

Preventing the Decodability Attack Based Cross-Matching in a Fuzzy Commitment Scheme

Emile J. C. Kelkboom, Jeroen Breebaart, *Member, IEEE*, Tom. A. M. Kevenaer, Ileana Buhan, and Raymond N. J. Veldhuis

Abstract—Template protection techniques are used within biometric systems in order to safeguard the privacy of the system's subjects. This protection also includes unlinkability, i.e., preventing cross-matching between two or more reference templates from the same subject across different applications. In the literature, the template protection techniques based on fuzzy commitment, also known as the code-offset construction, have recently been investigated. Recent work presented the decodability attack vulnerability facilitating cross-matching based on the protected templates and its theoretical analysis. First, we extend the theoretical analysis and include the comparison between the system and cross-matching performance. We validate the presented analysis using real biometric data from the MCYT fingerprint database. Second, we show that applying a random bit-permutation process secures the fuzzy commitment scheme from cross-matching based on the decodability attack.

Index Terms— Biometrics, data security, data privacy, secure biometric data, template protection.

I. INTRODUCTION

WHEN using an application based on biometrics, first a *reference template* is generated from the biometric sample provided in the enrolment phase for later use. In the authentication phase, a new biometric sample is acquired and compared with the reference template. Hence, the application requires this reference template for a successful authentication and, therefore, it needs to be stored. Basically, there are two options of storage, namely on a token carried by the subjects themselves or in a centralized database. The latter case is considered to be more convenient for the subjects. However, storing unprotected biometric reference templates in centralized databases for each application increases the privacy risk. For example, if these databases are compromised, an adversary could check the types

of applications or services a specific subject has subscribed to. In the literature, this is known as *cross-matching*.

Therefore, it is not a surprise that the ISO guidelines [1] dictate the avoidance of centralized databases if possible. Some known countermeasures to safeguard the privacy and security by enforcing some of the ISO guidelines are 1) the practice of *data separation*, where the most privacy sensitive information is stored on an individual smartcard or token, 2) the use of *data minimization* principles, 3) the use of *classical encryption* techniques such as DES, AES, RSA to augment the confidentiality or integrity of the reference template, and 4) the implementation of *template protection* which creates irreversible, renewable, and unlinkable reference templates, i.e., protected reference templates. In our work, we focus only on the template protection method.

In the literature, numerous template protection methods such as the *Fuzzy Commitment Scheme* (FCS) [2], *Helper Data System* (HDS) [3]–[5], *Fuzzy Extractors* [6], [7], *Fuzzy Vault* [8], [9], and *Cancelable Biometrics* [10] have been proposed, with the claim of preventing cross-matching. However, recently it was presented in [11] that fuzzy vaults were susceptible to cross-matching and [12] solved this issue by hardening the protected reference template using a secret key or password provided by the subject. The requirement of keeping the key or password secret, however, has a serious impact on the convenience of the biometric system.

In the FCS construction, also known as the code-offset construction, the binary vector extracted from the biometric sample is XORed with a randomly selected codeword resulting into auxiliary data that is stored as part of the protected template. Certain implementations of the Helper Data System, Fuzzy Extractors are based on this FCS construction. Possible cross-matching vulnerabilities for template protection systems based on the FCS construction are briefly discussed in [13] and are based on attack methods using exhaustive search. More recently, a new vulnerability known as the *decodability attack* has been published for the case when the FCS is based on linear error-correcting codes (ECCs). To the best of our knowledge, the cross-matching vulnerability of the FCS construction was first published by the presentation of Stoianov at the European Biometrics Forum (EBF) Biometric Encryption Seminar [14]. Cross-matching is made possible by simply checking whether decoding the XOR of two auxiliary data elements stored in different databases leads to a valid codeword. If it leads to a valid codeword, the two auxiliary data most likely belong to the same subject and is labeled as genuine. Therefore, this vulnerability is also known as the decodability attack. More recently, a theoretical analysis was presented in [15] where the authors determine the probability that the decodability attack incorrectly labels two auxiliary data

Manuscript received July 28, 2010; revised October 15, 2010; accepted October 24, 2010. Date of publication November 11, 2010; date of current version February 16, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Arun Ross.

E. J. C. Kelkboom is with Philips Research, 5656 AE, Eindhoven, The Netherlands (e-mail: Emile.Kelkboom@philips.com; Jeroen.Breebaart@philips.com; Ileana.Buhan@philips.com).

J. Breebaart is with Civolution, 5656 AE Eindhoven, The Netherlands (e-mail: Jeroen.Breebaart@civolution.com).

T. A. M. Kevenaer is with priv-ID, 5656 AE, Eindhoven, The Netherlands (e-mail: Tom.Kevenaer@priv-id.com).

I. Buhan is with Siemens IT Solutions and Services, 2712 PN, Zoetermeer, The Netherlands (e-mail: ileana.buhan@siemens.com).

R. N. J. Veldhuis is with the University of Twente, Fac. EEMCS, 7500 AE, Enschede, The Netherlands (e-mail: R.N.J.Veldhuis@utwente.nl).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIFS.2010.2091637

from different subjects as genuine under the assumption that across the whole population the bits of the binary vector are independent and uniform.

1) *Contributions*: As our first contribution, we extend the theoretical analysis from [15] and show the relationship between the cross-matching performance with the template protection system performance itself. Furthermore, we empirically evaluate the theoretical analysis using real biometric data from the MCYT fingerprint database and show that if no care is taken cross-matching based on the decodability attack is indeed possible. However, as our second contribution we will show that this vulnerability can be prevented by implementing a bit-permutation or shuffling randomization process on the binary vector. Consequently, the cross-matching performance is close to random.

The outline of this paper is as follows. In Section II, we briefly describe the FCS construction, present the properties of a linear error-correcting code (ECC), and discuss a probability estimation case extensively used in the remainder of this work. In Section III, we discuss the possible cross-matching attacks including the newly published decodability attack [14], [15]. In Section IV, we theoretically analyze both the cross-matching and template protection system performance and show their relationship. Validation of the theoretical performances are conducted in Section V using the MCYT fingerprint database. In Section VI, we show that a bit-permutation randomization process reduces the effectiveness of the decodability attack. Conclusions are given in Section VII.

II. PRELIMINARIES

The template protection scheme under consideration is known as the Fuzzy Commitment Scheme (FCS) from [2] and is based on an error-correcting code (ECC). We first discuss the notations related to the ECC and thereafter we present the FCS. Furthermore, we discuss the estimation of the probability mass function (pmf) of the number of bit errors when XORing two random binary vectors, which is extensively used in the remainder of this work.

A. Linear Error-Correcting Code

We denote a t_c -error linear binary error-correcting code as $[n_c, k_c, t_c]$, where n_c is the length of the codeword C , k_c the length of the message or key \mathbf{K} , and t_c the error-correcting capability.

The *ECC Encoder* (*Enc*) function converts the key $\mathbf{K} \in \{0, 1\}^{k_c}$ into its corresponding codeword $C \in \{0, 1\}^{n_c}$. The codebook \mathcal{C} is the set of all valid codewords of the ECC with cardinality $|\mathcal{C}| = 2^{k_c}$. As the distance function we use the Hamming distance denoted as $d_H\{\cdot, \cdot\}$ and the Hamming weight denoted as $\|\cdot\|$. The minimum distance of the codebook \mathcal{C} is $d = 2t_c + 1$; therefore, it can correct up to t_c bit errors. Because of the linearity property of the ECC it holds that the XOR operation between any pair of codewords leads to another codeword from the same codebook \mathcal{C} , namely $\forall C_i, C_j \in \mathcal{C}: C_i \oplus C_j = C_k$, with $C_k \in \mathcal{C}$. Furthermore, we define $W_{\mathcal{C}}$ to be the set of possible weights w of the codewords from \mathcal{C} , while the function $N_{\mathcal{C}}(w)$ returns the number of codewords n_w with weight w , with $\sum_{w \in W_{\mathcal{C}}} N_{\mathcal{C}}(w) = |\mathcal{C}|$.

TABLE I
EXAMPLES OF THE BCH ECC GIVEN BY THE CODEWORD (n_c) AND KEY (k_c) LENGTH, THE CORRESPONDING CORRECTABLE BITS (t_c), AND THE RELATIVE ERROR CORRECTING CAPABILITY t_c/n_c

n_c [bits]	k_c [bits]	t_c [bits]	t_c/n_c
31	6	7	22.6%
	11	5	16.1%
	16	3	9.7%
63	7	15	23.8%
	16	11	17.5%
	24	7	11.1%
127	8	31	24.4%
	22	23	18.1%
	36	15	11.8%

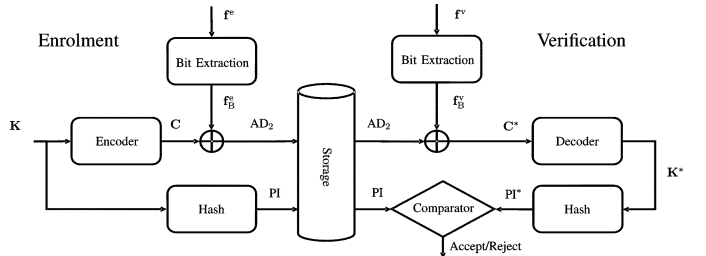


Fig. 1. Fuzzy commitment scheme (FCS) combined with a bit extraction module.

Given a word $w \in \{0, 1\}^{n_c}$ and the smallest distance to any codeword defined as $d_c(w, \mathcal{C}) \stackrel{\text{def}}{=} \min_{C \in \mathcal{C}} d_H(w, C)$, the *ECC Decoder* (*Dec*) function returns the key corresponding to the closest codeword from the codebook \mathcal{C} if the smallest distance $d_c(w, \mathcal{C})$ is smaller than or equal to the error-correcting capability t_c , i.e., $d_c(w, \mathcal{C}) \leq t_c$. When the smallest distance is larger than the error-correcting capability, $d_c(w, \mathcal{C}) > t_c$, then the word is not decodable and the *ECC Decoder* function either returns a decoding error or randomly selects a key.

In our experiments, we use the linear block type ECC “Bose, Ray-Chaudhuri, Hocquenghem” (BCH), with some $[n_c, k_c, t_c]$ settings given in Table I. For the BCH ECC, we use the maximum error-correcting capability t_c^* limited to around 25% of the codeword size n_c (see Table I), and if the word is not decodable it outputs the first k_c bits of the word as the key.

B. Fuzzy Commitment Scheme

The fuzzy commitment scheme (FCS) from [2] is one of the first template protection techniques and is based on the *bit commitment technique* known within the field of cryptography. The FCS works on discrete biometric data, while in practice most biometric data are continuous. Fig. 1 portrays the FCS construction combined with a bit extraction module.

In the enrolment phase, the real-valued column *feature vector* $\mathbf{f}^e \in \mathbb{R}^{N_F}$ is extracted from each N_e biometric enrolment sample by the feature extraction algorithm. From the N_e feature vectors, a single binary column vector $\mathbf{f}_B^e \in \{0, 1\}^{N_F}$ is created. For each component, we extract a single bit using a bit extraction scheme based on thresholding, where the mean of the background density is chosen as the threshold and estimated from a disjoint training set [3], [4], [16]. Prior to thresholding the mean of the N_e feature vectors is taken. Furthermore, a

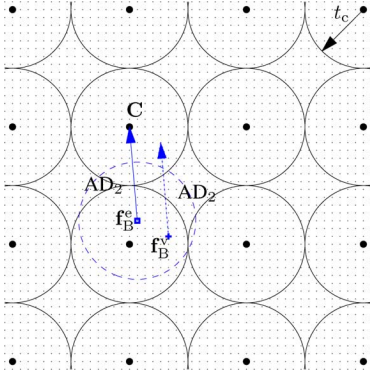


Fig. 2. Illustration of the FCS construction principles. The grid of small dots represent the word space $\{0, 1\}^{n_c}$, while the bigger dots represent the codewords from C with the error-correcting capability represented by the circles with radius t_c . AD_2 shifts the enrolment binary vector \mathbf{f}_B^e to the codeword C . In the verification phase, the same shift is applied to \mathbf{f}_B^v and will lead to a match if it is within the radius t_c of codeword C . Hence, all binary vectors \mathbf{f}_B^v within the dashed circle with radius t_c and center point \mathbf{f}_B^e will lead a match.

random key $\mathbf{K} \in \{0, 1\}^{k_c}$ is created and encoded by the *ECC Encoder* module into a codeword $C \in \{0, 1\}^{n_c}$ from \mathcal{C} . The fundamental property of the FCS is the XOR operation of the codeword C and the binary vector \mathbf{f}_B^e creating the offset AD_2 as helper data, $AD_2 = C \oplus \mathbf{f}_B^e$. The helper data AD_2 is also referred to as the *Auxiliary Data* in [17], in line with standardization activities in ISO [1]. Together with the hash of \mathbf{K} , also referred to as the *Pseudonymous Identifier* (PI), we obtain the protected template. As described in [2], \mathbf{f}_B^e is equivalent to the *witness* with which we commit the codeword C using the XOR operation considered to be similar to the one-time-pad encryption algorithm. The outcome of the commitment is the AD_2 and PI pair, which together is also known as the *blob*.

