FEHash: Full Entropy Hash for Face Template Protection

Thao M. Dang[1] Lam Tran[1] Thuc D. Nguyen[2] Deokjai Choi[1,*]
[1]Chonnam National University, Gwangju, South Korea [2]University of Science, VNU-HCMC, Vietnam

Abstract

In this paper, we present a hashing function for the application of face template protection, which improves the correctness of existing algorithms while maintaining the security simultaneously. The novel architecture constructed based on four components: a self-defined concept called padding people, Random Fourier Features, Support Vector Machine, and Locality Sensitive Hashing. The proposed method is trained, with one-shot and multi-shot enrollment, to encode the user's biometric data to a predefined output with high probability. The predefined hashing output is cryptographically hashed and stored as a secure face template. Predesigning outputs ensures the strict requirements of biometric cryptosystems, namely, randomness and unlinkability. We prove that our method reaches the `REQ-WBP` *(Weak Biometric Privacy) security level, which implies irreversibility. The efficacy of our approach is evaluated on the widely used CMU-PIE, FEI, and FERET databases; our matching performances achieve 100% genuine acceptance rate at 0% false acceptance rate for all three databases and enrollment types. To our knowledge, our matching results outperform most of state-of-the-art results.*

1. Introduction

Nowadays, the robustness of most security systems rely on the strength of cryptographic key (i.e., randomness, length). However, because it is challenging for human being to remember complicated string, user tends to use simple and meaningful password, or store it in somewhere, which can possibly be predicted or stolen by adversary. Meanwhile, various biometric traits (i.e., face [50], iris [27], signature [29]) have been found to contain individual unique pattern that can be used for user authentication or recognition. Authentication based on a concept of "who we are" is much more convenient than "what we remember" or "what we have". With the growing usage of biometric template, its protection becomes vital.

Since biometric data is permanently associated with the user and cannot be changed, the protection schemes must satisfy requirements of unlinkability and irreversibility [32].

Unlinkability: From the same biometric data, various versions of protected templates could be generated (i.e., renewability, cancelability, revocability). There is no correlation between transformed templates (i.e., independent, not cross-matching). In addition, transformed templates must not reveal any information about the original biometrics.

Irreversibility: It should be computationally hard to trace back the raw biometric data from the stored reference data (i.e., helper data, protected template). Ballard *et al*. [4] defined this requirement as `REQ-WBP` security level (*Weak Biometric Privacy*).

Because of intra-user variation property, biometrics cannot be handled effectively using information security techniques. Besides, this characteristic also leads to high false acceptance rate. Therefore, designing methods which satisfy both the requirements of security and performance is the main challenge in biometric template protection [18]. Several methods [30][1][45] take advantage of Convolutional Neural Network (CNN) models to enhance their matching performances. However, due to the nature of deep learning, their system security levels are partly unprovable.

1.1. Contribution

To tackle the above problem, we proposed a hashing function called Full Entropy Hash (FEHash) which constructed based on Locality Sensitive Hashing (LSH) [16]. Because of the *similarity preserving* nature of LSH, similar samples are more likely to have the same hash collision compared to dissimilar ones, so LSH can reduce the effect of the sampling variability issue of biometrics as well.

SimHash [6], a LSH based random projection method, uses a hyperplane to encode an input vector. The hashing function of SimHash is defined as follows:

$$h(\mathbf{x}) \triangleq sgn(\langle \mathbf{w}, \mathbf{x} \rangle + b). \tag{1}$$

That is, $h(\mathbf{x}) = \pm 1$ depending on which side of the hyperplane $\mathbf{x}$ lies, where vector $\mathbf{x} \in \mathbb{R}^d$, $(\mathbf{w}, b) \in \mathbb{R}^d \times \mathbb{R}$ is a random hashing hyperplane, and $\langle \cdot, \cdot \rangle$ denotes the inner product operator. Concatenating K distinct $h(\cdot)$ functions, we design a primitive system which produces a predefined hashing result of length K. However, any two of the random

*Corresponding author: Deokjai Choi {dchoi@jnu.ac.kr}

hyperplanes may be close to being linearly dependent, the resulting binary code may be less informative as it seems. An intuitive idea is collecting K hyperplanes from K different spaces. We projected biometric feature vectors into a new high-dimensional space. A hyperplane was selected based on the target output.

We introduced a concept called *padding people*, which ensures that projections of those hyperplanes are pairwise distinct in the input space. We used Random Fourier Features (RFF) [31] as a projection function in our scheme. Support Vector Machine (SVM) was used as a deterministic function to figure out the optimal hyperplane. Besides, we used deep CNN model to extract directly compact Euclidean feature vectors from face images; those vectors are inputs of our method.

To summarize, the main contributions of this paper are:

1. High security and privacy: The proposed scheme meets the requirements of unlinkability and irreversibility of biometric information protection (ISO/IEC FCD 24745) [17]. Furthermore, we proved that our method reaches the `REQ-WBP` [4] security level. We also provided a detailed security analysis of our method against available attacks.

2. Comprehensive evaluations: We fully implemented the proposed scheme and evaluated its performance on widely used face databases [37][44][28] in comparison with related works. Our experimental results indicated that FEHash is highly efficient. Specifically, we achieved 100% genuine acceptance rate (GAR) at 0% false acceptance rate (FAR) when using a matching method given in [30].

