

# Multi-Bits Biometric String Generation based on the Likelihood Ratio

C. Chen, R.N.J. Veldhuis  
Signals and Systems Group, Electrical Engineering  
University of Twente  
P.O. Box 217, 7500 AE Enschede, The Netherlands  
{c.chen, r.n.j.veldhuis}@utwente.nl

T.A.M. Kevenaar, A.H.M. Akkermans  
Philips Research  
Prof. Holstlaan 4, 5656 AA, Eindhoven, The Netherlands  
{tom.kevenaar, ton.h.akkermans}@philips.com

**Abstract**—Preserving the privacy of biometric information stored in biometric systems is becoming a key issue. An important element in privacy protecting biometric systems is the quantizer which transforms a normal biometric template into a binary string. In this paper, we present a user-specific quantization method based on a likelihood ratio approach (LQ). The bits generated from every feature are concatenated to form a fixed length binary string that can be hashed to protect its privacy. Experiments are carried out on both fingerprint data (FVC2000) and face data (FRGC). Results show that our proposed quantization method achieves a reasonably good performance in terms of *FAR/FRR* (when *FAR* is  $10^{-4}$ , the corresponding *FRR* are 16.7% and 5.77% for FVC2000 and FRGC, respectively).

## I. INTRODUCTION

Use of biometrics has brought considerable benefits in the area of access control and ICT security. Recently, however, protection of biometric template is becoming more important [1], because a biometric template may reveal personal information. Additionally, unprotected storage and transfer of biometric information allows direct steal-and-use impersonation. Once the biometric template is compromised, it can not be re-issued.

Biometric template protection aims to protect biometric reference information stored in biometric systems from abuse. In the past years, several techniques were developed to protect biometric information. In [2], [3] the authors discuss an approach known as ‘cancelable biometrics’. Before storing the image of a face or a fingerprint in a biometric system, it is distorted using a parametrized one-way geometric distortion function. The fuzzy vault method as introduced in [4] is a general cryptographic construction allowing to store a secret in a vault that can be locked using an unordered set of features. An initial attempt to use the fuzzy vault scheme in the setting of fingerprints is given in [5]. A third group of techniques, containing fuzzy commitments [6], fuzzy extractors [7] and helper data systems [8], derive a key from a biometric measurement and store an irreversibly hashed version of the key in the biometric system. It is the purpose of all these methods to protect the privacy of biometric information without reducing the performance of the biometric system in terms of False Acceptance Rate (*FAR*) and False Rejection Rate (*FRR*).

In this paper we will concentrate on the third group of methods. In order to extract a key, these methods assume

that a biometric template can be represented as a fixed length binary string. In effect, these methods define the similarity of two binary templates in terms of Hamming distance [9]. A binary template is usually obtained by quantizing the original biometric template using a quantizer. In order to work properly, many quantizers produce and use side-information [8], [9], [10] that must be stored in the biometric system. Since this side-information is user dependent, it may leak information about the original template. Side-information with low privacy leakage is therefore a design objective.

So far, few quantization-based template methods have been proposed. Tuyls et al. [10] first introduced the fixed-interval quantization (FQ) with one bit per feature, in which two intervals are separated at the mean of the background distribution. However, they report an Equal Error Rate (*EER*) which is quite high (5.3%) when compared with the *EER* of a likelihood ratio classifier (LC) on the same data. Moreover, the one-bit per feature quantization generates only short binary strings which may be vulnerable to a brute force attack. Zhang et al. [11] introduced fixed interval quantization with multi-bits per feature (ZQ), in which the quantization intervals are determined by the mean and the standard deviation of the feature. However, the quantization method they proposed is not optimal in terms of *FAR* and *FRR*, and the security issue is not addressed by them.

Therefore, in this paper, we propose a user-specific, likelihood ratio based quantizer (LQ) that allows to extract multiple bits from a single feature. Experiments are carried out on both fingerprint data (FVC2000) and face data (FRGC). Results show that our proposed quantization method achieves a reasonably good performance in terms of *FAR/FRR* (when *FAR* is  $10^{-4}$ , the corresponding *FRR* are 16.7% and 5.77% for FVC2000 and FRGC, respectively). In the mean time, the stored side-information retains high security.