In the verification phase, the binary vector \mathbf{f}_B^v is created by quantizing the mean of the N_v verification feature vectors \mathbf{f}^v . Hereafter, the auxiliary data AD_2 is XORed with \mathbf{f}_B^v resulting in the possibly corrupted codeword $C^* = AD_2 \oplus \mathbf{f}_B^v = C \oplus (\mathbf{f}_B^e \oplus \mathbf{f}_B^v) = C \oplus e$, where the Hamming distance $\epsilon = d_H(\mathbf{f}_B^e, \mathbf{f}_B^v) = \|e\|$ indicates the number of errors corrupting the codeword C . Decoding C^* by the *ECC Decoder* module leads to the candidate key \mathbf{K}^* . The candidate pseudonymous identifier PI^* is obtained by hashing \mathbf{K}^* . A match is returned by the *Comparator* module if both PI and PI^* are equal, which occurs only when \mathbf{K} and \mathbf{K}^* are equal. Both secrets are equal when the Hamming distance between the binary vectors \mathbf{f}_B^e and \mathbf{f}_B^v is smaller or equal to the error-correcting capability of the ECC, $\epsilon = d_H(\mathbf{f}_B^e, \mathbf{f}_B^v) \leq t_c$. Hence, to successfully decommit the blob, a new witness \mathbf{f}_B^v has to be provided that is within t_c bit differences with the original witness \mathbf{f}_B^e .

An illustration of the code-offset is presented in Fig. 2, where the n_c -dimensional problem is simplified into a 2-D problem. The grid of small dots represent the word space $\{0, 1\}^{n_c}$, while the bigger dots represents the codewords from C with the error-correcting capability represented by the circles with radius t_c . The auxiliary data AD_2 shifts the enrolment binary vector \mathbf{f}_B^e to the codeword C . In the verification phase, the same shift is applied to \mathbf{f}_B^v and will lead to a match only if it is within the radius t_c of codeword C . Hence, all binary vectors \mathbf{f}_B^v within

the dashed circle with radius t_c and center point \mathbf{f}_B^e will lead to a match.

In this work, we consider two cases of the FCS, namely the *unbalanced* and *balanced system*. For the unbalanced system there are $N_e \neq N_v$ enrolment samples with N_v verification samples, while for the balanced case the number of verification samples is equal to the number of enrolment samples, $N_v = N_e$.

C. Hamming Weight After XORing Two Random Binary Vectors

In many derivations in the remainder of this work we need a solution to the following problem. Consider the case of having two words w_1 and w_2 randomly selected from $\{0, 1\}^{n_c}$ with weights w_1 and w_2 , respectively. Defining the number of bit errors or differences ϵ between w_1 and w_2 , namely $\epsilon = d_H(w_1, w_2)$, we are interested in the probability mass function (pmf) of ϵ .

Lemma II.1 (Hamming Weight After the XOR of Two Binary Vectors): Given two random binary vectors w_1 and w_2 with Hamming weight w_1 and w_2 , respectively, and defining $w_{\min} = \min(w_1, w_2)$, and $w_{\max} = \max(w_1, w_2)$, the number of possible bit errors $\epsilon = d_H(w_1, w_2)$ is given by the set $E = \{\epsilon_{\min}, \epsilon_{\min} + 2, \dots, \epsilon_{\max} - 2, \epsilon_{\max}\}$ with probability $P_{w \times w}(\epsilon; w_1, w_2, n_c)$ defined as

$$P_{w \times w}(\epsilon; w_1, w_2, n_c) \stackrel{\text{def}}{=} \begin{cases} 0, & \text{if } \epsilon \notin E \\ \frac{1}{\binom{n_c}{w_{\min}}} \binom{w_{\max}}{w_{\min} - \frac{\epsilon - \epsilon_{\min}}{2}} \binom{n_c - w_{\max}}{\frac{\epsilon - \epsilon_{\min}}{2}}, & \text{if } \epsilon \in E \end{cases} \quad (1)$$

where $\epsilon_{\min} = |w_1 - w_2|$, and $\epsilon_{\max} = n_c - |w_1 + w_2 - n_c|$.

Proof: Because w_1 and w_2 have w_1 and w_2 bits of value 1, respectively, the minimum number of possible errors equals the difference $\epsilon_{\min} = |w_1 - w_2|$. For example, let $w_1 > w_2$, i.e., $w_{\max} = w_1$ and $w_{\min} = w_2$, and the first w_1 bits of w_1 have a value 1 while the remaining $n_c - w_1$ bits have a value 0. The case with ϵ_{\min} errors can be obtained by allocating the w_2 bits of value 1 as the first bits of w_2 . Overall, there are $\binom{w_1}{w_2}$ possible combinations of having w_2 bits of value 1 of w_2 at locations where the bits of w_1 have a value of 1. Thus, the probability of having ϵ_{\min} errors is equal to the ratio of the number of possibilities with respect to the number of binary vectors of length n_c with weight w_2 , namely $\binom{w_1}{w_2} / \binom{n_c}{w_2}$.

Note that two bit errors are introduced if one bit of value 1 of w_2 is allocated where w_1 has a bit value of 0 instead of value 1. Hence, there are $\binom{w_1}{w_2 - 1} \binom{n_c - w_1}{1}$ possible combinations of introducing 2 bit errors. The first binomial coefficient $\binom{w_1}{w_2 - 1}$ is the number of possibilities of locating $w_2 - 1$ bits of value 1 of w_2 at the w_1 locations where w_1 has bits of value 1. The second binomial coefficient $\binom{n_c - w_1}{1}$ is the number of possibilities of allocating a single bit of value 1 of w_2 at the $n_c - 1$ locations where w_1 has a bit of value 0. Similarly, four bit errors are introduced when two bits of value 1 of w_2 are allocated where w_1 has a bit value of 0 with $\binom{w_1}{w_2 - 2} \binom{n_c - w_1}{2}$ possible combinations.

The maximum number of bit errors ϵ_{\max} is introduced by allocating all w_2 bits of value 1 of w_2 at locations where the bits of w_1 have a value 0. When $w_1 + w_2 > n_c$, the number of bits of w_1 of value 0 is smaller than the number of bits of w_2 of value 1, namely $n_c - w_1 < w_2$, because of the $w_1 > w_2$

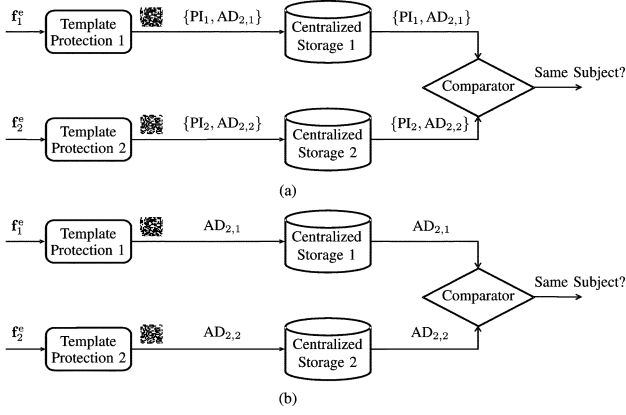


Fig. 3. Two cases of the cross-matching attack scenario between two application databases that are accessible by the adversary. In the first case (Case 1), both PI and AD₂ are stored in the centralized database. In the second case (Case 2), only AD₂ is stored in the centralized database accessible by the adversary, while AD₂ is assumed to be stored in a secure way and is not accessible by the adversary.

assumption. Consequently, the maximum number of bit errors is limited to $\epsilon_{\max} = n_c - |w_1 + w_2 - n_c|$. ■

III. CROSS-MATCHING ATTACKS

The setup of the cross-matching analysis is depicted in Fig. 3. We consider the scenario where there are two applications using the same biometric trait and identical template protection algorithms. Each application creates a protected template from independent enrolment samples of its subjects and stores it into its centralized database. We consider both centralized databases to be accessible by the adversary. Furthermore, we consider two cases differing on what is stored in the centralized database. In the first case, Case 1, both the auxiliary data AD₂ and the pseudonymous identifier PI are stored. Hence, the protected template for the first and second application is the pair $\{PI_1, AD_{2,1}\}$ and $\{PI_2, AD_{2,2}\}$, respectively. In the second case, Case 2, we consider only AD₂ to be stored in the centralized databases that are accessible, while PI may be stored within a personal storage device such as a smart-card which is not compromised. The adversary has access to all protected templates in both databases and tries to find subjects that are enrolled in both applications. Two protected templates, each taken from a different database, are compared by a *cross-matching classifier* in the *Comparator* module in order to determine whether they were derived from the same subject. The cross-matching classifier computes a cross-matching distance score s_{CM} on which to base its decision whether the two protected templates belong to the same subject (genuine) or not (imposter). The comparison between the protected templates of the same subject is referred to as a genuine comparison and between different subjects as an imposter comparison. The *false match rate* (FMR) at cross-matching α_{CM} is the rate of claiming two templates to be from the same subjects at an imposter comparison. The *false nonmatch rate* (FNMR) at cross-matching β_{CM} is the rate of claiming two templates to be from different subjects at a genuine comparison. Ideally, these error rates should be as large as possible.

In this section, we discuss several cross-matching classifier methods. We discuss the exhaustive search approach for Cases

1 and 2. We omitted the third possible case where only PI is stored in the centralized databases that are accessible by the adversary, because it can be easily shown that cross-matching is not possible. If the key could be derived from PI, they could still not be used for cross-matching because the keys were generated randomly within each application. Furthermore, we discuss the recently published method known as the decodability attack [14], [15], which is not based on an exhaustive search and only consists of an XOR and decoding operation by exploiting the linearity property of the ECC.

A. Exhaustive Search Attack

Given two protected templates, the exhaustive search type of the cross-matching attack relies on searching the complete codebook \mathcal{C} in order to determine whether the two protected templates belong to the same subject.

1) *Case 1: PI and AD₂*: Recall that the pseudonymous identifier PI is the hash of the randomly selected key \mathbf{K} . Because the PI is part of the protected template, a possible attack would be to search the key from the PI. Assuming that the probability of a collision is small, i.e., the probability that two different keys have the same hash value, the key leading to the hash value equal to PI can be found by searching the key space of $\{0, 1\}^{k_c}$ and taking its hash value. The enrolled binary vector \mathbf{f}_B^e can be obtained by computing the XOR of auxiliary data AD₂ and the codeword C corresponding to the obtained key \mathbf{K} , namely $\mathbf{f}_B^e = AD_2 \oplus C$. By performing this exhaustive search on each protected template, we obtain the binary vector $\mathbf{f}_{B,1}^e$ and $\mathbf{f}_{B,2}^e$ for the first and second application, respectively. As the cross-matching distance score s_{CM} we use the Hamming distance $s_{CM} = \epsilon_{CM} = d_H(\mathbf{f}_{B,1}^e, \mathbf{f}_{B,2}^e)$. On average only half of the key space has to be searched, hence the average effort of finding the key corresponding to PI is 2^{k_c-1} . Consequently, finding both keys separately only takes twice the effort, namely 2^{k_c} .

2) *Case 2: Only AD₂*: Because PI is not available, the distance measure has to be obtained from AD₂ only. By defining the XOR operation of the two auxiliary data as $AD_{\oplus} \stackrel{\text{def}}{=} AD_{2,1} \oplus AD_{2,2}$, we can rewrite AD_{\oplus} as

$$\begin{aligned} AD_{\oplus} &= AD_{2,1} \oplus AD_{2,2} \\ &= (\mathbf{f}_{B,1}^e \oplus C_1) \oplus (\mathbf{f}_{B,2}^e \oplus C_2) \\ &= (\mathbf{f}_{B,1}^e \oplus \mathbf{f}_{B,2}^e) \oplus (C_1 \oplus C_2) \\ &= e \oplus C_3 \end{aligned} \quad (2)$$

where $\mathbf{f}_{B,1}^e(C_1)$ and $\mathbf{f}_{B,2}^e(C_2)$ are the binary vectors (codewords) in the enrolment phase for application 1 and 2, respectively, e is the error pattern between the enrolment binary vectors, and we used the property of linear codes where the XOR of two codewords leads to another codeword from the same codebook. A graphical representation of the XOR operation is presented in Fig. 4. Hence, all possible error patterns can be computed by exhaustively taking the XOR of AD_{\oplus} with any codeword from \mathcal{C} , which is an effort of 2^{k_c} . As the cross-matching distance score s_{CM} we take the error pattern with the smallest Hamming weight, namely $s_{CM} = \min_{C \in \mathcal{C}} \|AD_{\oplus} \oplus C\|$. Note that it holds that $s_{CM} = \epsilon_{CM} = d_H(\mathbf{f}_{B,1}^e, \mathbf{f}_{B,2}^e)$ only when $\epsilon_{CM} = d_H(\mathbf{f}_{B,1}^e, \mathbf{f}_{B,2}^e) \leq t_c$, because in this case C_3 will lead

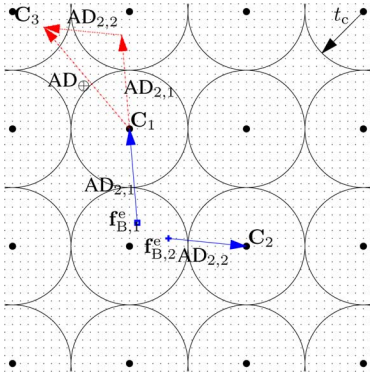


Fig. 4. Illustration of the XOR of $AD_{2,1}$ and $AD_{2,2}$ obtained from the enrolled binary samples $f_{B,1}^e$ and $f_{B,2}^e$ from the same subject.

to the smallest distance. For the case when $d_H(f_{B,1}^e, f_{B,2}^e) > t_c$ there is a probability that we obtain $s_{CM} \leq t_c$. Because the distance of AD_{\oplus} to C_3 is larger than t_c , there is a probability that another neighboring codeword is closer, due to the existence of multiple codewords at the minimum distance $d = 2t_c + 1$. The obtained cross-matching score is equal to $s_{CM} = \|e^*\| \leq t_c$ only if the error pattern can be rewritten as $e = e^* \oplus C_i$ with $\|e^*\| \leq t_c$ and $C_i \in \mathcal{C}$.