1.2. Related work

In general, template protection schemes can be classified as (i) feature transformation approach and (ii) biometric cryptosystem [18]. There were several works that combine face data with user specific key to obtain transformed template [34][42][41][43][22][7]. Biometric cryptosystems use cryptography-based approaches to achieve high security. These include Fuzzy commitment schemes in [46][25][2], Fuzzy vault in [48], and Fuzzy extractor in [39]. Feng *et al*. [12] proposed a hybrid approach that combines both (i) and (ii) to generate secure face template.

However, Fuzzy commitment schemes suffered poor error correcting capacity of short keys. In Fuzzy vault schemes, the genuine data was stored in the open between chaff points; Scheirer and Boult [35] listed some attacks against biometric fuzzy vaults. Besides, [5] and [21] implemented successfully attack frameworks to distinguish user data hidden among noisy points. Simoens *et al*. [38] proved that Fuzzy extractors can be broken by demonstrating how to link and reverse protected templates using the compromised helper data (i.e., code-offset δ). In addition, Apon *et al*. [3] pointed out that Fuzzy extractors violate the unlinkability if multiple independent helper data, generating from correlated inputs, are compromised.

On the computer vision side, CNN based algorithms like DeepFace [40] and FaceNet [36] have shown significant performance holding the state-of-the-art results for face recognition. We generally divide the recent works that used deep CNN models, to minimize intra-class variations and maximize inter-class variations, into two categories:

Chosen transformed template: Pandey *et al*. [30] assigned a distinct MEB code (Maximum Entropy Binary) to each user. Instead of generating a transformed template from biometric data, [30] predefined it. A shallow CNN model was used to learn to map the user face images to the corresponding binary code. The MEB code was cryptographically hashed to produce a protected template. [30] achieved high matching performance; however, the system was implemented for multi-shot enrollment only. Based on the idea of MEB code, Jindal *et al*. [1] improved the matching performance by using a deeper and better CNN model. [1] evaluated the system for both one-shot and multi-shot enrollment.

Generated transformed template: Talreja *et al*. [45] presented a method that integrated a deep hashing framework with a neural network decoder. Unlike [30] and [1], the binary codes were not predefined but generated as the outputs of the deep hashing component. The transformed templates were extracted from those generated binary codes. It led to the compatibility of the system with zero-shot enrollment. However, the system cannot provide the cancelability property because of this construction also.

In all three methods [30][1][45], the systems were trained end-to-end, which makes security levels rely on the strength of the cryptographic hash function only. Due to the characteristic of deep models, the security levels of the entire schemes are unprovable.

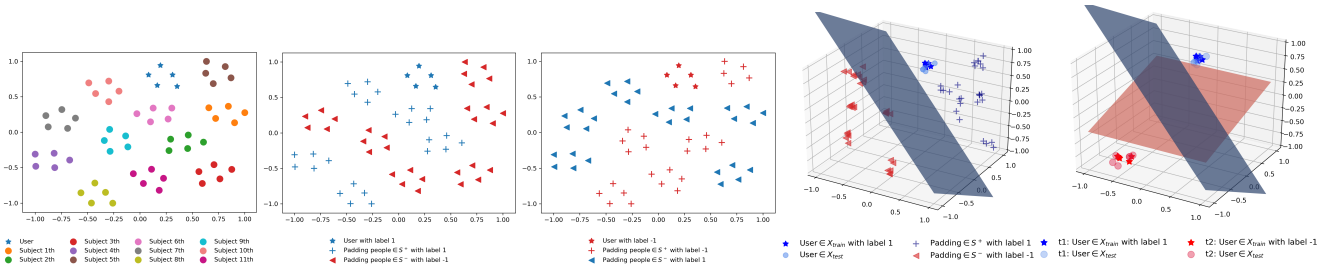
2. Methodology

2.1. Full Entropy Hash

To create FEHash function, we converted the task of finding hyperplane into binary classification problem. Particularly, we constructed a training set including feature vectors of user and other people. Some people had the same label (hashing result) with the user, the hyperplane divided two classes with respect to their labels. Figure 1 shows a toy illustration of FEHash.

Let us first define notations. Given a database containing N subjects, a set S has p elements that are indexes of those subjects, with $p \leqslant N$. $\mathcal{P}(S)$ is the power set of S, and the set of subsets of S of cardinality equal to q is denoted by $\mathcal{P}_q(S)$. A subset $S^+ \in \mathcal{P}_q(S)$ and S^- is the relative complement of S^+ in S. Formally speaking, $S^- = S \setminus S^+$, and $|S^-| = p - q$. In this paper, $p = 2q + 1$, $\{p, q\} \in \mathbb{N}^*$.

Padding people: Padding people is a set of p people



(a) 12 subjects of IMM (b) t1: $y_{genuine} = 1$ (c) t2: $y_{genuine} = -1$ (d) $F_1(\mathbf{x}_{genuine}) = 1$ (e) $F^2(\mathbf{x}_{genuine}) = 10$

Figure 1: Intuition of the hashing function FEHash. (a): t-SNE illustration of 12 subjects in the IMM-Frontal database [10]. (b and c): When training every hashing function, we assign labels to enroll biometric samples of user and padding people. (d): Those training data are projected into high dimensional space to find the optimal hashing hyperplane. (e): We hash user's feature vectors by using K hashing functions to obtain a determined $K-$length binary string (best view in color).

randomly chosen. The indexes of p subjects are elements of set S. We randomly choose q elements in S to establish two subsets S^+ and S^-. The people whose indexes $\in S^+$ have the same label with the user and vice versa.

