E2 and E3 have similar performance, with E2 yielding a somewhat lower genuine-pair BER than E3.

- In Table VII it may look strange that the EER in the rightmost column is sometimes lower than in the ‘analog’ column. We think this happens because there is no reliable component selection in the ‘analog’ procedure.

- Ideally the impostor BER is 50%. In the tables we see that the impostor BER can get lower than 50% when the ZLHDS is used and the enrollment method is E2. On the other hand, it is always around 50% in the ‘No HDS’ case. This seems

3 This is defined as a global property of the whole database. The selection of reliable components does not reveal anything about an individual. Note that [14] does not reveal personalised reliable components and obtains better FA and FN error rates.
Fig. 9: Performance result for several processing methods. FVC2000. Enrollment method E2 with $t = 3$.

Fig. 10: Performance result for several processing methods. MCYT. Enrollment method E2 with $t = 3$.

to contradict the Zero Leakage property of the helper data system. The ZLHDS is supposed not to leak, i.e. the helper data should not help impostors. However, the zero-leakage property is guaranteed to hold only if the variables are independent. In real-life data there are correlations between grid points and correlations between the real and imaginary part of a spectral function.

B. Error correction: Polar codes

The error rates in the genuine reconstructed $\hat{k}$ are rather high, at least 0.21. In order to apply the Code Offset Method with a decent message size it is necessary to use a code that has a high rate even at small codeword length. Consider the case of fusion of $M_{x,9}$ and $M_{x,3}$. The codeword length is 1280 bits (1024 if reliable component selection is performed). Suppose we need to distinguish between $2^{20}$ users. Then the message length needs to be at least 20 bits, in spite of the high bit error rate. Furthermore, the security of the template protection is determined by the entropy of the data that is input into the hash function (see Fig. 2); it would be preferable to have at least 64 bits of entropy.

We constructed a number of Polar codes tuned to the signal-to-noise ratios of the individual grid points. The codes are designed to find a set of reliable channels, which are then assigned to the information bits. Each code yields a certain FAR (impostor string accidentally decoding correctly) and FRR (genuine reconstruction string failing to decode correctly), and hence can be represented as a point in an ROC plot. This is shown in Fig. 13. For the MCYT database we have constructed a Polar code with message length 25 at an EER around 1.2% (compared to 0.7% before error correction). For
the FVC2000 database we have constructed a Polar code with message length 15 at an EER around 6% (compared to 3.3% EER before error correction). Note that the error correction is an indispensable part of the privacy protection and inevitably leads to a performance penalty. However, we see that the penalty is not that bad, especially for high-quality fingerprints. From our results we also see that even under the best circumstances (high-quality MCYT database) the entropy of the extracted string is severely limited ($\leq 25$ bits). In order to achieve a reasonable security level of the hash, at least two fingers need to be combined. We do not see this as a drawback of our helper data system; given that the EER for one finger is around 1%, which is impractical in real-life applications, it is necessary anyhow to combine multiple fingers.

C. Error correction: random codebooks

There is a large discrepancy between the message length of the Polar code ($k \leq 25$) and the known information content of a fingerprint. According to Ratha et al. [16], the reproducible entropy of a fingerprint image with $Z = 35$ robustly detectable minutiae should be more than 120 bits. Furthermore, the potential message size that can be carried in a 1024-bit string with a BER of 23% is $1024[1 - h(0.23)] = 227$ bits. (And 122 bits at 30% BER.)