Note that when the codeword C_3 is known, it is not possible to derive the binary vectors $f_{B,1}^e$ and $f_{B,2}^e$, because the codewords C_1 and C_2 are not known. Because of the linear property of the ECC there are 2^{k_c} possible combinations of C_1 and C_2 that lead to C_3 . Hence, with this cross-matching attack, we obtain only a distance measure between the two enrolment binary vectors $f_{B,1}^e$ and $f_{B,2}^e$ but not their actual value.

The effort of determining the cross-matching distance score s_{CM} is case-dependent. If we obtained a cross-matching score s_{CM} smaller than t_c , the average effort of the corresponding cross-matching attack equals 2^{k_c-1} , because the search can be stopped once a score smaller than t_c has been obtained. When $s_{CM} > t_c$ then the complete codebook had to be searched and the effort is then 2^{k_c} .

B. Decodability Attack

The decodability attack method presented in both [14] and [15] is based on cross-matching with only AD_2 . For linear ECCs, they show that when AD_{\oplus} is decodable, the two presented auxiliary data are most probably derived from the same subject. More formally, if $\text{Dec}(AD_{\oplus})$ is successful, the cross-matching classifier outputs a match. From (2) we can derive that AD_{\oplus} is decodable when $\|e\| \leq t_c$ or $e = e^* \oplus C_i$ with $\|e^*\| \leq t_c$ and $C_i \in \mathcal{C}$. Hence, the decodability attack exploits the same underlying mechanism as shown in Case 2 in Section III-A and has, therefore, the same performance. However, the effort is significantly reduced towards a single decoding operation by using the decoding function of the ECC. Similarly, only a distance measure between the binary vectors $f_{B,1}^e$ and $f_{B,2}^e$ can be obtained but not their actual value.

IV. RELATING THE CROSS-MATCHING AND SYSTEM PERFORMANCE

In Section III, we presented several cross-matching attack methods from which the decodability attack is the most serious

one because of its reduced effort towards a single decoding operation of the ECC. In this section, we will determine the cross-matching classification performance in terms of the FMR and FNMR under the assumption that the subjects are in both databases. Furthermore, we compare the cross-matching performance with the system performance of the fuzzy commitment scheme. We assume the extracted bits to be independent with equal bit-error probability.

A. False Match Rate Relationship

Lemma IV.1 (FMR Relationship): Under the assumption that the bits of $f_B \in \{0,1\}^{n_c}$ across the population are independent and uniform and given a t_c -error binary linear ECC, the cross-matching and system FMR, α_{CM} and α_{TP} , respectively, at the error correcting threshold t_c are related according to $\alpha_{CM}(t_c, n_c) = 2^{k_c} \alpha_{TP}(t_c, n_c)$.

Proof: The false-acceptance rate for the template protection system α_{TP} depends on the probability mass function (pmf) of the Hamming distance $\epsilon = d_H(f_B^e, f_B^v)$ at imposter comparisons. As presented in [18], under the assumption that the bits of f_B^e across the population are independent and uniform, the imposter Hamming distance pmf can be modeled by the binomial density

$$P_b(d; N, p) \stackrel{\text{def}}{=} \binom{N}{d} p^d (1-p)^{(N-d)} \quad (3)$$

with dimension $N = n_c$ and bit-error probability $p = P_e^{\text{im}} = 1/2$, where P_e^{im} is the bit-error probability at imposter comparisons. Due to the single-bit extraction scheme employing a quantization threshold that is equal to the background mean, the bit-error probability P_e^{im} does not depend on either the number of enrolment N_e or verification N_v samples. Hence, the false-acceptance α_{TP} rate at threshold t_c is the following sum of the binomial pmf:

$$\begin{aligned} \alpha_{TP}(t_c, n_c) &\stackrel{\text{def}}{=} \sum_{i=0}^{t_c} P_b(i; n_c, P_e^{\text{im}}) \\ &= \frac{1}{2^{n_c}} \sum_{i=0}^{t_c} \binom{n_c}{i} \\ &= \frac{1}{2^{n_c}} V_2(n_c, t_c) \end{aligned} \quad (4)$$

where $V_2(n, r) = \sum_{i=0}^r \binom{n}{i}$ is the number of vectors in a sphere with radius r in $\{0,1\}^n$. An illustration of the binary vectors that will lead to a match at the verification phase is depicted in Fig. 2. Examples of $\alpha_{TP}(t_c, n_c)$ at several BCH ECC settings are given in Table II. Increasing the codeword size n_c decreases the FMR. Increasing the key size k_c and, therefore, decreasing the error-correcting capability t_c , also decreases the FMR.

As shown in Section III-B, the FMR of the cross-matching classifier α_{CM} is the probability that the XOR of the auxiliary data from two different subjects is decodable. As defined in [15], under the assumption that the bits of f_B are independent and uniform with $P_e^{\text{im}} = 1/2$, the α_{CM} is equal to the probability of randomly selecting a word $w \in_R \{0,1\}^{n_c}$ that is decodable, i.e., within t_c bits of any codeword from \mathcal{C} , namely

$$\alpha_{CM}(t_c, n_c) \stackrel{\text{def}}{=} \mathcal{P}\{d_c(w, \mathcal{C}) \leq t_c\} = \frac{2^{k_c} V_2(n_c, t_c)}{2^{n_c}}. \quad (5)$$

TABLE II
EXAMPLES OF α_{TP} AND α_{CM} FOR DIFFERENT $n_c \in \{127, 63, 31\}$ AND $[k_c, t_c]$ SETTINGS

$n_c = 127$				
$[k_c, t_c]$	[8, 31]	[22, 23]	[36, 15]	[78, 7]
α_{TP}	$3.16 \cdot 10^{-9}$	$8.48 \cdot 10^{-14}$	$7.89 \cdot 10^{-20}$	$5.57 \cdot 10^{-28}$
α_{CM}	$8.10 \cdot 10^{-7}$	$3.56 \cdot 10^{-7}$	$5.42 \cdot 10^{-9}$	$1.68 \cdot 10^{-4}$
$n_c = 63$				
$[k_c, t_c]$	[7, 15]	[16, 11]	[24, 7]	[45, 3]
α_{TP}	$1.88 \cdot 10^{-5}$	$8.37 \cdot 10^{-8}$	$6.82 \cdot 10^{-11}$	$4.52 \cdot 10^{-15}$
α_{CM}	$2.41 \cdot 10^{-3}$	$5.48 \cdot 10^{-3}$	$1.14 \cdot 10^{-3}$	$1.59 \cdot 10^{-1}$
$n_c = 31$				
$[k_c, t_c]$	[6, 7]	[11, 5]	[16, 3]	[26, 1]
α_{TP}	$1.66 \cdot 10^{-3}$	$9.61 \cdot 10^{-5}$	$2.32 \cdot 10^{-6}$	$1.49 \cdot 10^{-8}$
α_{CM}	$1.06 \cdot 10^{-1}$	$1.97 \cdot 10^{-1}$	$1.52 \cdot 10^{-1}$	1.00

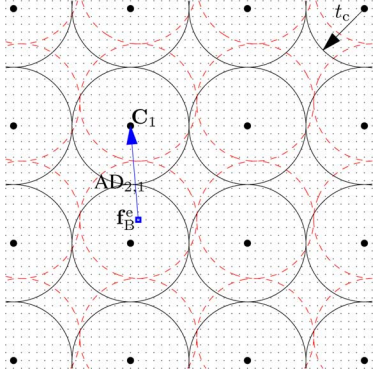


Fig. 5. Illustration of the binary vectors that would lead to a match.

An illustration of the binary vectors that will lead to a match is shown in Fig. 5. The $\alpha_{\text{CM}}(t_c, n_c)$ is equal to the ratio of all possible vectors within the dashed circles with respect to all possible vectors in the $\{0, 1\}^n$ space. Examples of $\alpha_{\text{CM}}(t_c, n_c)$ at some BCH ECC settings are given in Table II. Increasing the codeword size n_c decreases $\alpha_{\text{CM}}(t_c, n_c)$; however, increasing the key size k_c does not always decrease $\alpha_{\text{CM}}(t_c, n_c)$. Note the special case of $n_c = 31$ with $[k_c, t_c] = [26, 1]$, where $\alpha_{\text{CM}} = 1$ because the full $\{0, 1\}^{n_c}$ space is decodable. Thus, this $[n_c, k_c, t_c]$ setting of the BCH ECC coincides with the Hamming code which is known to be perfect.

By combing the system FMR α_{TP} from (4) and the cross-matching FMR α_{CM} from (5), we obtain

$$\alpha_{\text{CM}}(t_c, n_c) = 2^{k_c} \alpha_{\text{TP}}(t_c, n_c) \quad (6)$$

which implies that the cross-matching FMR is 2^{k_c} times larger than the system FMR under the assumption that the bits of \mathbf{f}_B^e across the population are independent and uniform. ■

B. False Nonmatch Rate Relationship

Lemma IV.2 (FNMR Relationship): Under the assumption that the bits of $\mathbf{f}_B \in \{0, 1\}^{n_c}$ are independent with equal bit-error probability P_e^{ge} , given a balanced system where $N_v = N_e$ and a t_c -error binary linear ECC, the cross-matching β_{CM} at the error correcting threshold t_c is smaller than the system FNMR β_{TP} , namely $\beta_{\text{CM}}(t_c, n_c) < \beta_{\text{TP}}(t_c, n_c)$.

Proof: For the template protection system, a false nonmatch occurs when $\epsilon = d_H(\mathbf{f}_B^e, \mathbf{f}_B^v) > t_c$ at genuine comparisons. Similar as in Section IV-A, we model the pmf of ϵ with a binomial density with dimension n_c , however, with bit-error

probability P_e^{ge} . The theoretical FNMR of the template protection system at threshold t_c , $\beta_{\text{TP}}(t_c, n_c)$, is the following sum of the binomial pmf:

$$\beta_{\text{TP}}(t_c, n_c) \stackrel{\text{def}}{=} \sum_{i=t_c+1}^{n_c} P_b(i; n_c, P_e^{\text{ge}}). \quad (7)$$

For the cross-matching classifier, β_{CM} is the probability that the XOR of the auxiliary data $\text{AD}_{2,1}$ and $\text{AD}_{2,2}$ from the same subject at different databases is not decodable, hence a non-match at a genuine comparison. As discussed in Section III-A, the decodability probability is determined by the Hamming distance between the binary vectors at enrolment, namely $\epsilon_{\text{CM}} = d_H(\mathbf{f}_{B,1}^e, \mathbf{f}_{B,2}^e)$. Because of the balanced system assumption the bit-error probability is also equal P_e^{ge} , consequently the pmf of $\epsilon_{\text{CM}} = d_H(\mathbf{f}_{B,1}^e, \mathbf{f}_{B,2}^e)$ is equal to the pmf of $\epsilon = d_H(\mathbf{f}_B^e, \mathbf{f}_B^v)$ and for convenience we use ϵ in the remainder of this section. As discussed in Section III, there is also a probability that when $\epsilon > t_c$, the XOR of the auxiliary data AD_{\oplus} will also be decodable and hence correctly labeled as genuine. We define the decodability probability $P_{\text{AD}_{\oplus}}(\epsilon; t_c, \mathcal{C})$ as

$$P_{\text{AD}_{\oplus}}(\epsilon; t_c, \mathcal{C}) \stackrel{\text{def}}{=} \mathcal{P}\{d_c(\text{AD}_{\oplus}, \mathcal{C}) \leq t_c \mid \epsilon\} \quad (8)$$

which has to be taken into account when estimating β_{CM} according to

$$\beta_{\text{CM}}(t_c, n_c) \stackrel{\text{def}}{=} \sum_{i=t_c+1}^{n_c} (1 - P_{\text{AD}_{\oplus}}(i; t_c, \mathcal{C})) P_b(i; n_c, P_e^{\text{ge}}). \quad (9)$$

Observe that $\beta_{\text{CM}}(t_c, n_c)$ from (9) is equal to $\beta_{\text{TP}}(t_c, n_c)$ from (7) when $(1 - P_{\text{AD}_{\oplus}}(\epsilon; t_c, \mathcal{C})) = 1$ for $\epsilon > t_c$. In other words, $P_{\text{AD}_{\oplus}}(\epsilon; t_c, \mathcal{C}) = 0$ stating that AD_{\oplus} should not be decodable for any cases of \mathbf{f}_B^e and \mathbf{f}_B^v with error pattern of weight $\epsilon > t_c$. However, $\beta_{\text{CM}}(t_c, n_c) < \beta_{\text{TP}}(t_c, n_c)$ if there is at least one case of \mathbf{f}_B^e and \mathbf{f}_B^v with error pattern of weight $\epsilon > t_c$ where AD_{\oplus} is decodable. Hence, it suffices to prove that there is at least one case of \mathbf{f}_B^e and \mathbf{f}_B^v with error pattern of weight $\epsilon > t_c$ where AD_{\oplus} is decodable.