In section II, our algorithm is presented. In section III, experiments on synthetic and real data are explained. In section IV, we discuss the method while conclusions and directions for further research are given in section V.

## II. MULTI-BITS QUANTIZATION

The framework that we describe is similar to the Helper Data scheme proposed in [10]. It basically includes three parts: (1) extracting features; (2) quantization and coding per feature and concatenating the output codes; (3) applying

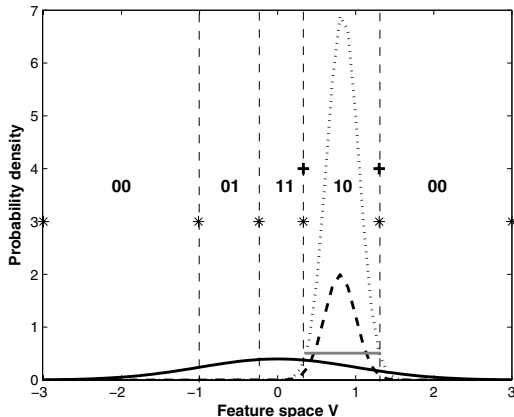


Fig. 1. An example of constructing a one-dimensional quantizer based on the likelihood ratio  $L_\omega$  (dotted). The background pdf is  $G(v, 0, 1)$  (solid), the genuine user pdf is  $G(v, \mu_\omega = 0.8, \sigma_\omega = 0.2)$  (dashed), threshold  $t$  (grey). + illustrates the genuine user interval, whilst \* illustrates the complete quantization intervals and the intervals are labeled with Gray code.

error correction coding (ECC) and hashing. However, in this paper, we propose a new approach for the first two items.

#### A. Extracting reliable, distinctive and independent features

One important step before applying quantization is to extract reliable, distinctive and independent features. In this paper our models assume Gaussian distributions and equal within-class variations. Therefore, a sufficient number of samples is required to provide reliable Gaussian parameters. Additionally, we require distinctive features, with small within-class variation and large between-class variation [12], to reduce quantization errors. Furthermore, we require features that are independent, with respect to both the background distributions and the genuine user distribution. Independent features can reduce the quantization error and subsequently generate independent bits. To extract features which meet the above requirements, we choose the PCA/LDA processing method described in [13].

#### B. Quantization and concatenation

The user-specific quantization is applied independently to each feature dimension, and the output codes are concatenated as the binary string. The idea of using likelihood ratio is driven by its optimal *FAR/FRR* performance in many biometric applications [14]. In a one-dimensional feature space  $\mathbb{V}$  the likelihood ratio of user  $\omega$  is defined as:

$$L_\omega = \frac{G(v, \mu_\omega, \sigma_\omega)}{G(v, \mu_0, \sigma_0)}, \quad (1)$$

where  $v$ ,  $\mu$  and  $\sigma$  are scalars. Due to the PCA/LDA processing, we have  $G(v, \mu_0, \sigma_0)$  with  $(\mu_0 = 0; \sigma_0 = 1)$  as the background probability density function (pdf) and  $G(v, \mu_\omega, \sigma_\omega)$  as the genuine user pdf [14].

Fig. 1 shows an example of constructing a one-dimensional quantizer, given both probability density functions. By applying a threshold  $t \in [0, \infty)$  to the likelihood

ratio  $L_\omega$ , a genuine quantization interval  $Q_{\text{genuine}, \omega}$  is determined in the feature space  $\mathbb{V}$ , in which the genuine user  $\omega$  is assigned:

$$Q_{\text{genuine}, \omega} = \{v \in \mathbb{V} \mid L_\omega \geq t\}. \quad (2)$$

With  $Q_{\text{genuine}, \omega}$ , the probability  $P_\omega$  for an impostor to be inside the genuine quantization interval can be calculated:

$$P_\omega = \int_{Q_{\text{genuine}, \omega}} G(v, 0, 1) dv. \quad (3)$$

We construct the rest of the quantization intervals such that they have the same probability mass  $P_\omega$  in the background distribution. This gives an attacker no additional information on which is the genuine interval. Furthermore, it can be seen that this might lead to independent bits derived from a single feature. Thus we have:

$$\begin{aligned} \bigcup_{k=1}^{K_\omega} Q_{k, \omega} &= \mathbb{V}, \\ Q_{k, \omega} \cap Q_{l, \omega} &= \emptyset, k \neq l, \\ Q_{k, \omega} &= Q_{\text{genuine}, \omega}, \text{ for certain } k, \\ \int_{Q_{k, \omega}} G(v, 0, 1) dv &= P_\omega, \end{aligned} \quad (4)$$

where  $K_\omega$  is the number of quantization intervals and  $Q_{k, \omega}$  is the quantization interval. In the following part, we will see that  $P_\omega$  presented in (3) equals the *FAR* for a single feature.