We experimented with random codebooks to see if we could extract more entropy from the data than with polar codes. At low code rates, a code based on random codewords can be practical to implement. Let the message size be $\ell$, and the
<table>
<thead>
<tr>
<th>#images (t)</th>
<th>Analog</th>
<th>No HDS</th>
<th>ZLHDS</th>
<th>ZLHDS+r.c.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(M_{x\theta})</td>
<td>6.0%</td>
<td>9.4%</td>
<td>9.0%</td>
<td>8.0%</td>
</tr>
<tr>
<td></td>
<td>0.39</td>
<td>0.50</td>
<td>0.37</td>
<td>0.50</td>
</tr>
<tr>
<td>(M_{x\beta})</td>
<td>6.1%</td>
<td>9.4%</td>
<td>9.0%</td>
<td>8.0%</td>
</tr>
<tr>
<td></td>
<td>0.39</td>
<td>0.50</td>
<td>0.37</td>
<td>0.50</td>
</tr>
<tr>
<td>Fusion</td>
<td>4.8%</td>
<td>7.3%</td>
<td>5.2%</td>
<td>4.5%</td>
</tr>
<tr>
<td></td>
<td>0.39</td>
<td>0.50</td>
<td>0.37</td>
<td>0.50</td>
</tr>
</tbody>
</table>

TABLE II: EERs and BERs for the FVC2000 database. Enrollment methods E1 and E2.

<table>
<thead>
<tr>
<th>#images (t)</th>
<th>Analog</th>
<th>No HDS</th>
<th>ZLHDS</th>
<th>ZLHDS+r.c.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(M_{x\theta})</td>
<td>4.5%</td>
<td>7.2%</td>
<td>5.7%</td>
<td>5.0%</td>
</tr>
<tr>
<td></td>
<td>0.37</td>
<td>0.50</td>
<td>0.33</td>
<td>0.47</td>
</tr>
<tr>
<td>(M_{x\beta})</td>
<td>4.8%</td>
<td>7.9%</td>
<td>5.7%</td>
<td>5.0%</td>
</tr>
<tr>
<td></td>
<td>0.38</td>
<td>0.50</td>
<td>0.34</td>
<td>0.47</td>
</tr>
<tr>
<td>Fusion</td>
<td>3.9%</td>
<td>5.1%</td>
<td>5.0%</td>
<td>4.1%</td>
</tr>
<tr>
<td></td>
<td>0.37</td>
<td>0.50</td>
<td>0.33</td>
<td>0.47</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#images (t)</th>
<th>Analog</th>
<th>No HDS</th>
<th>ZLHDS</th>
<th>ZLHDS+r.c.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(M_{x\theta})</td>
<td>3.0%</td>
<td>5.6%</td>
<td>5.3%</td>
<td>4.4%</td>
</tr>
<tr>
<td></td>
<td>0.36</td>
<td>0.50</td>
<td>0.31</td>
<td>0.46</td>
</tr>
<tr>
<td>(M_{x\beta})</td>
<td>3.2%</td>
<td>7.2%</td>
<td>5.7%</td>
<td>4.9%</td>
</tr>
<tr>
<td></td>
<td>0.37</td>
<td>0.50</td>
<td>0.32</td>
<td>0.46</td>
</tr>
<tr>
<td>Fusion</td>
<td>2.2%</td>
<td>4.5%</td>
<td>4.0%</td>
<td>3.5%</td>
</tr>
<tr>
<td></td>
<td>0.37</td>
<td>0.50</td>
<td>0.32</td>
<td>0.46</td>
</tr>
<tr>
<td>(M_{x\theta})</td>
<td>2.1%</td>
<td>5.5%</td>
<td>5.3%</td>
<td>4.8%</td>
</tr>
<tr>
<td></td>
<td>0.37</td>
<td>0.50</td>
<td>0.31</td>
<td>0.45</td>
</tr>
<tr>
<td>(M_{x\beta})</td>
<td>2.2%</td>
<td>7.1%</td>
<td>6.3%</td>
<td>5.0%</td>
</tr>
<tr>
<td></td>
<td>0.37</td>
<td>0.50</td>
<td>0.32</td>
<td>0.46</td>
</tr>
<tr>
<td>Fusion</td>
<td>1.3%</td>
<td>4.3%</td>
<td>4.3%</td>
<td>3.4%</td>
</tr>
<tr>
<td></td>
<td>0.37</td>
<td>0.50</td>
<td>0.31</td>
<td>0.45</td>
</tr>
</tbody>
</table>

TABLE III: EERs and BERs for the FVC2002 database. Enrollment methods E1 and E2.