Let the codebook be $\mathcal{C} = \{C_1, C_2, C_3\}$ with minimum distance $d = 2t_c + 1$, where the codewords C_1 and C_2 are used in the enrolment phase of applications 1 and 2, respectively, and $C_3 = C_1 \oplus C_2$. Note that the XOR of the auxiliary data can be rewritten as $\text{AD}_{\oplus} = (\mathbf{f}_{B,1}^e \oplus C_1) \oplus (\mathbf{f}_{B,2}^e \oplus C_2) = e \oplus C_3$ with $\epsilon = \|e\|$ and is decodable for the $\epsilon > t_c$ cases only if the error pattern can be rewritten as $e = e^* \oplus C_i$ with $\|e^*\| \leq t_c$ and $C_i \in \{C_1, C_2\}$. Hence, there are at least two cases where AD_{\oplus} with $\epsilon > t_c$ is decodable, namely the cases $\text{AD}_{\oplus} = C_1 \oplus C_3$ or $\text{AD}_{\oplus} = C_2 \oplus C_3$ where $\|e^*\| = 0$. ■

Lemma IV.2 only states that $\beta_{\text{CM}}(t_c, n_c) < \beta_{\text{TP}}(t_c, n_c)$ for any settings of t_c and n_c . In order to know the actual difference between $\beta_{\text{CM}}(t_c, n_c)$ and $\beta_{\text{TP}}(t_c, n_c)$, we have to determine $P_{\text{AD}_{\oplus}}(\epsilon; t_c, \mathcal{C})$ given a specific codebook \mathcal{C} . Assume we have an ECC with the codebook \mathcal{C} consisting of one codeword of weight 0 (C_0) and n_c (C_{n_c}) and n_w codewords C_w of weight w . Because of the properties of linear codes, each codeword has n_w neighbors at a distance w and one codeword at a distance n_c . Consider the case of being at codeword C_0 and having a binary vector w_ϵ with ϵ errors with respect to C_0 , hence having the weight w_ϵ . There are n_w neighboring codewords at a distance of w bits from C_0 , thus they have a weight of w . Furthermore,

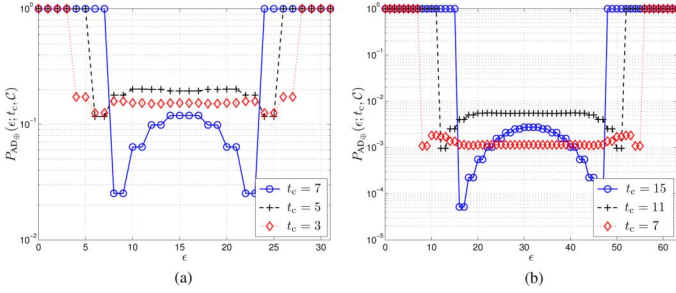


Fig. 6. $P_{AD_{\oplus}}(\epsilon; t_c, C)$ values for different t_c settings at $n_c \in \{31, 63\}$ (a) $n_c = 31$; (b) $n_c = 63$.

the error-correcting capability is equal to t_c . The fundamental question we want to answer is the decodability probability of the binary vector of w_ϵ . If its weight w_ϵ is within the error-correcting capability t_c , $w_\epsilon \leq t_c$, w_ϵ will always be decodable with respect to C_0 . However, if $w_\epsilon > t_c$, the binary vector w_ϵ will not be decodable with respect to C_0 but there is a probability that w_ϵ is decodable with respect to one of the n_w neighboring codewords at distance w . w_ϵ will only be decodable if its distance to the neighboring codewords is smaller or equal to t_c , i.e., $\|w_\epsilon \oplus C_w\| \leq t_c$. In Section II-C we have discussed the probability $P_{w \times w}(\epsilon; w_1, w_2, n_c)$ of the weight of the binary vector after XORing two binary vectors of length n_c and weights w_1 and w_2 , respectively. Hence, the decodability probability with respect to the n_w neighboring codewords of weight w is equal to $n_w \sum_{i=0}^{t_c} P_{w \times w}(i; w_\epsilon, w, n_c)$. Similarly, the decodability probability with respect to the codeword C_{n_c} has to be included, which is equal to $\sum_{i=0}^{t_c} P_{w \times w}(i; w_\epsilon, n_c, n_c)$.

For a general codebook \mathcal{C} , the decodability probability at ϵ errors, $P_{AD_{\oplus}}(\epsilon; t_c, \mathcal{C})$, is given by

$$P_{AD_{\oplus}}(\epsilon; t_c, \mathcal{C}) = \sum_{w \in W_{\mathcal{C}}} N_{\mathcal{C}}(w) \sum_{i=0}^{t_c} P_{w \times w}(i; \epsilon, w, n_c) \quad (10)$$

where $W_{\mathcal{C}}$ is the set of the unique weights w of the codewords from \mathcal{C} and the function $N_{\mathcal{C}}(w)$ returns the number of codewords n_w with weight w , with $\sum_{w \in W_{\mathcal{C}}} N_{\mathcal{C}}(w) = |\mathcal{C}|$. Some examples of $P_{AD_{\oplus}}(\epsilon; t_c, \mathcal{C})$ for the BCH code we consider are portrayed in Fig. 6 for $n_c \in \{31, 63\}$ and different t_c settings. From these figures, we can conclude that when $\epsilon \geq n_c - t_c$, AD_{\oplus} will always be decodable, because of the existence of a complementary codeword at distance n_c with respect to each codeword from \mathcal{C} . Furthermore, when $n_c = 31$ at most $\approx 20\%$ of the cases where $t_c < \epsilon < n_c - t_c$ are still decodable, which is significantly decreased to $\approx 0.6\%$ when $n_c = 63$. Some examples of β_{TP} and β_{CM} for different $n_c \in \{31, 63\}$ and $P_e^{ge} \in \{0.20, 0.15\}$ settings are given in Table III. There is no significant difference between β_{TP} and β_{CM} for the $n_c = 63$ case; however, there is a clear difference for the $n_c = 31$ case.

C. Performance Relationship

1) *Conjecture IV.1 (Performance Relationship)*: Under the assumption that the bits of $\mathbf{f}_B \in \{0, 1\}^{n_c}$ are independent with equal bit-error probability P_e^{ge} and $P_e^{im} = 1/2$ at genuine and imposter comparisons, respectively, given a balanced system where $N_v = N_e$, the cross-matching performance is worse than the system performance.

TABLE III
COMPARISON BETWEEN β_{TP} AND β_{CM} FOR DIFFERENT $n_c \in \{31, 63\}$, $[k_c, t_c]$, AND $P_e^{ge} \in \{0.15, 0.20\}$ SETTINGS

$n_c = 31$				
$[k_c, t_c]$		[6, 7]	[11, 5]	[16, 3]
$P_e^{ge} = 0.15$	β_{TP}	0.0822	0.3173	0.7039
	β_{CM}	0.0796	0.2749	0.5948
$P_e^{ge} = 0.20$	β_{TP}	0.2700	0.6069	0.8930
	β_{CM}	0.2598	0.5176	0.7592
$n_c = 63$				
$[k_c, t_c]$		[7, 15]	[16, 11]	[24, 7]
$P_e^{ge} = 0.15$	β_{TP}	0.0215	0.2287	0.7471
	β_{CM}	0.0215	0.2283	0.7460
$P_e^{ge} = 0.20$	β_{TP}	0.1789	0.6246	0.9527
	β_{CM}	0.1789	0.6231	0.9513

With Lemma IV.1 we showed that FMR between the cross-matching and system is related according to $\alpha_{CM}(t_c, n_c) = 2^{k_c} \alpha_{TP}(t_c, n_c)$, where the cross-matching FMR is 2^{k_c} worse than the system FMR. However, with Lemma IV.2 we showed that the FNMR at cross-matching is better than the system FNMR; however, the difference is marginal at larger codeword lengths. In order to compare the overall performance we use the *receiver operating characteristic* (ROC) curves as illustrated in Fig. 7 for the $n_c \in \{31, 63\}$ and $P_e^{ge} = 0.15$ settings. The system performance is given by the ROC labeled as TP_b , while the cross-matching performance is indicated by the points labeled with different markers representing the different $[k_c, t_c]$ settings of the ECC. Note that a performance is considered as being better when it is closer to the upper-left corner of the graph. Because the system ROC curve is clearly closer to the upper-left corner, we have shown that the system performance is better than the cross-matching performance.

V. EXPERIMENTS

In this section, we empirically estimate both the template protection system and cross-matching performance based on a fingerprint database in Sections V-B and V-C, respectively. The biometric database, feature extraction, and evaluation protocol are described in Section V-A.

A. Experimental Setup

1) *Biometric Modality and Database*: The database we use is the Ministerio de Ciencia y Tecnología (MCYT) containing fingerprint images from a capacitive and optical sensor as described in [19]. It contains 12 images of all 10 fingers from $N_s = 330$ subjects for each sensor. However, we limit our dataset to the images of the right-index finger from the optical sensor.

2) *Feature Extraction Algorithms*: In order to compensate for possible translations between the enrolment and verification measurements, a translation-only prealignment step is performed during the feature extraction process. Such prealignment requires extraction of the core point which is performed according to the algorithm described in [20]. Around the core point we define a 17×17 grid with eight pixels between each grid point. The feature extraction algorithm extracts a feature value on each grid point. Our feature extraction algorithm failed to extract a feature vector from a single subject, so we excluded

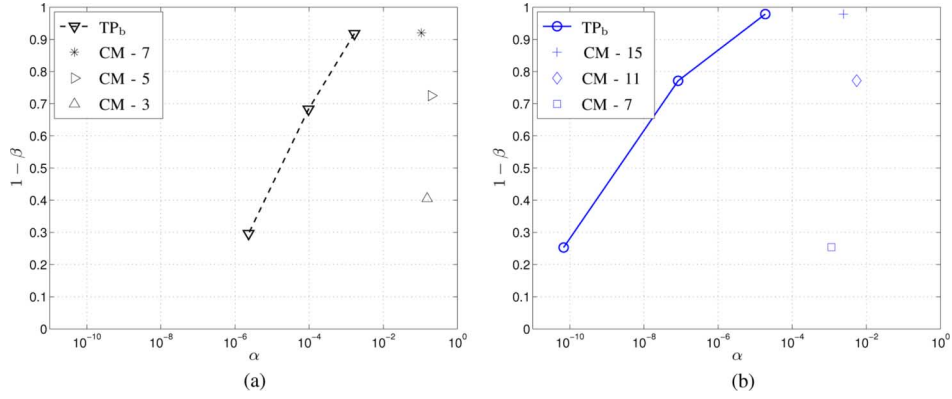


Fig. 7. Performance comparison between the template protection system (TP_b) and cross-matching performance (CM) for the (a) $n_c = 31$ and (b) $n_c = 63$ case, under the assumption of independent bits with bit-error probabilities $P_e^{im} = 0.5$ and $P_e^{ge} = 0.15$, and a balanced system $N_e = N_v$. The suffix indicates t_c .

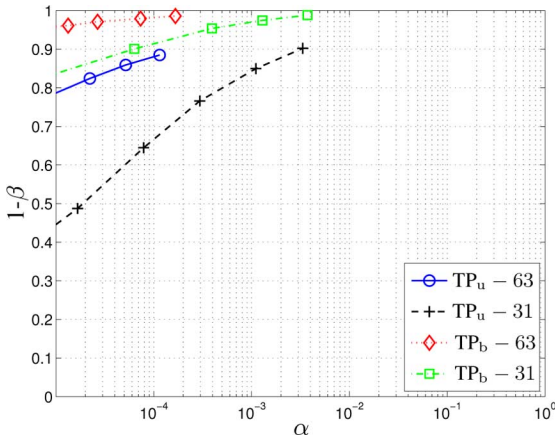


Fig. 8. ROC curve of the balanced and unbalanced and (TP_u) template protection system derived from the $s_{TP} = d_H(\mathbf{f}_B^e, \mathbf{f}_B^v)$ scores for the $n_c \in \{31, 63\}$ settings. For the balanced case, we have $N_e = N_v = 6$, while $N_e = 6$ and $N_v = 1$ for the unbalanced case.

it from the dataset, hence there are effectively $N_s = 329$ subjects.

The feature extraction method is based on the Gabor filter response, described in [21], where each grid point is filtered using a set of four 2-D Gabor filters at angles of $\{0, \pi/4, \pi/2, 3\pi/4\}$, respectively. The feature vector is the concatenation of the modulus of the four complex responses at each grid point, resulting into a feature vector dimension of $N_F = 1156$.

3) *Performance Evaluation Protocol*: The performance evaluation protocol consists of randomly selecting 219 out of $N_s = 329$ subjects as the training set and the remaining 110 subjects as the evaluation set, which is referred to as the training-evaluation-set split. To decorrelate the feature components we use the principle component analysis (PCA) and the linear discriminant analysis (LDA) techniques. The PCA and LDA transformation matrices are computed using the training set, where N_{PCA} is the reduced dimension after applying the PCA transformation and N_{LDA} is the reduced dimension after applying the LDA transformation. Furthermore, the template protection system parameters such as the quantization thresholds, used within the *Bit Extraction* module, are also estimated on the training set.