Intuitively, different pairs of S^+ and S^- render different geometric shapes of positive and negative classes in the input space (Figure 1b, 1c), which implies that the boundary decision changes for each pair of S^+ and S^-. Therefore, we construct K classification problems to find K hyperplanes. To guarantee that there are at least K distinct pairs of S^+ and S^-, the value of p has a lower bound as:

$$\begin{pmatrix} p \\ q \end{pmatrix} \geqslant K, \tag{2}$$

where $\binom{p}{q}$ denotes the combination of p elements taken q at a time without repetition. However, the construction like this leads to the fact that the training set is not linearly separable (i.e., biometric data belonging to various people share a single label). Therefore, we need to project the non-linearly separable data to linearly separable data in higher dimensions, so data points belonging to different classes are allocated to different sides of classifier hyperplane.

Projection function: Random Fourier Features (RFF) [31] is used to project d-dimensional vector to higher D-dimensional vector space. Hence, the Equation 1 is updated to:

$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } sgn(\langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b) \geqslant 0 \\ -1 & \text{otherwise} \end{cases}, \tag{3}$$

where $\phi(\mathbf{x}) \triangleq \sqrt{\frac{2}{D}} \cos(\langle \Omega, \mathbf{x} \rangle + \mathbf{r})$, matrix $\Omega \in \mathbb{R}^{D \times d}$ is drawn i.i.d from Normal distribution, random vector $\mathbf{r} \in [0, 2\pi]^D$ is sampled uniformly.

Hashing function: We use Support Vector Machine (SVM) to determine the optimal hyperplane which has maximal margin between two classes. To avoid bias, we construct a complete balanced training set consisting of n instances. Let $X_{train} = \{\mathbf{x}_i\}_{i=1}^n$ be a set containing feature vectors of user and padding people; and $Y_{train} = \{y_i\}_{i=1}^n$ is a set of corresponding labels. The training set is designed with the constraint $\sum_{i=1}^n y_i = 0$, which means that the numbers of instances in each class are equal. Besides, SVM Linear under the $primal form$ [8] is used to prevent information leakage (i.e., the $dual form$ requires to store training data). The optimal hyperplane is found by solving the following problem:

$$\mathbf{w}^* = \underset{\mathbf{w}}{argmin} \|\mathbf{w}\|_2, \tag{4}$$

subject to $y_i(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) \geqslant 1$, with $i = 1, \ldots, n$, and release the maximal margin hyperplane:

$$b^* = -\frac{max_{y_i=-1}(\langle \mathbf{w}^*, \phi(\mathbf{x}_i) \rangle) + min_{y_i=1}(\langle \mathbf{w}^*, \phi(\mathbf{x}_i) \rangle)}{2}. \tag{5}$$

Putting it altogether, we define the FEHash function $F_{\mathbf{w},b,\Omega,\mathbf{r}} : \mathbb{R}^d \to \{0, 1\}$ via:

$$F_{\mathbf{w},b,\Omega,\mathbf{r}}(\mathbf{x}) \triangleq \frac{1}{2}[1 + h_{\mathbf{w},b}(\phi_{\Omega,\mathbf{r}}(\mathbf{x}))]. \tag{6}$$

From now on, we often omit the subscripts $\mathbf{w}$, b, Ω, $\mathbf{r}$ and write F, h, ϕ for the brevity. A set of K hashing functions is denoted as $F^K(\cdot) = (F_1(\cdot), \ldots, F_K(\cdot))$. Algorithm 1 summarizes the procedure of generating a single hashing function.

Noise rate: The genuine data are hidden among noisy padding data. Each individual gives a number of instances in X_{train}. Noise rate is the ratio of number of feature vectors belonging to user to the total number of instances in the training set.

$$NR = \left(1 - \frac{|X_{genuine}|}{|X_{train}|} \right) \times 100\%. \tag{7}$$

In this study, the default noise rate $NR_d = \frac{p}{p+1} \times 100\%$ happens when individuals contribute equally.

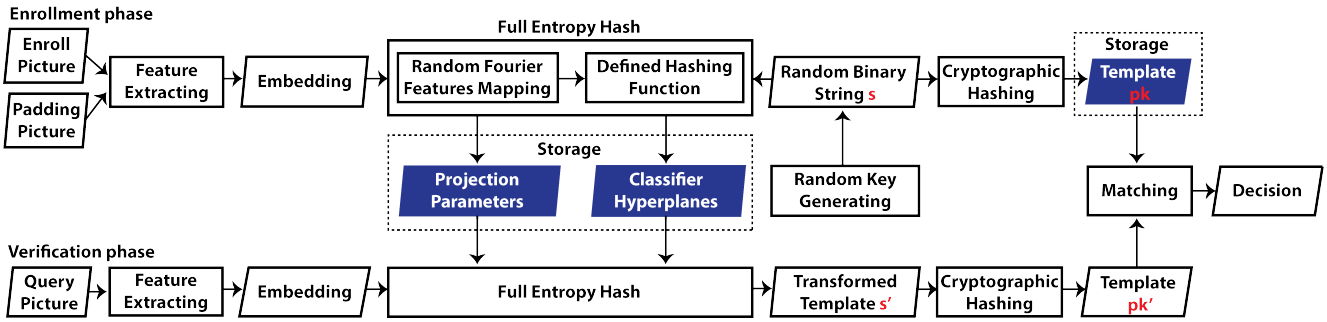


Figure 2: Block diagram of the proposed scheme to generate biometric protected template.