Given an arbitrary  $t$ , it is not always possible to let each quantization interval have this  $P_\omega$  probability. Usually the left-end and the right-end interval have a probability mass less than  $P_\omega$ . Therefore, we address them as one wrap-around interval. In order to meet (4), only thresholds  $t$  that can generate

$$P_\omega = 1/K_\omega, \quad (5)$$

are applicable in our algorithm. Based on the above procedure, a  $K_\omega$ -interval quantizer is established (\* in Fig. 1). Note that  $K_\omega$  might not be an exponential of 2 and it varies with different users. In most of the applications, we need to obtain a fixed code length  $L$  for all the users. For this reason, the code length need to be extended from  $\log_2 K_\omega$  to  $L$ ,  $L = \lceil \log_2 K_\omega \rceil$ .

Quantization intervals are labeled with a Gray code [15] which limits the Hamming distance of two adjacent code words to a single bit (see Fig. 1). This reduces the number of bit errors due to within-class variation.

Besides the binary code generated above, the quantizer information (known as side-information  $Q_\omega$ ) has to be stored for user  $\omega$  as well. Since the background pdf is known, we only have to randomly select one quantization interval ( $Q_{k, \omega} \mid k \in [1, K_\omega]$ ) as the side-information to be stored.

To extend the quantization to the  $m$ -dimensional case, we simply need to apply the above method to each feature dimension. The output binary string  $S_\omega$  is a concatenation of binary codes corresponding to the genuine intervals of each dimension, and the side-information is the collection of quantizer information for each dimension.

### C. FAR/FRR and security

Given a threshold  $t$ , the false acceptance rate  $FAR_{i,\omega}(t)$  and false rejection rate  $FRR_{i,\omega}(t)$  of user  $\omega$  with the one-dimensional feature  $i$  is given by:

$$FAR_{i,\omega}(t) = \int_{Q_{\text{genuine},\omega}} G(v, 0, 1) dv, \quad (6)$$

$$FRR_{i,\omega}(t) = 1 - \int_{Q_{\text{genuine},\omega}} G(v, \mu_\omega, \sigma_\omega) dv. \quad (7)$$

Assuming that the PCA/LDA process results in independent features, the FAR and FRR in the  $m$ -dimensional feature space  $\mathbb{V}^m$  for user  $\omega$ , with the threshold vector  $\mathbf{T} = [t_1 \dots t_m]$ , is defined as:

$$FAR_\omega(\mathbf{T}) = \prod_{i=1}^m FAR_{i,\omega}(t_i), \quad (8)$$

$$FRR_\omega(\mathbf{T}) = 1 - \prod_{i=1}^m (1 - FRR_{i,\omega}(t_i)). \quad (9)$$

In a conventional biometric system, FAR represents the security at the real-valued biometric representation level. In our system, since we derive a binary string as the output representation, it is necessary to consider the security at the binary string level as well. Thus ideally the entropy of the output string  $H(S_\omega)$  should be high, and the mutual information  $I(S_\omega; Q_\omega)$  between the output binary string and the published side-information should be zero [10].

For one-dimensional feature  $i$ , given the number of quantization intervals  $K_{i,\omega}$ , the way to achieve a high binary string entropy and a mutual information zero is to build the quantization according to (4), which means an equal probability  $P_\omega$  for each quantization interval. This requires a threshold  $t$  that gives  $FAR_{i,\omega} = 1/K_{i,\omega}$ . Under this condition, the binary string entropy  $H_i(S_{i,\omega})$  and its relation with  $FAR_{i,\omega}$  is given by (10). In our implementation, the wrap-around interval, with less than  $P_\omega$  probability mass for each of the left-end and right-end interval, will never be a genuine interval. Due to this effect, the mutual information is (11).