From the evaluation set we evaluate both the system and cross-matching classification performance.

- For the *system performance* evaluation, N_e samples of each subject are randomly selected as the enrolment samples while the remaining samples are considered as the verification samples. The protected template is generated using all the N_e enrolment samples and compared with disjoint groups of N_v verification samples where the mean of the feature vectors is taken prior to the bit extraction process.
- For the *cross-matching performance* evaluation, we randomly select N_e samples for the enrolment for the first application and another random N_e samples for the second application as such that we have distinct samples for each application. For each application we create the protected template and compare all protected templates using the cross-matching classifier.

This split of creating the enrolment and verification set or the enrolment set for application one and two is referred to as the enrolment-verification split. If the verification sample is from the same subject as of the protected template, it is referred to as a genuine comparison, otherwise it is an imposter comparison.

Both the training-evaluation-set and the enrolment-verification splits are performed five times. Note that the splits are performed randomly; however, the seed at the start of the protocol is always the same, hence all the splits are equal for the performance tests at different settings. Therefore, the splitting process does not contribute to any performance differences.

B. Template Protection System Performance

We evaluate the template protection system classification performance using the evaluation protocol in Section V-A3 with $N_e = 6$ and $N_v \in \{1, 6\}$. The case where $N_v = N_e$ is referred to as the balanced (TP_b) case and the unbalanced (TP_u) case when $N_v \neq N_e$.

The optimal N_{PCA} setting was found to be around 220 components and we set N_{LDA} equal to n_c to evaluate the performance. Note that we assume the FCS construction to act as a Hamming distance classifier as discussed in Section II, hence we actually evaluate the scores $s_{TP} = \epsilon = d_H(\mathbf{f}_B^e, \mathbf{f}_B^v)$ and limit the ROC curve at the threshold equal to t_c . The ROC curves for $n_c \in \{31, 63\}$ settings are portrayed in Fig. 8. The ROC curves are obtained by varying the k_c and t_c settings. For both n_c settings, the balanced case has a better performance because taking the average of N_v feature vectors suppresses the noise during

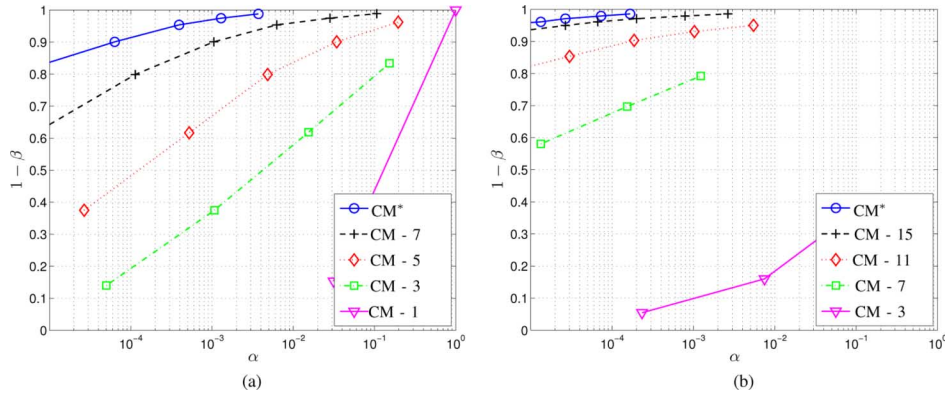


Fig. 9. ROC curve of cross-matching using AD_2 (CM) at different n_c and t_c indicated by the suffix. As reference, the ROC curve corresponding to $\epsilon_{CM} = d_H(\mathbf{f}_{B,1}^e, \mathbf{f}_{B,2}^e)$ is used and is labeled as CM*.

TABLE IV
EXPERIMENTALLY OBTAINED SYSTEM PERFORMANCE (α_{TP}, β_{TP})
AND CROSS-MATCHING PERFORMANCE (α_{CM}, β_{CM}) FOR DIFFERENT
 $n_c \in \{31, 63\}$ AND $[k_c, t_c]$ SETTINGS

	$n_c = 63$			
$[k_c, t_c]$	[7, 15]	[16, 11]	[24, 7]	[45, 3]
α_{TP}	$1.67 \cdot 10^{-4}$	$3.34 \cdot 10^{-6}$	≈ 0	≈ 0
α_{CM}	$2.64 \cdot 10^{-3}$	$5.47 \cdot 10^{-3}$	$1.23 \cdot 10^{-3}$	$1.58 \cdot 10^{-1}$
β_{TP}	$1.41 \cdot 10^{-2}$	$5.02 \cdot 10^{-2}$	$2.08 \cdot 10^{-1}$	$7.12 \cdot 10^{-1}$
β_{CM}	$1.42 \cdot 10^{-2}$	$4.98 \cdot 10^{-2}$	$2.08 \cdot 10^{-1}$	$5.89 \cdot 10^{-1}$
	$n_c = 31$			
$[k_c, t_c]$	[6, 7]	[11, 5]	[16, 3]	[26, 1]
α_{TP}	$3.72 \cdot 10^{-3}$	$3.94 \cdot 10^{-4}$	$3.34 \cdot 10^{-6}$	≈ 0
α_{CM}	$1.07 \cdot 10^{-1}$	$1.96 \cdot 10^{-1}$	$1.53 \cdot 10^{-1}$	1.00
β_{TP}	$1.20 \cdot 10^{-2}$	$4.62 \cdot 10^{-2}$	$2.01 \cdot 10^{-1}$	$6.25 \cdot 10^{-1}$
β_{CM}	$1.13 \cdot 10^{-2}$	$3.78 \cdot 10^{-2}$	$1.66 \cdot 10^{-1}$	0

verification which significantly improves the performance. Because of the BCH error-correcting limitation the FNMR is lower bounded and the FMR is upper bounded. The performance of the $n_c = 63$ case is better; however, the BCH limitation has a greater impact on the FNMR and FMR.

Note that the experimentally obtained α_{TP} , given in Table IV, for both the balanced and unbalanced case are very similar, however they deviate from the theoretical expectation presented in Section IV. Comparing Tables II and IV, the experimentally obtained α_{TP} at t_c is roughly an order of magnitude larger for the $n_c = 63$ case, while twice larger for the $n_c = 31$ case. We conjecture that the main cause of the deviation is the fact that the bits are still slightly dependent, while the theoretic work assumed independent bits. We omitted the $n_c = 127$ case due to the limited dataset with respect to its small theoretic FMR at the maximum error-correcting capability t_c^* , namely $\alpha_{TP}(t_c^* = 31, n_c = 127) \approx 3.16 \cdot 10^{-9}$.

C. Cross-Matching Performance Evaluation

As discussed in Section V-A3 for the cross-matching (CM) performance evaluation, we create two datasets containing the same subjects with $N_e = 6$ distinct samples of each subject. The two datasets represent the enrolment samples for the two applications. From each dataset we compute the binary vectors $\mathbf{f}_{B,1}^e$ and $\mathbf{f}_{B,2}^e$, and auxiliary data $AD_{2,1}$ and $AD_{2,2}$ from two randomly generated codewords C_1 and C_2 , respectively.

The cross-matching classifier from the decodability attack, as presented in Section III-B, is based on the property whether the

XOR of the auxiliary data $AD_{\oplus} = AD_{2,1} \oplus AD_{2,1}$ is decodable, i.e., $\text{Dec}(AD_{\oplus})$ is successful, where Dec is the ECC decoding function. When successful the classifier outputs a match, otherwise a nonmatch. The decoding function of the BCH ECC we use does not return an error when it is not decodable, but returns the first k_c bits of AD_{\oplus} as the key instead. Therefore, we compute the cross-matching distance score s_{CM} as

$$\begin{aligned} s_{CM} &= d_{CM}(AD_{2,1}, AD_{2,2}) \\ &= d_H(AD_{\oplus}, \text{Enc}(\text{Dec}(AD_{\oplus}))) \end{aligned} \quad (11)$$

where d_{CM} is the distance measure of the cross-matching classifier, and Enc and Dec are the encoding and decoding function of the BCH ECC, respectively. Consequently, we can extend the cross-matching classifier beyond the decision of either match or nonmatch with a score indicating how similar the comparison is.

The cross-matching performance ROC curves (CM) are depicted in Fig. 9 for $n_c = \{31, 63\}$ and different $[k_c, t_c]$ settings. Because of the availability of a score value instead of a decision, the ROC curves consist of multiple points instead of a single point as in Fig. 7, where the outmost right-upper point corresponds to the decision-based performance. The α_{CM} and β_{CM} values of these points are provided in Table IV. We also show the ROC curve from the Hamming distance of the enrolled binary vectors, $\epsilon_{CM} = d_H(\mathbf{f}_{B,1}^e, \mathbf{f}_{B,2}^e)$, indicated by CM*. Note that the CM* ROC curve is equal to the balanced system performance ROC curve TP_b from Fig. 8. Thus confirming the assumption made in Section IV-B that the pmf of $\epsilon = d_H(\mathbf{f}_B^e, \mathbf{f}_B^e)$ is equal to the pmf of $\epsilon_{CM} = d_H(\mathbf{f}_{B,1}^e, \mathbf{f}_{B,2}^e)$. Note that when comparing Tables II and IV, the experimentally obtained α_{CM} are close to their theoretical expectation. Also note that we do not observe the same order of estimation errors as for the case of the system performance α_{TP} .

With Fig. 9 we also experimentally validate Lemma IV.1 dictating that the cross-matching performance is always worse than the balanced system performance. Also note that the difference significantly increases when t_c is decreased and thus increasing k_c . However, the cross-matching performance can be better than the unbalanced system performance as shown by the comparison of the $TP_u - 31$ and $TP_u - 63$ ROC curves from Fig. 8 with the CM-7 and CM-15 curves from Fig. 9(a) and (b), respectively. Hence, designing a balanced system with $N_e = N_v$

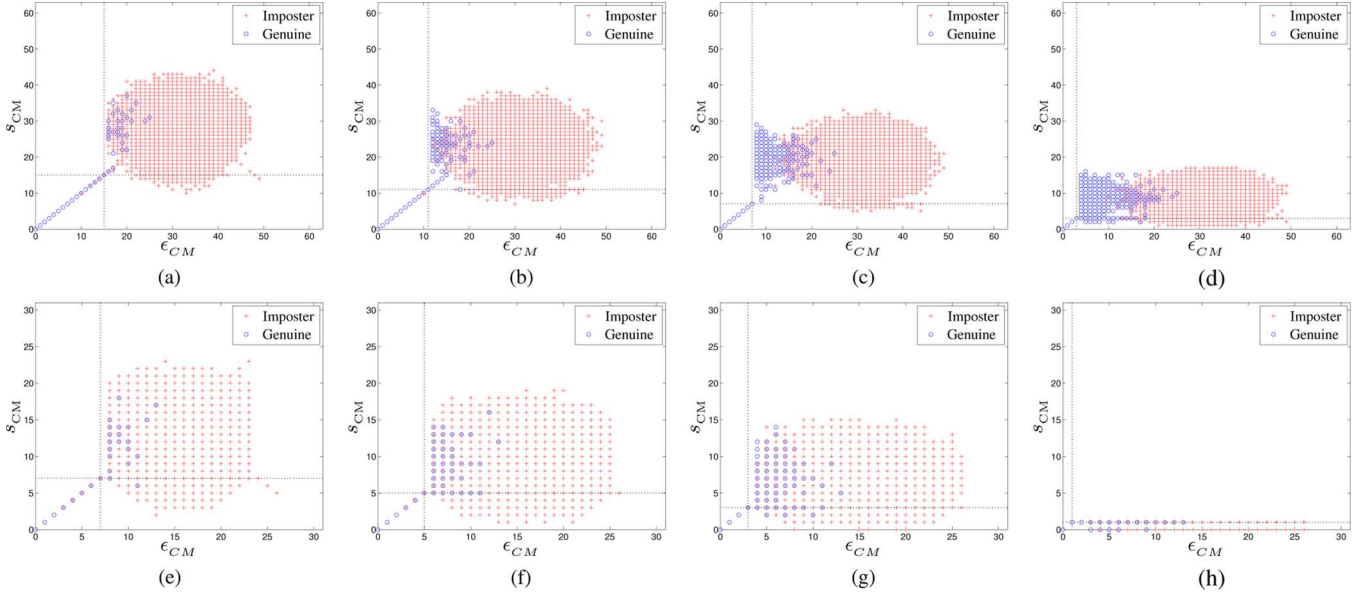


Fig. 10. Comparison between $\epsilon_{CM} = d_H(\mathbf{f}_{B,1}^e, \mathbf{f}_{B,2}^e)$ and $s_{CM} = d_{CM}(AD_{2,1}, AD_{2,2})$ for $n_c = \{31, 63\}$ and different $[k_c, t_c]$ settings (a) $n_c = 63, t_c = 15$, (b) $n_c = 63, t_c = 11$, (c) $n_c = 63, t_c = 7$, (d) $n_c = 63, t_c = 3$, (e) $n_c = 31, t_c = 7$, (f) $n_c = 31, t_c = 5$, (g) $n_c = 31, t_c = 3$, and (h) $n_c = 31, t_c = 1$.

guarantees that the cross-matching performance is always worse than the system performance itself.