Algorithm 1. Generate $F_{\mathbf{w},b,\Omega,\mathbf{r}}$

Input:

A set of enrolled instances $X_{train} = \{\mathbf{x}_i\}_{i=1}^n$ and its corresponding label $Y_{train} = \{y_i\}_{i=1}^n$, where:
$X_{train} \in \mathbb{R}^{n\times d}, \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{-1, 1\}$,
$\sum_{i=1}^n y_i = 0$ *#balanced training set constraint*

Output:

Projection function parameters: $\Omega \in \mathbb{R}^{D\times d}, \mathbf{r} \in \mathbb{R}^D$.
Hashing plane parameters: $\mathbf{w} \in \mathbb{R}^D, b \in \mathbb{R}$.

Workflow:

$\Omega \sim \mathcal{N}(0, I)$;
$\mathbf{r} \leftarrow_{\$} [0, 2\pi]$;
$X_{train} \in \mathbb{R}^{n\times D} \leftarrow \phi(X_{train})$;
$\mathbf{w}, b \leftarrow \underset{\mathbf{w},b}{argmin} \|\mathbf{w}\|_2$;

return $F_{\mathbf{w},b,\Omega,\mathbf{r}}$;

2.2. Protected template

We proposed a novel scheme generating the protected template in Figure 2. The top pipeline demonstrates the enrollment phase; the bottom pipeline shows the verification phase. Our scheme is generic and could be applied to multiple biometrics traits (i.e., face, iris, fingerprint).

Enrollment phase: Facial images of user and padding people are fed to feature extractor to produce biometric feature vectors (which can interchangeably be called embedding or data point). A unique binary string **s** is assigned randomly to the user. Next, the system iterates Algorithm 1 K times to obtain a set of hashing functions F^K. Finally, the system cryptographically hashes string **s** to get the protected template **pk** using SHA3-512.

Verification phase: Since there are two matching approaches were used in this paper to achieve a verification decision (Section 3.3), we outline the main flow in this phase as follows. A new facial image is fed through the feature extractor, which outputs an embedding $\mathbf{x}'$. This embedding is then fed through the stored FEHash to obtain the binary code $\mathbf{s}' = F^K(\mathbf{x}')$. After that, the template $\mathbf{pk}'$ is reproduced via $\mathbf{pk}' = SHA(\mathbf{s}')$. Based on the similarity nature of LSH, the probability of $\mathbf{pk} = \mathbf{pk}'$ is high when the queried biometric sample belongs to the user, and is negligible otherwise.

Remark. *In the end of the enrollment phase, all related training data and variables are discarded to preserve the user's privacy. We only store the protected template **pk** and a set of hashing functions F^K belonging to the user.*

3. Experiments

3.1. Setup

The Randomness Beacon [20] project has been running by NIST (National Institute of Standards and Technology) since 2011 until now. This project broadcasts publicly a consistent, 512-bit, full-entropy random number every 60 seconds. To simulate the system's key generator, we downloaded one million unpredictable bit-strings from Beacon's official website and used those strings as ground truth labels (string **s**) when training FEHash.

The push and pull manner of Triplet loss [36] helps to increase the discrimination between the intra and inter distributions. Hence, we used the famous model, FaceNet, as the feature extractor. In practice, we applied the pretrained model of David Sandberg [1] to extract 512-dimensional embedding per 160×160 pixels crop. Multi-task CNN [49] was adopted to detect and align face images.

We adopted the Simplest Color Balance (SCB) [24] and then used gamma correction to lessen the effect of illumination for every images. To reduce the burden of computation for large scale training sets, we applied geometric transformations such as shear, rotation, zoom, scale and horizontal flip for the color normalized enrollment faces, yielding in total 13 augmented crops per input. We demonstrate the color normalization and augmentation steps in Figure 3.

[1] Source code: https://github.com/davidsandberg/facenet

Figure 3: Color normalization and augmentation.

We assigned labels for padding people in S^+ and S^-; therefore, the padding people also have their own deterministic hash codes. Intuitively, the larger p is, the better the uniformity of the hash code distribution becomes. Given a database that has N subjects, for an extreme value $p = N - 1$, the probability of the false collisions would be minimal, and the database would be perfectly shattered after iterating Algorithm 1 K-times, with $2^K \geqslant N$. In practice, we chose $p \approx 10\%$ of N to avoid closed-set settings. Of note, including unseen padding people's images in a test set would reduce the value of FAR since their hashing results were trained and completely different with the user. Therefore, only the *unseen* impostors (i.e., remaining $(N-1-p)$ subjects) were used in the verification phase to reflect the exactness of FAR.