$$H_i(S_{i,\omega}) = \log_2 K_{i,\omega} = -\log_2 FAR_{i,\omega}, \quad (10)$$

$$I_i(S_{i,\omega}; Q_{i,\omega}) = \log_2 K_{i,\omega} - \log_2 (K_{i,\omega} - 1). \quad (11)$$

In the  $m$ -dimensional feature space  $\mathbb{V}^m$ , the  $m$  features are independent because of the PCA/LDA process. Hence, the binary string entropy and the mutual information becomes:

$$H = \sum_{i=1}^m H_i, \quad (12)$$

$$I = \sum_{i=1}^m I_i. \quad (13)$$

### D. Optimization

A good biometric system requires low  $FAR_\omega/FRR_\omega$  with high  $H$ . A well-defined method is to construct a *receiver operating characteristic* (ROC) curve based on all possible  $m$ -dimensional  $FAR_\omega$  and  $FRR_\omega$  [11]. Every point on the

ROC curve corresponds to a threshold vector  $\mathbf{T}$ . An optimal system can be found by minimizing the overall  $FRR_\omega$  given the  $FAR_\omega$  constraint:

$$\arg \min_{\mathbf{T}} (FRR_\omega(\mathbf{T})), \text{ given } FAR_\omega(\mathbf{T}) = \alpha. \quad (14)$$

The above optimization procedure needs a full range of  $\mathbf{T}$  vectors, while in our case, only some  $\mathbf{T}$  vectors are acceptable according to requirement (5). To solve this problem, we proposed a sub-optimal method. We will explain the detail of this method in section III-B.

## III. EXPERIMENTS AND RESULTS

To examine the performance of this likelihood ratio based quantization method, we conducted experiments on both synthetic and real data sets.

### A. Synthetic data experiments

We first carried out an experiment on the synthetic Gaussian data, with six methods: (1) likelihood ratio classifier (LC); (2) Zhang's multi-bits quantization (ZQ) [11]. In this method, each feature component is quantized with multiple intervals and each interval has the same fixed size ( $k\sigma$ ), where  $\sigma$  denotes the standard deviation of the genuine user pdf; (3) fixed one-bit quantization (FQ1) [10]. In this method, each feature component is quantized with 2 fixed intervals which have equally 0.5 background probability mass; (4) fixed two-bits quantization (FQ2). In this method, each feature component is quantized with 4 fixed intervals which have equally 0.25 background probability mass; (5) fixed three-bits quantization (FQ3). In this method, each feature component is quantized with 8 fixed intervals which have equally 0.125 background probability mass; (6) our likelihood ratio based multi-bits quantization (LQ).

We first performed a one-dimensional simulation on both a distinctive ( $\sigma = 0.2$ ) and a non-distinctive ( $\sigma = 0.8$ ) feature example. Fig. 2 shows the ROC performance of the overall user population. Our LQ method has the best FAR/FRR performance, the same as a likelihood ratio classifier. For fixed quantization FQ1, FQ2 and FQ3, it is not possible to tune any parameter, and their performance is worse than our LQ method. When the user within-class variation is small (e.g.  $\sigma = 0.2$ ), LQ has similar performance as ZQ, when the user within-class variation is large (e.g.  $\sigma = 0.8$ ), LQ outperforms ZQ.

We applied the LQ method on two-dimensional synthetic data, based on the assumption that the user within-class variance for the first two dimensions was  $\sigma_1 = 0.2$  and  $\sigma_2 = 0.8$  respectively. The optimal ROC curve was constructed by the process described in section II-D. Fig. 3 plots the two-dimensional overall ROC performance, and it suggests that the combined ROC curve constructed from our LQ method does not introduce a large degradation compared to the performance of LC.