Comparing Tables II and IV, for further analysis we show the comparison between the cross-matching Hamming distance ϵ_{CM} and distance score s_{CM} in Fig. 10. Note that the attacker only knows s_{CM} but not ϵ_{CM} . These figures illustrate that for both the genuine and imposter comparisons if $\epsilon_{CM} \leq t_c$ then $s_{CM} \leq t_c$. Furthermore, from the imposter comparisons, notably for the $n_c = 31$ case, we also observe that when $\epsilon_{CM} \geq n_c - t_c$ then it holds that $s_{CM} = n_c - \epsilon_{CM}$, because for each codeword there also exists its complementary one with a distance of n_c bits. For the case when $t_c < \epsilon < n_c - t_c$, AD_{\oplus} is occasionally decodable leading to a score $s_{CM} \leq t_c$ with probability $P_{AD_{\oplus}}(\epsilon; t_c, \mathcal{C})$ from (10) only when we can rewrite $(\mathbf{f}_{B,1}^e \oplus \mathbf{f}_{B,2}^e) = C_i \oplus e^*$ with $\|e^*\| \leq t_c$ and $C_i \in \mathcal{C}$.

Also note that the average of the scores s_{CM} , for the cases when AD_{\oplus} is not decodable and leading to a score $s_{CM} > t_c$, decreases when t_c decreases. Because of the systematic implementation of the BCH ECC and the fact that the decoding function of the ECC returns the first k_c bits as the key, guarantees that the first k_c bits between the corresponding codeword and AD_{\oplus} are always equal while the remaining bits will be random. Hence, the expected bit difference is equal to $n_c - k_c/2$.

VI. DECODABILITY ATTACK RESILIENCE WITH BIT-PERMUTATION RANDOMIZATION

We have shown that cross-matching is possible by using the decodability attack. However, if the system is designed as such that it is balanced, namely $N_e = N_v$, the cross-matching performance is always worse than the system performance, but still having a discriminating power. Ideally, it is preferred that the cross-matching performance is as close as possible to random.

In this section, we introduce a randomization module within the FCS construction rendering the cross-matching performance close to random. As illustrated in Fig. 11, prior to the XOR operation of the binary vector \mathbf{f}_B^e and the codeword, we randomize

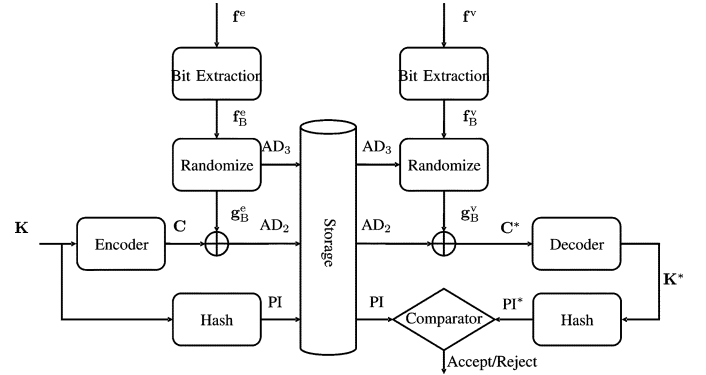


Fig. 11. Code-offset system with randomization.

\mathbf{f}_B^e by multiplying it with a bit-permutation matrix $A_{\pi} \in \Pi$, obtaining $g_B^e = A_{\pi} \mathbf{f}_B^e$, where A_{π} is a $n_c \times n_c$ matrix derived by randomly permuting the rows of the identity matrix and Π is the set of all possible permutation matrices. Because A_{π} is an orthogonal matrix, its inverse is equal to its transpose, $A_{\pi}^{-1} = A_{\pi}^T$. At each enrolment, a new randomly generated bit-permutation matrix is used and stored as auxiliary data AD_3 and is considered as public. It is important to note that in the current approach the randomization matrix A_{π} is not considered to be secret, which is in contrast to earlier methods such as [13].

The XOR of the auxiliary data AD_{\oplus} can now be rewritten as

$$\begin{aligned} AD_{\oplus} &= (g_{B,1}^e \oplus C_1) \oplus (g_{B,2}^e \oplus C_2) \\ &= (A_{\pi,1} \mathbf{f}_{B,1}^e \oplus A_{\pi,2} \mathbf{f}_{B,2}^e) \oplus (C_1 \oplus C_2) \\ &= e_{\pi} \oplus C_3 \end{aligned} \quad (12)$$

with $\epsilon_{\pi} = \|e_{\pi}\| = d_H(A_{\pi,1} \mathbf{f}_{B,1}^e, A_{\pi,2} \mathbf{f}_{B,2}^e) = d_H(g_{B,1}^e, g_{B,2}^e)$ being the number of errors after permutation instead of $\epsilon_{CM} = d_H(\mathbf{f}_{B,1}^e, \mathbf{f}_{B,2}^e)$ when no permutation has been applied. Because of the randomization process it is likely that at genuine comparisons more errors are introduced, namely $\epsilon_{\pi} > \epsilon_{CM}$,

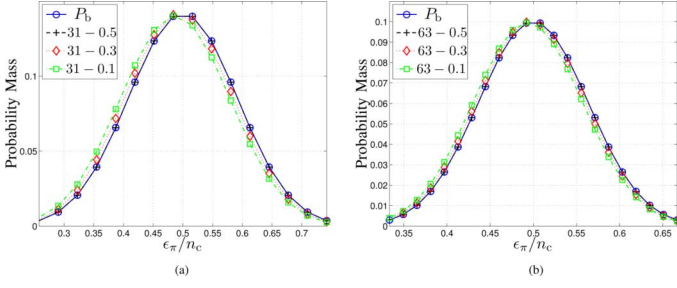


Fig. 12. The pmf of $\epsilon_\pi = d_H(g_{B,1}^e, g_{B,2}^e)$ from (13) at genuine comparisons for settings of $P_e^{\text{ge}} \in \{1/10, 3/10, 1/2\}$ and $n_c = \{31, 63\}$ compared with a binomial distribution $P_b(\epsilon_\pi; n_c, 1/2)$ (a) $n_c = 31$ (b) $n_c = 63$.

hence decreasing the probability that AD_\oplus is decodable, which significantly decreases when $\epsilon_\pi > t_c$ (see Fig. 6). As discussed in Section III, under the assumption of having independent bits with bit-error probability P_e^{ge} between genuine comparisons, the pmf of ϵ_π can be modeled by a binomial distribution with dimension n_c and $p = P_e^{\text{ge}}$, namely $P_b(\epsilon_\pi; n_c, P_e^{\text{ge}})$. However, the pmf of ϵ_π will depend on both the pmf of ϵ_π and on the effect of the permutation, which we will analyze further. When the weight of the binary vectors $\mathbf{f}_{B,1}^e$ and $\mathbf{f}_{B,2}^e$ are w_1 and w_2 , respectively, the probability of ϵ_π number of errors after randomizing them is thus equal to $P_{w \times w}(\epsilon_\pi; w_1, w_2, n_c)$ as discussed in Section II-C. Hence, the expected probability of ϵ_π irrespective of the weights $P_{\epsilon_\pi}(\epsilon_\pi; P_e^{\text{ge}}, n_c)$ is the average of $P_{w \times w}(\epsilon_\pi; w_1, w_2, n_c)$ across all possible weights. The possible combinations of w_1 and w_2 depend on the number of errors ϵ_{CM} between $\mathbf{f}_{B,1}^e$ and $\mathbf{f}_{B,2}^e$. If w_1 and ϵ_{CM} are known, then the probability of w_2 is determined by $P_{w \times w}(w_2; w_1, \epsilon_{\text{CM}}, n_c)$, because the error pattern can be considered as another binary vector of weight ϵ_{CM} . With the probability of randomly selecting a binary vector of weight w_1 equal to $P_b(w_1; n_c, 1/2)$, we obtain

$$\begin{aligned}
 & P_{\epsilon_\pi}(\epsilon_\pi; P_e^{\text{ge}}, n_c) \\
 & \stackrel{\text{def}}{=} \sum_{\epsilon_{\text{CM}}=0}^{n_c} \sum_{w_1=0}^{n_c} \sum_{w_2=0}^{n_c} P_{w \times w}(\epsilon_\pi; w_1, w_2, n_c) \\
 & \quad \times P_{w \times w}(w_2; w_1, \epsilon_{\text{CM}}, n_c) P_b\left(w_1; n_c, \frac{1}{2}\right) \\
 & \quad \times P_b(\epsilon_{\text{CM}}; n_c, P_e^{\text{ge}}). \tag{13}
 \end{aligned}$$

Fig. 12 portrays the pmf of ϵ_π at genuine comparisons obtained with (13) for different settings of $P_e^{\text{ge}} \in \{1/10, 3/10, 1/2\}$ and $n_c = \{31, 63\}$. As a reference, we use the case where e_π is a random binary vector with the pmf of its weight defined by the binomial pmf $P_b(\epsilon_\pi; n_c, 1/2)$. The figures show that the expected pmf of ϵ_π is very close to the case of being random, if either P_e^{ge} and n_c increases the difference becomes smaller. If $P_e^{\text{ge}} = 1/2$, the pmf of ϵ_π is equal to the case of being random.

Experimental results of the effects of the permutation randomization process, based on the same experimental setup from Section V, are shown in Fig. 13. We observe that the pmf of ϵ_π at genuine comparisons is close, however not equal, to the pmf at imposter comparisons, implying that it is difficult to distinguish a genuine comparison from an imposter comparison. These results confirm the theoretical expectations presented in

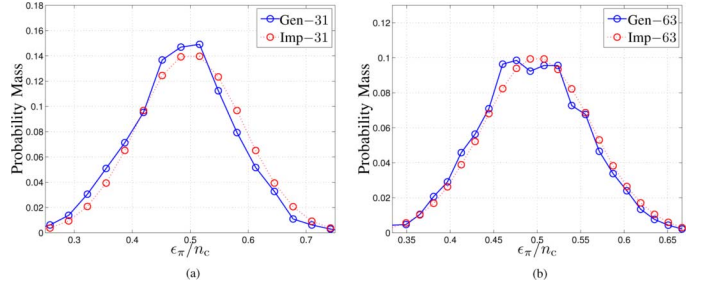


Fig. 13. The pmf of $\epsilon_\pi = d_H(g_{B,1}^e, g_{B,2}^e)$ at both the genuine (Gen) and imposter (Imp) comparisons for (a) $n_c = 31$ and (b) $n_c = 63$ settings.

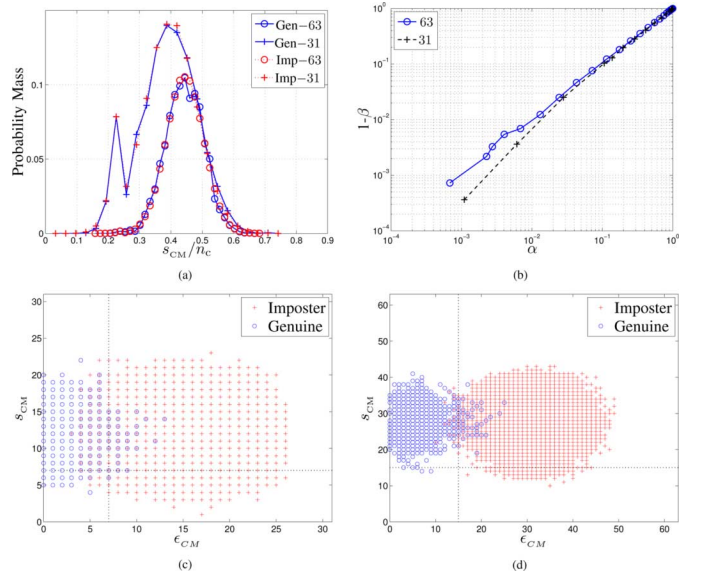


Fig. 14. (a) The pmf of s_{CM} and (b) the cross-matching ROC curve on logarithmic axes for $n_c \in \{63, 31\}$, and the comparison of s_{CM} against $\epsilon_{\text{CM}} = d_H(\mathbf{f}_{B,1}^e, \mathbf{f}_{B,2}^e)$ for (c) $n_c = 31$ and (d) $n_c = 63$, (a) s_{CM} pmf. (b) ROC curves (c) $n_c = 31, t_c = 15$, (d) $n_c = 63, t_c = 7$.

Fig. 12. Note that due to the fewer number of genuine comparisons than imposter comparisons, the pmf for the genuine case is more noisy.

Finally, the cross-matching performance with the randomization process is estimated based on the score s_{CM} from (11) and the results are shown in Fig. 14. Fig. 14(a) depicts the pmf of s_{CM} at genuine (Gen) and imposter comparisons (Imp) for the $n_c = \{31, 63\}$ settings. In contrast to the results in Fig. 9, we also include the scores larger than t_c . Both the genuine and imposter pmfs are very similar, hence no distinguishing performance can be extracted by the adversary. The cross-matching ROC curve for the $n_c = \{31, 63\}$ settings are shown in Fig. 14(b). As expected, the ROC curves are close to the one of a random classifier whose ROC curve is defined by $1 - \beta = \alpha$. Because of the limited genuine comparisons, the ROC curve for the $n_c = 63$ case looks to be a bit worse than the random classifier. Furthermore, the comparison between s_{CM} and $\epsilon_{\text{CM}} = d_H(\mathbf{f}_{B,1}^e, \mathbf{f}_{B,2}^e)$ are portrayed in Fig. 14(c) and (d) for the $n_c = 31$ and $n_c = 63$ case, respectively. Due to the bit-permutation randomization process, the relationship between s_{CM} and ϵ_{CM} (the straight line in the lower left quadrant), as observed in Fig. 10, no longer exists.