In terms of the two adjustable parameters, we used the default noise rate NR_d for all experiments. The *grid search* method was applied to determine appropriate values of D (Section 3.5.2).

3.2. Databases

For a fair comparison with the state-of-the-art template protection methods, we used three popular databases: CMU-PIE, FEI, and FERET.

The CMU-PIE [37] database consists of 41,368 images of 68 subjects. Each subject has images under 43 different illumination conditions, 13 different poses and 4 different expressions. We used 5 poses (p05, p07, p09, p27, p29) and all illumination settings for our experiments, so that each subject has 105 images. In one-shot enrollment, we randomly selected one image per user for training and the rest were used for testing. In multi-shot enrollment, 10 images were randomly chosen for training and the remaining were used for testing, as done in [12][11][30][1][45]. Due to the lower bound requirement of p in the Equation 2, we used parameter $p = 13$ for experiments using the PIE database.

The FEI [44] database contains 2,800 color images of 200 subjects. Each subject has 14 images with pose rotation up to about 180 degrees. We used 9 poses (p03, p04, p05, p06, p07, p08, p11, p12, p13) for our experiments. In one-shot enrollment, we randomly selected one image per user for training and the rest were used for testing. In multi-shot enrollment, 4 images were randomly chosen for training and the remaining were used for testing, as done in [1]. We chose $p = 21$ for experiments using the FEI database.

The FERET [28] database contains 14,126 facial images of 1,199 individuals. We chose 238 subjects, each having 4 color pictures of their frontal face. 77 subjects have all of their pictures taken in one session; 105 in two sessions; 35 and 21 in 3 and 4 sessions, respectively. There are variations of pose, illumination, expression, hairstyle, makeup, and occlusion in the database. In one-shot enrollment, we randomly selected one image per user for training and the rest were used for testing. In multi-shot enrollment, 2 images were randomly chosen for training and the remaining were used for testing, as done in [12][1]. We chose $p = 21$ for all the experiments using the FERET database.

3.3. Evaluation metrics and Matching methods

Since the train-test datasets are generated randomly, we report the mean and standard deviation of Equal Error Rate (EER) of 5 different train-test splits as the evaluation metric. We also record the mean value of GAR at different FAR.

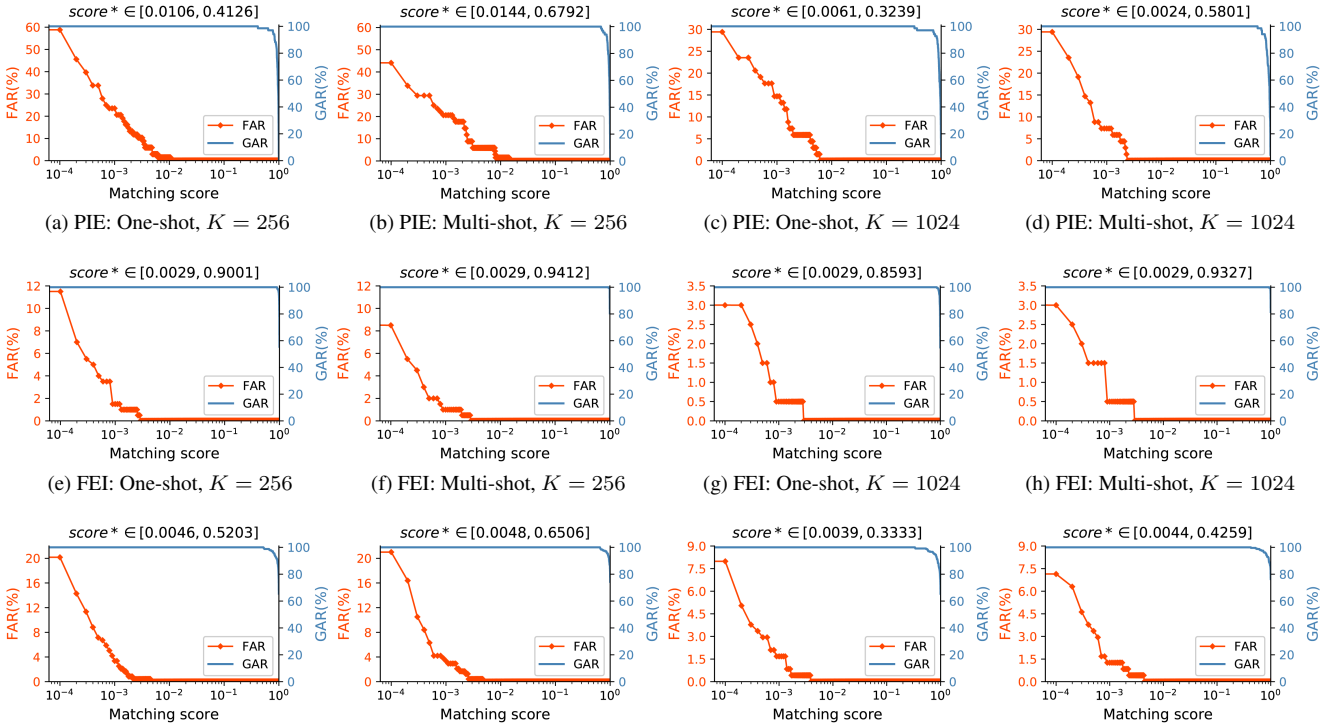
A common way, for making the decision in verification phase, is comparing the stored template **pk** to the protected template of query sample $\mathbf{pk}'$ [32]. If they are identical, the verification is successful. This approach was applied in various face template protection schemes [12][11]. However, the *fixed* matching method has been argued to have limited use in practice [30] (i.e., since the matching $score \in \{0, 1\}$, it leaves no room for implementing threshold approach).