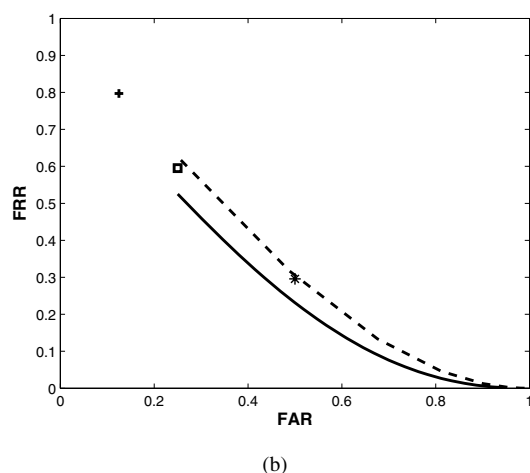
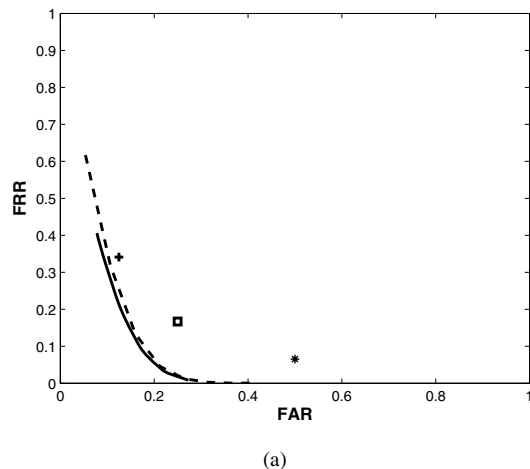


Fig. 2. One-dimensional simulation result: (a) Overall ROC with  $\sigma = 0.2$ ; (b) Overall ROC with  $\sigma = 0.8$ . ZQ (dashed); LQ and LC (solid); FQ1 (\*); FQ2 ( $\square$ ); FQ3 (+).

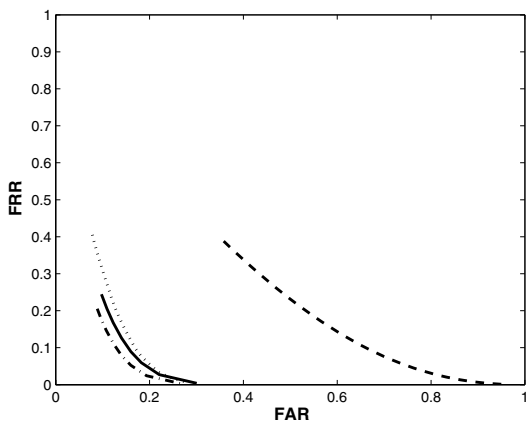


Fig. 3. Two-dimensional simulation result: ROC of the one-dimensional feature  $\sigma_1 = 0.2$  (dotted); ROC of the one-dimensional feature  $\sigma_2 = 0.8$  (dashed); ROC of the two-dimensional features  $\sigma_1 = 0.2$  and  $\sigma_2 = 0.8$  from LQ (solid); ROC of the same two-dimensional features from LC (dash-dotted).

## B. Real data experiments

The real data experiments were conducted on two data sets: a fingerprint data set FVC2000 (DB2) [16], [17] and a face data set FRGC (version 1) [18]. Both data sets were extracted into fixed length feature vectors.

- **FVC2000(DB2)**: This fingerprint data set contains 8 images of 110 different users. The original feature vector length extracted from the image was 1536 [10]. Features include the squared directional field and the Gabor response.
- **FRGC(ver1)**: This face data set contains variable images of 275 different users. The images were taken under controlled conditions and they were aligned using manually labeled landmarks. The original feature vector length extracted from the image was 8762. Features are the grey value of the face images.

The experiments consist of three steps: training, enrollment and verification. During the initial off-line training step, PCA/LDA was applied on the training data to reduce the feature dimension. Afterwards, an enrollment step was conducted in which the quantizers were constructed based on the enrollment data, in particular the means of the features after dimensionality reduction. The output reference binary string and the side-information were stored. In the verification step, verification data were quantized based on the quantizer side-information, and the output query string was compared to the reference string for the final decision. To split the data, 75% (FVC2000) and 50% (FRGC) of the samples per user were used for both training and enrollment, and the rest 25% (FVC2000) and 50% (FRGC) of the samples were used for verification. For both data sets, we extracted 50 features from their original measurements. To compare the query and the reference binary strings, we applied a Hamming distance classifier, in which the Hamming distance represents the number of different bits between the enrollment and verification binary string. The Hamming distance classifier replaces the ECC present in many template protection methods (e.g. [10]). Assigning a threshold  $D$  to the distance has the same effect as applying an ECC that can correct at most  $D$  bits. By varying the threshold  $D$ , a ROC curve on the verification data can be constructed. To obtain a reasonable error on the results, we repeated the above procedure with 20 random splits of enrollment and verification data.