<table>
<thead>
<tr>
<th>#images (t)</th>
<th>Analog</th>
<th>No HDS</th>
<th>ZLHDS</th>
<th>ZLHDS+r.c.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(M_{x\theta})</td>
<td>5.8%</td>
<td>12.1%</td>
<td>10.9%</td>
<td>8.8%</td>
</tr>
<tr>
<td></td>
<td>0.38</td>
<td>0.50</td>
<td>0.37</td>
<td>0.50</td>
</tr>
<tr>
<td>(M_{x\beta})</td>
<td>6.4%</td>
<td>10.9%</td>
<td>10.9%</td>
<td>9.2%</td>
</tr>
<tr>
<td></td>
<td>0.39</td>
<td>0.50</td>
<td>0.38</td>
<td>0.50</td>
</tr>
<tr>
<td>Fusion</td>
<td>5.5%</td>
<td>9.4%</td>
<td>9.3%</td>
<td>7.0%</td>
</tr>
<tr>
<td></td>
<td>0.39</td>
<td>0.50</td>
<td>0.38</td>
<td>0.50</td>
</tr>
<tr>
<td>(M_{x\theta})</td>
<td>5.4%</td>
<td>10.9%</td>
<td>9.8%</td>
<td>7.3%</td>
</tr>
<tr>
<td></td>
<td>0.39</td>
<td>0.50</td>
<td>0.35</td>
<td>0.48</td>
</tr>
<tr>
<td>(M_{x\beta})</td>
<td>5.5%</td>
<td>10.7%</td>
<td>8.4%</td>
<td>7.4%</td>
</tr>
<tr>
<td></td>
<td>0.39</td>
<td>0.50</td>
<td>0.36</td>
<td>0.48</td>
</tr>
<tr>
<td>Fusion</td>
<td>4.4%</td>
<td>9.8%</td>
<td>7.3%</td>
<td>5.9%</td>
</tr>
<tr>
<td></td>
<td>0.39</td>
<td>0.50</td>
<td>0.36</td>
<td>0.48</td>
</tr>
</tbody>
</table>

TABLE IV: EERs and BERs for the FVC2002 database. Enrollment method E3.

VI. SUMMARY AND DISCUSSION

A Helper Data System protects privacy but causes a fingerprint recognition degradation in the form of increased EER. We have built a HDS from a spectral function representation of fingerprint data, combined with a Zero Leakage quantisation scheme. It turns out that our HDS causes only a very small EER penalty when the fingerprint quality is high. The best results were obtained with the ‘superfinger’ enrollment method (E2, taking the average over multiple enrollment images in the spectral function domain), and with fusion of the \(M_{x\theta}, M_{x\beta}\) functions. The superfinger method performs slightly better than the E3 method and also has the advantage that it is not restricted to an odd number of enrollment captures.

For the high-quality MCYT database, our HDS achieves an EER around 1% and extracts a 1024-bit string with ≤ 25 bits of entropy. In practice multiple fingers need to be used in order to obtain an acceptable EER. This automatically increases the entropy of the hashed data. The entropy can be further increased by employing tricks like the Spammed Code Offset Method [19].

As topics for future work we mention (i) testing the HDS on more databases; (ii) further optimisation of parameter choices such as the number of reliable components, and the number of minutiae used in the computation of the spectral functions; (iii) further tweaking of the Polar codes.

ACKNOWLEDGMENTS

Part of this work was supported by NWO grant 628.001.019 (ESPRESSO), and grant 61701155 from the National Natural Science Foundation of China (NSFC).

REFERENCES

Fig. 13: FRR versus FAR achieved by Polar codes and random codebooks (average over three random codebooks). The ◆ markers denote random codebook, and happen to coincide with the line connecting the Polar markers.