In some recent works [30][1][45], the *tunable* matching approach was used to achieve a desired value of GAR/FAR. A set of augmented images of the query face is generated, and $\mathbf{pk}'$ is then calculated for each one, yielding a set of templates $\mathcal{T}$. As given in [30], the final matching score is defined by the number of templates $\mathbf{pk}'$ in $\mathcal{T}$ that match the stored template **pk**, yielding the matching $score \in [0, 1]$.

For a fair comparison and showing the robustness of our method, we report the system performances in both two matching ways. In tunable matching, we apply shear, zoom, rotation, scaling, and horizontal flip to generate five augmented images (with the same manner in [1]). For each augmented of size $m \times m$, we extract all possible crops of size $n \times n$; the crops are then resized back to $m \times m$. The cardinality of set $\mathcal{T}$ is $|\mathcal{T}| = 1+5\times(m-n+1)\times(m-n+1)$. In our experiments, we chose $m = 160$ and $n = 157$.

3.4. Results

The experimental results using tunable approach as matching method are illustrated in Figure 4. The average values of GAR and FAR from 5 different train-test splits are displayed in the figure. We also report the optimal matching $score^*$, in which the EER is lowest. At the strict operating point of 0% FAR, we achieved 100% GARs for every type of enrollment, values of K, and databases. In addition, this



(a) PIE: One-shot, $K = 256$ (b) PIE: Multi-shot, $K = 256$ (c) PIE: One-shot, $K = 1024$ (d) PIE: Multi-shot, $K = 1024$

(e) FEI: One-shot, $K = 256$ (f) FEI: Multi-shot, $K = 256$ (g) FEI: One-shot, $K = 1024$ (h) FEI: Multi-shot, $K = 1024$

(i) FERET: One-shot, $K = 256$ (j) FERET: Multi-shot, $K = 256$ (k) FERET: One-shot, $K = 1024$ (l) FERET: Multi-shot, $K = 1024$

Figure 4: Verification results measured by *tunable* matching from various databases (best view in color).

perfect condition (i.e., EER = 0%) was held during a long interval of value $score^*$.

Database	Enroll. Type	K	GAR@FAR	EER
PIE	One-shot	256	96.35±0.49%@0.09%	1.78±0.24%
		1024	94.98±0.05%@0.06%	2.47±0.02%
	Multi-shot	256	96.54±0.35%@0.09%	1.67±0.18%
		1024	95.77±0.22%@0.03%	2.09±0.11%
FEI	One-shot	256	98.76±0.16%@0.01%	0.61±0.08%
		1024	98.25±0.35%@0.006%	0.87±0.17%
	Multi-shot	256	99.44±0.13%@0.01%	0.27±0.06%
		1024	98.90±0.14%@0.006%	0.54±0.07%
FERET	One-shot	256	97.56±0.36%@0.01%	1.21±0.18%
		1024	96.49±0.59%@0.005%	1.74±0.29%
	Multi-shot	256	98.11±0.59%@0.01%	0.93±0.29%
		1024	97.48±0.01%@0.005%	1.25±0.04%

Table 1: Verification results from various databases (*fixed* matching method).

The verification results measuring by fixed matching method are shown in Table 1. We achieved $\geqslant 95\%$ GARs, while maintaining very low FARs on all three databases. Another observation from Figure 4 and Table 1 is that GARs and FARs tend to decrease simultaneously upon increasing K. The reason for this observation will be discussed in the next section.

We compared our results with alternative algorithms on

Method	Enroll. Type	K	GAR@FAR	EER
Hybrid Approach [12]	Multi-shot	210	90.61%@1%	6.81%
BDA [11]	Multi-shot	76	96.38%@1%	-
MEB Encoding [30]	Multi-shot	256	93.22%@0%	1.39%
		1024	90.13%@0%	1.14%
Deep CNN [1]	One-shot	256	91.91%@0.1%	4.00%
		1024	91.34%@0.1%	3.60%
	Multi-shot	256	97.35%@0%	0.15%
		1024	96.53%@0%	0.35%
DH-NND [45]	One-shot	255	96.2%@0.01%	0.99%
		1023	96.0%@0.01%	1.32%
	Multi-shot	255	99.9%@0.01%	0.051%
		1023	99.0%@0.01%	0.078%
Our Method	One-shot	256	**100%@0%**	**0%**
		1024	**100%@0%**	**0%**
	Multi-shot	256	**100%@0%**	**0%**
		1024	**100%@0%**	**0%**

Table 2: Performance comparison with other algorithms on PIE database.