We conducted two types of experiments. In the first experiment, we examined the feature extraction performance via the PCA/LDA process, followed by the FQ1 quantization. The result was compared to the reliable bits selection method proposed in [10], in which the output binary strings are selected directly from the original feature measurements, with a pre-selection based on the reliability of FQ1 results on each enrollment sample and a selection based on the ratio of within-class variation and between-class variation. Fig. 4 plots the log-ROC curves derived from both PCA/LDA method and reliable bits selection method. For both FVC2000 and FRGC, the performance increases dramatically with PCA/LDA. Such result suggests that features

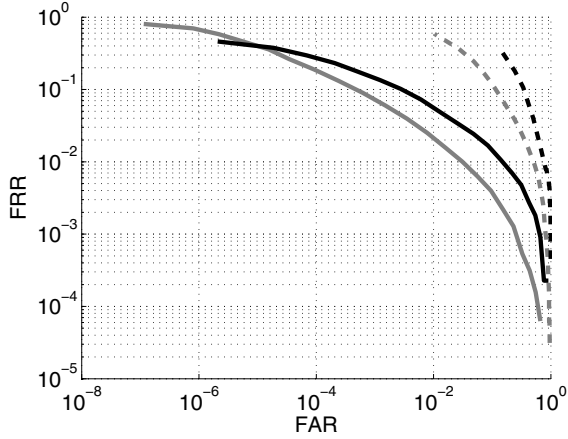


Fig. 4. Results of PCA/LDA feature extraction compared to the reliable bits selection method on FVC2000 (black) and FRGC (grey) (feature dimension for PCA is 100 and feature dimension for LDA is 50). Reliable bits selection method (dashed); PCA/LDA/FQ1 method (solid).

extracted from PCA/LDA method are more reliable and distinctive, which provides a crucial precondition for the upcoming quantization step.

In the second experiment, we examined the different quantization performances. To do a high-dimensional quantization experiment, we need to construct a ROC curve for high-dimensional features, but the optimization method described by (14) in section II-D is not feasible and constructing an optimal ROC curve is a point of further research. However, since a fixed length binary string as output is often preferred, we propose an alternative sub-optimal LQ $n$  method. The core idea is to quantize each feature dimension into  $n$  bits, which also means that the FAR per dimension is fixed to  $2^{-n}$ . As a result, the output string will have a fixed length.

We performed the experiments of LQ2 ( $n = 2$ ) and LQ3 ( $n = 3$ ) on both data sets, followed by the three-step procedure described above. The feature dimension after feature extraction was set to 50. Consequently, each user ended up with 100 and 150 bit string. (Note that the above likelihood ratio based quantization is user customized, which means each user has his own optimized quantization configuration.) Afterwards, we compared the LQ2 and LQ3 performance with FQ1, FQ2, FQ3 and LC methods.

Fig. 5 and 6 show the ROC plots for FVC2000 and FRGC data sets. It can be seen that results from all the methods are consistent on both data sets. LC is superior to all the quantization based methods. Apparently, FQ1, FQ3 and LQ3 do not provide comparable performance to LQ2 and FQ2. Compared to LQ2, FQ2 has a slightly worse performance. That means LQ2 consistently outperforms all the quantization methods, and its performance is not significantly degraded compared to the LC result. Table I lists the performance of LQ2 under different FAR/FRR requirements, compared to the LC performance. For a reasonable application requiring FAR =  $10^{-4}$ , the corresponding FRR are 16.7% (FVC2000) and 5.77% (FRGC) respectively, which

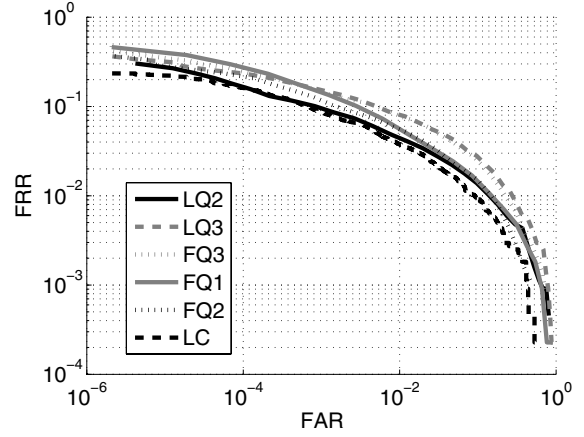


Fig. 5. Log-ROC curve of the fingerprint FVC2000 data.