A. Inverting the Randomization Process

The randomization process and the bit-permutation matrix $A_{\pi,1}$ stored as auxiliary data AD_3 are considered as public. Hence, the adversary could apply the inverse on AD_2 , namely $A'_{\pi,1}AD_2$ with $A'_{\pi,1} = A_{\pi,1}^{-1}$, before applying the decodability attack on AD_{\oplus} . With the inverse process AD_{\oplus} becomes

$$\begin{aligned} AD_{\oplus} &= A'_{\pi,1}AD_{2,1} \oplus A'_{\pi,2}AD_{2,2} \\ &= A'_{\pi,1}(g_{B,1}^e \oplus C_1) \oplus A'_{\pi,2}(g_{B,2}^e \oplus C_2) \\ &= (A'_{\pi,1}A_{\pi,1}\mathbf{f}_{B,1}^e \oplus A'_{\pi,2}A_{\pi,2}\mathbf{f}_{B,2}^e) \\ &\quad \oplus (A'_{\pi,1}C_1 \oplus A'_{\pi,2}C_2) \\ &= (\mathbf{f}_{B,1}^e \oplus \mathbf{f}_{B,2}^e) \oplus (A'_{\pi,1}C_1 \oplus A'_{\pi,2}C_2) \end{aligned} \quad (14)$$

with $A'_{\pi}A_{\pi} = I$. Note that due to the inverse operation, additional errors may be introduced by the fact that both codewords are permuted by two different bit-permutation matrices, namely $(A'_{\pi,1}C_1 \oplus A'_{\pi,2}C_2) \in \mathcal{C}$. The additional errors guarantee that the cross-matching performance will be worse than the system performance. The only case where no errors are introduced is when $(A'_{\pi,1}C_1 \oplus A'_{\pi,2}C_2) \in \mathcal{C}$. We will show that this probability is very small, and thus there is a high probability that the cross-matching performance after taking the inverse is still worse than the system performance.

We will analyze this problem in two steps. First, given the codebook \mathcal{C} we estimate the probability of obtaining a binary vector of weight w from $(A'_{\pi,1}C_1 \oplus A'_{\pi,2}C_2)$, defined as $P_{\pi-1}(w; \mathcal{C})$. Hereafter, we estimate the probability that this binary vector is indeed a codeword, namely $P_{\pi-1}(\mathcal{C})$.

With $W_{\mathcal{C}}$ defined as the set of possible weights w of the codewords from \mathcal{C} and the function $N_{\mathcal{C}}(w)$ returning the number of codewords n_w with weight w with $\sum_{w \in W_{\mathcal{C}}} N_{\mathcal{C}}(w) = |\mathcal{C}| = 2^{k_c}$, the probability $P_{\pi-1}(w; \mathcal{C})$ is equal to

$$\begin{aligned} P_{\pi-1}(w; \mathcal{C}) &\stackrel{\text{def}}{=} \mathcal{P} \{w = \|A'_{\pi,1}C_1 \oplus A'_{\pi,2}C_2\| \\ &\quad | \forall C_1, C_2 \in \mathcal{C}, A_{\pi,1}, A_{\pi,2} \in \Pi\} \\ &= \sum_{\substack{w_1 \in W_{\mathcal{C}} \\ w_2 \in W_{\mathcal{C}}}} \left\{ P_{w \times w}(w; w_1, w_2, n_c) \frac{N_{\mathcal{C}}(w_1)N_{\mathcal{C}}(w_2)}{2^{2k_c}} \right\} \end{aligned} \quad (15)$$

where we take the sum, across all possible weights w_1 and w_2 of codewords C_1 and C_2 , of the product of $P_{w \times w}(w; w_1, w_2, n_c)$ from (1) which is the probability that the XOR of two random binary vectors of weights w_1 and w_2 will lead to a binary vector of weight w , and $N_{\mathcal{C}}(w_1)N_{\mathcal{C}}(w_2)/2^{2k_c}$ which is the probability of randomly selecting two codewords of weights w_1 and w_2 from \mathcal{C} . Fig. 15 illustrates $P_{\pi-1}(w; \mathcal{C})$ for different n_c and $[k_c, t_c]$ settings of the BCH ECC, compared with a binomial distribution $P_b(w; n_c, 1/2)$. Note that $P_{\pi-1}(w; \mathcal{C})$ is very similar to the binomial probability except at weights zero and n_c , where the difference increases when t_c increases. The weight, $w = \|A'_{\pi,1}C_1 \oplus A'_{\pi,2}C_2\|$ is zero when both $\|C_1\|$ and $\|C_2\|$ are zero or n_c , or equal to n_c when one of the codewords has weight

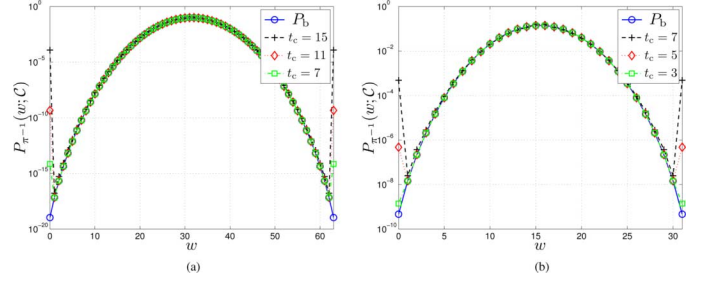


Fig. 15. Probability of obtaining a binary vector of weight $w = \|C_1 \oplus A_{\pi}C_2\|$ given by $P_{\pi-1}(w; \mathcal{C})$ from (15) for different n_c and t_c settings compared to a binomial distribution $P_b(\epsilon_{\pi}; n_c, 0.5)$. (a) $n_c = 63$. (b) $n_c = 31$.

TABLE V
PROBABILITY $P_{\pi-1}(\mathcal{C})$ AND $P_{\pi-1}(\mathcal{C} \setminus \{0, n_c\})$ FOR DIFFERENT SETTINGS OF n_c AND $[k_c, t_c]$

$n_c = 31$			
$[k_c, t_c]$	[6, 7]	[11, 5]	[16, 3]
$P_{\pi-1}(\mathcal{C})$	$9.7660 \cdot 10^{-4}$	$1.9103 \cdot 10^{-6}$	$3.0521 \cdot 10^{-5}$
$P_{\pi-1}(\mathcal{C} \setminus \{0, n_c\})$	$2.8424 \cdot 10^{-8}$	$9.5271 \cdot 10^{-7}$	$3.0517 \cdot 10^{-5}$
$n_c = 63$			
$[k_c, t_c]$	[7, 15]	[16, 11]	[24, 7]
$P_{\pi-1}(\mathcal{C})$	$2.4414 \cdot 10^{-4}$	$9.3133 \cdot 10^{-10}$	$1.8332 \cdot 10^{-12}$
$P_{\pi-1}(\mathcal{C} \setminus \{0, n_c\})$	$1.3555 \cdot 10^{-17}$	$7.1052 \cdot 10^{-15}$	$1.8190 \cdot 10^{-12}$

of zero and the other one n_c . Both cases have the probability $P_{\pi-1}(0; \mathcal{C}) = P_{\pi-1}(n_c; \mathcal{C}) = 2/2^{2k_c}$.

With $P_{\pi-1}(w; \mathcal{C})$, we can estimate the probability $P_{\pi-1}(\mathcal{C})$ of the occurrence where no additional errors are introduced when the adversary applies the inverse, namely

$$\begin{aligned} P_{\pi-1}(\mathcal{C}) &\stackrel{\text{def}}{=} \mathcal{P} \{ (A'_{\pi,1}C_1 \oplus A'_{\pi,2}C_2) \in \mathcal{C} \\ &\quad | \forall C_1, C_2 \in \mathcal{C}, \forall A_{\pi,1}, A_{\pi,2} \in \Pi \} \\ &= \sum_{w \in W_{\mathcal{C}}} P_{\pi-1}(w; \mathcal{C}) \frac{N_{\mathcal{C}}(w)}{\binom{n_c}{w}} \end{aligned} \quad (16)$$

where $N_{\mathcal{C}}(w)/\binom{n_c}{w}$ is the probability that the binary vector of weight w is a codeword. Some examples of $P_{\pi-1}(\mathcal{C})$ for different n_c and $[k_c, t_c]$ settings are given in Table V. At smaller k_c settings, $P_{\pi-1}(\mathcal{C})$ is close to $4/2^{2k_c}$, which is the probability of only selecting codewords of either weight zero or n_c . For those cases, no additional errors are introduced by $A'_{\pi,1}C_1 \oplus A'_{\pi,2}C_2$. The probability $P_{\pi-1}(\mathcal{C})$ can be reduced even further by removing these two codewords from the original codebook, thus obtaining the codebook $\mathcal{C} \setminus \{0, n_c\}$. The probability is then given by $P_{\pi-1}(\mathcal{C} \setminus \{0, n_c\})$ and its value for the same n_c and $[k_c, t_c]$ settings are given in Table V. At smaller k_c values, $P_{\pi-1}(\mathcal{C} \setminus \{0, n_c\})$ is significantly smaller than $P_{\pi-1}(\mathcal{C})$. Hence, in order to be more robust against the inverse of the bit-permutation process prior to the decodability attack, it is recommended not to use the codewords of weight zero or n_c . The drawback is that the key space is reduced to $2^{k_c} - 2$, which becomes negligible for larger k_c values. However, at larger k_c values both $P_{\pi-1}(\mathcal{C})$ and $P_{\pi-1}(\mathcal{C} \setminus \{0, n_c\})$ converge to each other. From the results of Fig. 15, we observe that at

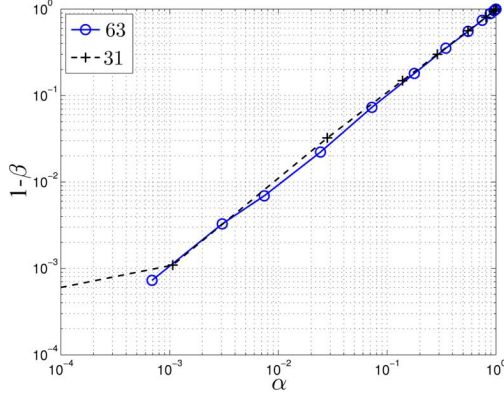


Fig. 16. Cross-matching ROC curve when applying the decodability attack after inverting the randomization process on logarithmic axes for the $n_c \in \{63, 31\}$ settings.

larger k_c values it holds that $P_{\pi^{-1}}(w; \mathcal{C}) \approx P_b(w; n_c, 1/2)$, consequently (16) becomes

$$\begin{aligned}
 P_{\pi^{-1}}(\mathcal{C}) &= \sum_{w \in W_{\mathcal{C}}} P_{\pi^{-1}}(w; \mathcal{C}) \frac{N_{\mathcal{C}}(w)}{\binom{n_c}{w}} \\
 &\approx \sum_{w \in W_{\mathcal{C}}} P_b\left(w; n_c, \frac{1}{2}\right) \frac{N_{\mathcal{C}}(w)}{\binom{n_c}{w}} \\
 &= \sum_{w \in W_{\mathcal{C}}} \frac{\binom{n_c}{w} N_{\mathcal{C}}(w)}{2^{n_c}} \\
 &= \frac{1}{2^{n_c}} \sum_{w \in W_{\mathcal{C}}} N_{\mathcal{C}}(w) \\
 &= 2^{k_c - n_c}
 \end{aligned} \tag{17}$$

which is the probability of randomly guessing a codeword from \mathcal{C} . Empirical results shown in Fig. 16 confirm that inverting the randomization process prior to applying the decodability attack does not give the adversary an advantage when using the decodability attack, because the ROC curve is still close to random.

B. Ineffectiveness of the Noise-Addition Randomization Method

We will show that not all randomization processes will work. For example, taking the XOR of \mathbf{f}_B^e with a random bit pattern δ , hence obtaining $g_B^e = \mathbf{f}_B^e \oplus \delta$ does not work, because this randomization process is fully reversible. When taking the XOR between $\text{AD}_{2,1}$ and $\text{AD}_{2,2}$, we obtain

$$\begin{aligned}
 \text{AD}_{2,1} \oplus \text{AD}_{2,2} &= (g_{B,1}^e \oplus C_1) \oplus (g_{B,2}^e \oplus C_2) \\
 &= ((\mathbf{f}_{B,1}^e \oplus \delta_1) \oplus (\mathbf{f}_{B,2}^e \oplus \delta_2)) \oplus (C_1 \oplus C_2) \\
 &= (\delta_1 \oplus \delta_2) \oplus (\mathbf{f}_{B,1}^e \oplus \mathbf{f}_{B,2}^e) \oplus (C_1 \oplus C_2).
 \end{aligned} \tag{18}$$

Hence, it is sufficient to take the XOR of the auxiliary data AD_2 with the publicly known bit pattern δ prior to applying the decodability attack, namely

$$(\delta_1 \oplus \text{AD}_{2,1}) \oplus (\delta_2 \oplus \text{AD}_{2,2})$$

$$\begin{aligned}
 &= (\delta_1 \oplus \delta_2) \oplus (\text{AD}_{2,1} \oplus \text{AD}_{2,2}) \\
 &= (\mathbf{f}_{B,1}^e \oplus \mathbf{f}_{B,2}^e) \oplus (C_1 \oplus C_2)
 \end{aligned} \tag{19}$$

because $(\delta_1 \oplus \delta_2) \oplus (\delta_1 \oplus \delta_2)$ cancel each other out. Hence, the adversary obtains the same error pattern $(\mathbf{f}_{B,1}^e \oplus \mathbf{f}_{B,2}^e) \oplus (C_1 \oplus C_2)$ with which cross-matching is possible, as shown in Section III.