PIE database in Table 2. In terms of tunable matching, our method dominated all other approaches using the same matching method [30][1][45]. In one-shot enrollment, we achieved 100%GAR@0%FAR for $K = 256$, which was 3.8% improvement in matching performance compared to 96.2%GAR@0.01%FAR in [45]. In terms of fixed matching, for security perspective, the values of K and FAR were

used to compare to the counterparts in the algorithms [12] and [11]. Our matching performance surpassed [12] which offered acceptable security level against brute force attack; and was competitive to [11] which provided weaker security strength against brute force and dictionary attacks.

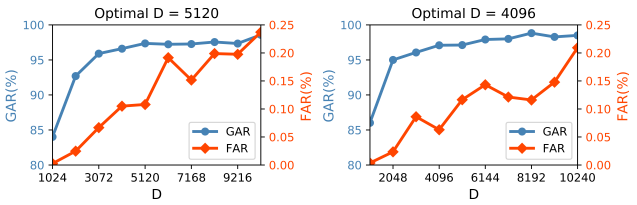
3.5. Discussion

3.5.1 Parameter K

The well-known limitation of LSH is that it requires long codes to achieve high accuracy [23][47][15]. Ji *et al*. [19] mathematically proved that the variance of LSH's estimation is large, yielding large estimation error. So, the value of K should be sizeable enough to reduce false collisions (FAR). However, a large value of K decreases the collision probability (GAR) between similar samples as well. This may explain the mentioned observation: GAR and FAR are inversely proportional to the size of K. However, our verification results are more stable compared to state-of-the-art studies [1][45]. There is no drastic change in GAR or EER for different K values. Thus, parameter K could be selected flexibly based on requirements of security level.

3.5.2 Parameter D

Since the sets of functions F^{256} and F^{1024} are constructed with the same manner, we did grid search experiments for the three databases under $K = 256$ setting to find the ideal values of D. The magnitude of D is calculated by $D = i \times 1024$, with i in range $[1..10]$. In this analysis, we used the fixed matching method to compute the values of GAR and FAR for the sake of simplicity. We observed the same pattern of all 3 databases: the higher dimension, the better GAR; however, the size of D is directly proportional to FAR. The reason may be that increasing the *complexity* would make SVM overfit the data. Figure 5 shows the grid search results of the PIE database in one-shot and multi-shot enrollments.



(a) PIE: One-shot, $K = 256$ (b) PIE: Multi-shot, $K = 256$

Figure 5: Grid search results with respect to parameter D.

For the efficiency concerns (i.e., computation cost of SVM, storage space), the small-size D is favor. Besides, we prefer verification performance achieving high GAR at an *acceptable* FAR value. Table 3 briefly summarizes the optimal values of D with respect to our *priorities* (i.e., high GAR at FAR $\leqslant 0.1\%$, small-size D). Those selections were applied for all experiments in Section 3.4.

Database	PIE		FEI		FERET	
Enroll. Type	One	Multi	One	Multi	One	Multi
D	5120	4096	4096	4096	5120	5120

Table 3: Optimal value of D for different databases.

3.5.3 Parameter NR

We carried out a simple experiment on the FEI database to explore the influence of NR parameter. Table 4 shows GAR and its corresponding FAR when varying NR value. Regarding the tradeoff between user-friendliness and security level, we could say that NR affects the performance of the system, which strengthens our expectation of hiding user's data by padding people. In this analysis, we used a following condition: $p = 21$, $K = 512$, $D = 3072$, one-shot enrollment type, and fixed matching method.

NR	75%	87.5%	93.75%	96.875%	98.4375%
GAR	98.94%	98.31%	98.25%	98.13%	97.25%
FAR	0.008%	0.008%	0.005%	0.005%	0.004%

Table 4: Verification results from different NR values.

3.5.4 Post-quantum application

User's transformed templates $F^K(\mathbf{x})$ have the properties of randomness and changeability. Besides, the cryptographic hash function (in here: SHA3-512) could be any function that follows the random oracle model. Hence, the template **pk** can played as a secret factor in several quantum-secure algorithms (i.e., [13][33][26]), depending on its representation form. In this study, the protected template **pk** is a perfect fit for the binary private key of the "Dual" Regev asymmetric encryption [13]. The success decryption rate is definitely equal to GAR of our novel scheme that achieved high matching rate.

3.6. Requirements of biometric cryptosystem

Unlinkability: We ensure the requirement of unlinkability by assigning the unpredictable hashing output for user. Since each predefined binary string (string **s**) is uncorrelated totally with biometric data and is independent with others, the transformed template inherits those properties also.

Irreversibility: FEHash reaches `REQ-WBP` by taking advantage of LSH and SVM, and replying on the strength of cryptographic hash function.

1. Since SHA3-512 is a one-way function, it is hard to obtain the transformed template from the compromised protected template.

2. From LSH viewpoint, it is clear that the adversaries cannot get any fruitful information due to a sign function (we omit the easy proof).