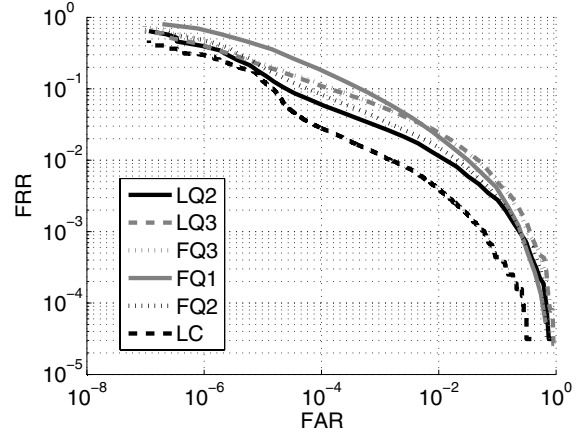


Fig. 6. Log-ROC curve of the face FRGC data.

is acceptable as compared to the performance of the LC classifier. The Hamming distance threshold needed to achieve such system performance is 29 from 100 bits for both data sets.

Now we analyze the security of the output binary string. Under the assumption of independent features, the output average string entropy for FQ2, LQ2, FQ3 and LQ3 are 100, 100, 150 and 150 respectively. However, in practice these numbers will be lower due to dependency of the individual features. The mutual information  $I$  between the output binary string and the side-information is zero for the FQ method,

TABLE I  
THE PERFORMANCE OF LC, LQ2 UNDER DIFFERENT SYSTEM REQUIREMENTS

	FAR = $10^{-2}$		FAR = $10^{-3}$		FAR = $10^{-4}$	
	FRR	D	FRR	D	FRR	D
FVC2000-LC	3.8%	N/A	8.7%	N/A	16.2%	N/A
FVC2000-LQ2	4.3%	37	8.7%	33	16.7%	29
FRGC-LC	0.41%	N/A	1.20%	N/A	2.80%	N/A
FRGC-LQ2	1.03%	37	2.60%	33	5.77%	29

but not zero for our LQ method. For instance, the mutual information for LQ2 is 0.415 bit per feature component. This can be viewed as a sacrifice of security since we introduced more user-specific information in the LQ quantization.

#### IV. DISCUSSION

The performance of the quantization methods is affected by two factors: the quality of the features and the quantization interval size. In our case, the quality of the features is defined as the within-class variation of each feature component after the PCA/LDA process, and the quantization interval size is driven by the number of quantization bits per feature dimension: quantization into 1 bit per feature (FQ1); quantization into 2 bits per feature (FQ2/LQ2) and quantization into 3 bits per feature (FQ3/LQ3). An investigation on the within-class variation of the feature components after PCA/LDA process demonstrates that for both FVC2000 and FRGC data sets, the within-class variance of the 50 features range from  $0.14^2$  to  $0.60^2$ . If FQ1 is applied, which has relatively large quantization intervals compared to the feature variation, the *FRR* per feature dimension is low. However, in this case the *FAR* of 0.5 per dimension is quite high. This results also in a high *FAR* in the high dimensional experiment (8). If FQ3 and LQ3 are applied, which have relatively small quantization intervals compared to feature variation, the *FAR* reduces to 0.125 per feature dimension. In contrast, the *FRR* per feature dimension will be high. This results in a high *FRR* in the high dimensional experiment (9). Therefore, FQ2 and LQ2 turn out to be a good compromise with respect to the *FAR/FRR* requirements. This explains why in Fig. 5 and Fig. 6, LQ2 and FQ2 outperforms FQ1, FQ3 and LQ3.

#### V. CONCLUSIONS

In this paper we discussed the problem of transforming biometric feature vectors into binary strings which are to be used in recently introduced methods for privacy protection of biometric information. We proposed to pre-process the feature vectors using a PCA/LDA transformation followed by a quantizer based on a likelihood ratio approach. Depending on the setting, our quantizer allows to extract multiple bits from a single feature. Comparison of our approach with a number of quantizers known from the literature, using both synthetic and real-life data, shows that the likelihood quantizer outperforms the other quantizers. Moreover, its performance is not significantly degraded as compared to a traditional likelihood classifier.