C. Effect on the Exhaustive Search Attack

In Section III, we discussed both the decodability attack and the attacks based on exhaustive searches. With the bit-permutation process, we reduced the effectiveness of the decodability attack, however, both exhaustive attack methods still exist. With the bit-permutation process, the exhaustive search type of Case 1, where both the auxiliary data AD_2 and Pseudonymous Identifier PI are available, remains unchanged. By guessing the codeword from PI, the permuted binary vector g_B^e can be computed from which we can obtain \mathbf{f}_B^e by inverting the bit-permutation process with A_{π} . However, the exhaustive search type of Case 2, where only the auxiliary data is available, changes. The exhaustive search attack without the bit-permutation process as discussed in Section III-A has to search for a single codeword from the codebook \mathcal{C} leading to the smallest distance score $s_{\text{CM}} = \min_{C \in \mathcal{C}} \|\text{AD}_{\oplus} \oplus C\|$ with an average effort around $\approx 2^{k_c - 1}$. However, once the codeword was found there was still an ambiguity about the binary vector \mathbf{f}_B^e of 2^{k_c} possibilities. With the bit-permutation process, the XOR of the inverse of the auxiliary data of (2) becomes

$$\text{AD}_{\oplus} = (\mathbf{f}_{B,1}^e \oplus \mathbf{f}_{B,2}^e) \oplus (A'_{\pi,1} C_1 \oplus A'_{\pi,2} C_2) \tag{20}$$

where the linear property of the ECC no longer holds as in (2). Instead of searching the codebook \mathcal{C} only once, all combinations of the permuted codewords $A'_{\pi,1} C_1 \oplus A'_{\pi,2} C_2$ with known bit-permutation matrices has to be searched leading to the smallest distance score $s_{\text{CM}} = \min_{C_1, C_2 \in \mathcal{C}} \|\text{AD}_{\oplus} \oplus A'_{\pi,1} C_1 \oplus A'_{\pi,2} C_2\|$. Thus, the effort has significantly increased towards $\approx 2^{2k_c - 1}$. However, once the codewords C_1 and C_2 have been found, the binary vector \mathbf{f}_B^e is fully known. Hence, there is a trade-off between the case where cross-matching with the effortless decodability attack is possible with protection of the binary vectors or the case where cross-matching is possible with a significantly increased effort of $2^{2k_c - 1}$ but revealing the binary vectors at a successful cross-match.

VII. CONCLUSIONS

We analyzed the cross-matching performance of the auxiliary data AD_2 of the Fuzzy Commitment Scheme (FCS). We showed two attacks based on an exhaustive search, resulting in a significant attack effort, as well as a recently introduced attack requiring only a single decoding operation of the ECC, known as the decodability attack. Both attacks have the same cross-matching performance. To the best of our knowledge, the decodability attack was first presented in [14] and theoretically analyzed in [15]. We extended this theoretical analysis and showed the relationship between the balanced template protection system where $N_v = N_e$ and the cross-matching performance. The FMR at cross-matching is 2^{k_c} larger than the

FMR of the system, where k_c is the key size of the ECC. On the contrary, the FNMR at cross-matching is smaller than the FNMR of the system. However, the difference significantly decreases for larger n_c values. When comparing both the FMR and FNMR in an ROC curve, we showed that the cross-matching performance is clearly worse than the system performance. We empirically validated the presented theoretical analysis using real biometric data from the MCYT fingerprint database. Concluding, designing a balance template protection system with $N_v = N_e$ guarantees that the cross-matching performance is always worse than the system performance itself.

Ideally, the cross-matching performance should be close to random. We provided a solution based on a bit-permutation randomization process that reduces the cross-matching performance of the decodability attack very close to random under the assumption that independent samples are taken for each application. During the enrolment phase, a random bit-permutation matrix is generated and used to permute the binary vector prior to creating the auxiliary data. We can consider the bit-permutation matrix of the randomization process to be publicly known because we have shown that the cross-matching performance is still close to random even when inverting the bit-permutation randomization process.

We showed the following trade-off. Without the proposed bit-permutation randomization process, the decodability cross-matching attack is effortless, however, without revealing the enrolled binary vectors. With the bit-permutation randomization process, the decodability cross-matching attack is neutralized, however, cross-matching based on exhaustive search is still possible. The effort of the exhaustive search increased towards 2^{2k_c-1} , instead of 2^{k_c} , when the bit-permutation randomization process is not applied. However, the effort increase is obtained with a drawback, namely revealing the enrolled binary vectors at a successful cross-match.

REFERENCES

- [1] *Information Technology—Security Techniques—Biometric Template Protection*, ISO/IEC JTC1 SC27. FCD 24745, 2010.
- [2] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proc. 6th ACM Conf. Computer and Communications Security*, Nov. 1999, pp. 28–36.
- [3] E. J. C. Kelkboom, B. Gökberk, T. A. M. Kevenaar, A. H. M. Akkermans, and M. van der Veen, "3D face: Biometric template protection for 3D face recognition," in *Proc. Int. Conf. Biometrics*, Seoul, South Korea, Aug. 2007, pp. 566–573.
- [4] T. A. M. Kevenaar, G.-J. Schrijen, A. H. M. Akkermans, M. van der Veen, and F. Zuo, "Face recognition with renewable and privacy preserving binary templates," in *Proc. 4th IEEE Workshop on AutoID*, Buffalo, NY, Oct. 2005, pp. 21–26.
- [5] J.-P. Linnartz and P. Tuyls, "New shielding functions to enhance privacy and prevent misuse of biometric templates," in *Proc. 4th Int. Conf. AVBPA*, 2003, pp. 393–402.
- [6] E.-C. Chang and S. Roy, "Robust extraction of secret bits from minutiae," in *Proc. Int. Conf. Biometrics*, Seoul, South Korea, Aug. 2007, pp. 750–759.
- [7] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong secret keys from biometrics and other noisy data," in *Proc. Advances in Cryptology (Eurocrypt 2004, LNCS 3027)*, 2004, pp. 532–540.
- [8] A. Juels and M. Sudan, "A fuzzy vault scheme," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 237–257, Feb. 2006.
- [9] K. Nandakumar, A. K. Jain, and S. Pankanti, "Fingerprint-based fuzzy vault: Implementation and performance," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 4, pp. 744–757, Dec. 2007.
- [10] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle, "Generating cancelable fingerprint templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 561–572, Apr. 2007.
- [11] W. J. Scheirer and T. E. Boulton, "Cracking fuzzy vaults and biometric encryption," in *Proc. Biometric Symp.*, Baltimore, MD, Sep. 2007.
- [12] K. Nandakumar, A. Nagar, and A. K. Jain, "Hardening fingerprint fuzzy vault using password," in *Proc. 2nd Int. Conf. Biometrics*, Seoul, South Korea, Aug. 2007, pp. 927–937.
- [13] J. Bringer, H. Chabanne, and B. Kindarji, "The best of both worlds: Applying secure sketches to cancelable biometrics," *Sci. Comput. Program.*, vol. 74, pp. 43–51, Dec. 2008.
- [14] F. Carter and A. Stoianov, "Implications of biometric encryption on wide spread use of biometrics," in *Proc. EBF Biometric Encryption Seminar*, Amsterdam, The Netherlands, Jun. 2008 [Online]. Available: [http://www.eubiometricsforum.com/pdfs/be/BE-Carter Stoianov.pdf](http://www.eubiometricsforum.com/pdfs/be/BE-Carter%20Stoianov.pdf)
- [15] K. Simoons, P. Tuyls, and B. Preneel, "Privacy weaknesses in biometric sketches," in *Proc. 2009 IEEE Symp. Security and Privacy*, 2009, pp. 188–203, IEEE Computer Soc..
- [16] P. Tuyls, A. H. M. Akkermans, T. A. M. Kevenaar, G.-J. Schrijen, A. M. Bazen, and R. N. J. Veldhuis, "Practical biometric authentication with template protection," in *Proc. 5th Int. Conf. (AVBPA)*, Rye Brook, NY, Jul. 2005.
- [17] J. Breebaart, C. Busch, J. Grave, and E. Kindt, "A reference architecture for biometric template protection based on pseudo identities," in *Proc. BIOSIG*, Darmstadt, Germany, Sep. 2008.
- [18] J. Daugman, "The importance of being random: Statistical principles of iris recognition," *Pattern Recognit.*, vol. 36, no. 2, pp. 279–291, 2003.
- [19] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J. J. Igarza, C. Vivaracho, D. Escudero, and Q. I. Moro, "MCYT baseline corpus: A bimodal biometric database," in *Special Issue on Biometrics on the Internet, Proc. Inst. Elect. Eng. Vision, Image and Signal Processing*, Dec. 2003, pp. 395–401.
- [20] M. van der Veen, A. Bazen, T. Ignatenko, and T. Kalker, "Reference point detection for improved fingerprint matching," *Proc. SPIE*, pp. 60720G.1–60720G.9, 2006.
- [21] A. M. Bazen and R. N. J. Veldhuis, "Likelihood-ratio-based biometric verification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 86–94, Jan. 2004.



Emile J. C. Kelkboom was born in Oranjestad, Aruba, in 1980. He received the M.Sc. degree in electrical engineering from Delft University of Technology, The Netherlands, in June 2004. Since August 2006, he has been working toward the Ph.D. degree at Philips Research and the Department of Electrical Engineering Mathematics and Computer Science, University of Twente, The Netherlands.

From October 2004 to July 2006, he worked as an Application Engineer on CD, DVD, and Blu-ray drives within the Storage Engines Department of Philips Semiconductors. His focus is on safeguarding the privacy of the biometric information of subjects within biometric systems, namely the field of template protection. His research interests include biometrics, pattern recognition, signal processing, and security.

Mr. Kelkboom won the European Biometrics Forum (EBF) Research Award among Ph.D. students in Europe in 2009.



Jeroen Breebaart (M'07) received the M.Sc. degree in biomedical engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 1997, and the Ph.D. degree in auditory psychophysics from the same university in 2001.

From 2001 to 2007, he was with the Digital Signal Processing Group, Philips Research, conducting research in the areas of spatial hearing, parametric audio coding, automatic audio content analysis, and audio effects processing. Since 2007, he has been the leader of the biometrics cluster of the Information and System Security Group at Philips Research, expanding his research scope toward secure and convenient identification. He is a member of the AES. He contributed to the development of audio coding algorithms as recently standardized in MPEG and 3GPP such as HEAAC, MPEG Surround, and the upcoming standard on spatial audio object coding. He also actively

participates in the ISO/IEC IT security techniques standardization committee and is significantly involved in several EU-funded projects. He published more than 50 papers at international conferences and journals.



Tom A. M. Kevenaar received the M.Sc. and Ph.D. degrees from the Technical University of Eindhoven, The Netherlands, in 1986 and 1993, respectively.

As a postdoc, first at the University of Eindhoven and later at Hitachi Central Research Laboratory, Tokyo, Japan, he worked on several subjects in the field of design automation for analogue and RF circuits. In 1996, he joined Philips Research where he guided the development of a large variety of analogue design tools such as analogue/RF circuit simulators, optimization tools, fault simulation tools,

etc. In 2001, his attention moved to the field of cryptography with a focus on low-power cryptographic algorithms and standards and slightly later became involved in privacy enhancing technologies for biometric applications. In 2008, he cofounded a start-up company, which develops and sells products to protect the privacy of biometric information. He has over 40 publications in international conferences and journals and is coeditor of the book *Security with Noisy Data* (Springer, 2007).



Ileana Buhan received the Ph.D. degree from the University of Twente, The Netherlands, in 2008.

She has conducted research into security applications involving noisy data and secure spontaneous interaction. While at Philips Research in the Netherlands, she worked on developing techniques for the protection of biometric data. She is now a security consultant with Siemens IT Solutions and Services in The Netherlands.

In 2008, Dr. Buhan received the EBF European Biometric Research Industry Award for her work on combining secure spontaneous interaction with biometrics.



Raymond N. J. Veldhuis received the engineer degree in electrical engineering from the University of Twente, Enschede, The Netherlands, in 1981, and the Ph.D. degree from Nijmegen University, Nijmegen, The Netherlands, in 1988. His dissertation was titled "Adaptive restoration of lost samples in discrete-time signals and digital images."

From 1982 to 1992, he worked as a Researcher at Philips Research Laboratories, Eindhoven, The Netherlands, in various areas of digital signal processing, such as audio and video signal restoration and audio source coding. From 1992 to 2001, he worked at the Institute of Perception Research (IPO), Eindhoven, in speech signal processing and speech synthesis. From 1998 to 2001, he was program manager of the Spoken Language Interfaces research program. He is now an Associate Professor at the University of Twente, working in the fields of biometrics and signal processing. His expertise involves digital signal processing for audio, images and speech, statistical pattern recognition, and biometrics. He has been active in the development of MPEG standards for audio source coding.