In our current experiments we extracted the same number of bits for every feature. In practice, however, not all features are equally distinctive. Therefore, an adaptive coding method, in which more bits are assigned to distinctive features and less bits to non-distinctive features, is a point of future research.

#### VI. ACKNOWLEDGMENTS

This research is supported by the research program Sentinels ([www.sentinels.nl](http://www.sentinels.nl)). Sentinels is being financed by Technology Foundation STW, the Netherlands Organization

for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

#### REFERENCES

- [1] Umüt Uludag, Sharath Pankanti, Salil Prabhakar, and Anil K. Jain. Biometric cryptosystems: Issues and challenges. *Proceedings of the IEEE*, 92(6):948–960, 2004.
- [2] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, 40(3):614–634, 2001.
- [3] Nalini K. Ratha, Sharat Chikkerur, Jonathan H. Connell, and Ruud M. Bolle. Generating cancelable fingerprint templates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(4):561–572, 2007.
- [4] Ari Juels and Madhu Sudan. A fuzzy vault scheme. *Des. Codes Cryptography*, 38(2):237–257, 2006.
- [5] Umüt Uludag, Sharath Pankanti, and Anil K. Jain. Fuzzy vault for fingerprints. In Takeo Kanade, Anil K. Jain, and Nalini K. Ratha, editors, *Fifth Int. Conf. on Audio- and Video-Based Biometric Person Authentication (AVBPA 2005)*, volume 3546 of *Lecture Notes in Computer Science*, pages 310–319. Springer-Verlag, 2005.
- [6] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In G. Tsudik, editor, *Sixth ACM Conference on Computer and Communications Security*, pages 28–36. ACM Press, 1999.
- [7] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology — Eurocrypt 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540. Springer-Verlag, 2004.
- [8] Jean-Paul M. G. Linnartz and Pim Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In J. Kittler and M. Nixon, editors, *Conference on Audio and Video Based Person Authentication*, volume 2688 of *Lecture Notes in Computer Science*, pages 238–250. Springer-Verlag, 2003.
- [9] Tom A. M. Kevenaar, Geert Jan Schrijen, Michiel van der Veen, Anton H. M. Akkermans, and Fei Zuo. Face recognition with renewable and privacy preserving binary templates. In *IEEE Workshop on Automatic Identification Advanced Technologies (AutoID 2005)*, pages 21–26. IEEE Computer Society, 2005.
- [10] Pim Tuyls, Anton H. M. Akkermans, Tom A. M. Kevenaar, Geert Jan Schrijen, Asker M. Bazen, and Raymond N. J. Veldhuis. Practical biometric authentication with template protection. In *AVBPA*, pages 436–446, 2005.
- [11] Yao-Jen Chang, Wende Zhang, and Tsuhan Chen. Biometrics-based cryptographic key generation. In *ICME*, pages 2203–2206, 2004.
- [12] Andrew R. Webb. Feature selection and extraction. In *Statistical Pattern Recognition*, chapter 9. John Wiley & Sons, LTD, second edition, 2002.
- [13] Raymond N. J. Veldhuis, Asker M. Bazen, Joost A. Kauffman, and Pieter H. Hartel. Biometric verification based on grip-pattern recognition. In *Security, Steganography, and Watermarking of Multimedia Contents*, pages 634–641, 2004.
- [14] Asker M. Bazen and Raymond N. J. Veldhuis. Likelihood-ratio-based biometric verification. *IEEE Trans. Circuits Syst. Video Techn.*, 14(1):86–94, 2004.
- [15] Martin Gardner. The binary gray code. In *Knotted Doughnuts and Other Mathematical Entertainments*, chapter 2. W. H. Freeman and Co., 1986.
- [16] Dario Maio, Davide Maltoni, Raffaele Cappelli, James L. Wayman, and Anil K. Jain. FVC2000: Fingerprint verification competition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(3):402–412, 2002.
- [17] FVC2000 fingerprint verification competition. <http://bias.csr.unibo.it/fvc2000/default.asp>.
- [18] NIST: FRGC face recognition grand challenge. <http://www.frvt.org/frgc/>.