3. In SVM, hyperplane is independent of other training samples except support vectors:

$$\langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b = 1 \vee \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b = -1. \quad (8)$$

Theorem 1. *Fix* $\Omega \in \mathbb{R}^{D \times d}, \boldsymbol{r} \in \mathbb{R}^D, \boldsymbol{w} \in \mathbb{R}^D, b \in \mathbb{R}$. *We have* $\phi(\boldsymbol{x}) = (x'_1, x'_2, \ldots, x'_D)$, $\phi(\boldsymbol{x}) \in [-\sqrt{\frac{2}{D}}, \sqrt{\frac{2}{D}}]^D$, *where* $\boldsymbol{x} \in [-1, 1]^d$[2]. *The equation:* $\langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle + b = y$, *with* $y \in \{-1, 1\}$ *has infinitely many solutions.*

Proof. $\langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b = y$

$\Leftrightarrow w_1 x'_1 + w_2 x'_2 + \cdots + w_D x'_D = y - b$, which is equivalent to an underdetermined system of one equation in D unknowns, and therefore, no unique solution can be reached. □

In the worst case (i.e., user's projected data are support vectors), Theorem 1 proved that adversaries learn nothing useful about value of projected data $\phi(\mathbf{x})$ from a single hashing function. However, some elements of $\phi(\mathbf{x})$ could be computed if there are several linearly independent equations that have a same number of unknown elements. However, even though given a set of hashing functions, the system still achieves `REQ-WBP` security level. Since the projection function is pairwise independent, attackers cannot collect more than one equation from the same $\phi_{\Omega,\mathbf{r}}(\cdot)$. Therefore, FEHash is computationally infeasible to reverse to $\mathbf{x}$ from the stored information $F^K_{\mathbf{w},b,\Omega,\mathbf{r}}(\cdot)$.

4. Security analysis

We assume that the attackers could access the system's feature extractor (i.e., FaceNet). The helper data of user were compromised (i.e., projection parameters $\{\Omega, \mathbf{r}\}^K$ and classifier hyperplanes $\{\mathbf{w}, b\}^K$).

4.1. Auxiliary data based attack

The adversary's concern is finding a vector $\mathbf{x}_a = \mathbf{x}_g$, where $\mathbf{x}_g \in X_{genuine}$. The attackers can reverse the stolen projection function to get $\mathbf{x}_g$ if they have the value of $\phi(\mathbf{x}_g)$. Thanks to the *primal form* of SVM, we did not store any value of $\phi(\mathbf{x}_i)$, where $\mathbf{x}_i \in X_{train}$, and $i = 1, \ldots, |X_{train}|$. Theorem 1 proved that adversaries cannot exploit the relation of the hyperplane to its support vectors to calculate the training data values. Besides, the user's projected data have only $(100 - NR)\%$ chance of being the tips of the support vectors. Thus, FEHash is secure against auxiliary data based attacks.

[2]Triplet loss has a constraint on feature embedding (i.e., L_2 norm).

4.2. Similarity based attack

FEHash has the similarity preserving characteristic of LSH, so it could be broken by similarity-based attacks. The attackers try to find a vector $\mathbf{x}_a \approx \mathbf{x}_g$ satisfying $SHA(F^K(\mathbf{x}_a)) = SHA(F^K(\mathbf{x}_g))$. Generally, they use a huge set of images to exploit the FAR of the system (i.e., dictionary attack, collision attack). More sophisticatedly, they can apply Genetic Algorithm based Similarity-based Attack Framework (GASAF) [9] to get the *approximate* vector $\mathbf{x}_a$. However, GASAF requires $F^K(\mathbf{x}_g)$ (or string $\mathbf{s}$) that is used internally and discarded, which makes the attack framework impractical.

4.3. Exhaustive search attack

Since $F^K(\mathbf{x}_g)$ is generated based on the unpredictable string $\mathbf{s}$, there is no specific key space of $F^K(\mathbf{x}_g)$. Brute force attack is the only method that could be implemented to guess $F^K(\mathbf{x}_g)$. Thus, the security strength against brute force attack on $F^K(\mathbf{x}_g)$ is K bits (using classical computer) or $\sqrt{K} \approx \frac{K}{2}$ bits (using quantum computer [14]).

5. Scope and Future work

The system randomly and internally chose a set of padding people. However, if the attackers have information of all p people (i.e., faces, embeddings), they can determine the value of every bit $F_i(\mathbf{x}_g)$ based on the occurrence of 1 and 0 in $F_i(X_{padding})$, with $F_i \in F^K$, and $i = 1, \ldots, K$. It happens because p is odd number. We easily bypass this problem by using even number p, with the constraint $|S^+| = |S^-| = \frac{p}{2}$.

Tailoring a complete balanced training set from even number padding people and deploying a post-quantum biocryptosystem are left for future work.

6. Conclusion

We proposed a hashing function that produces the predefined output for biometric samples belonging to the user. We achieved 100% GARs at the strict operating point of zero FAR when measuring by tunable matching method, and achieved high GARs (95∼99%) when evaluating by fixed matching approach. The provable security level and outperforming results on several popular face benchmarks demonstrate the superiority and potentiality of our method.

Acknowledgement

We thank Thang Hoang and Van Thong Huynh for helpful advice. This research was supported by NRF-2017R1D1A1B03035343 and funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant number NCM2019-18-01.

References

[1] S. Chalamala A. K. Jindal and S. K. Jami. Face template protection using deep convolutional neural network. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 575–5758, 2018. 1, 2, 5, 6, 7

[2] M. Ao and S. Li. Near infrared face based biometric key binding. 2009. 2

[3] D. Apon, C. Cho, K. Eldefrawy, and J. Katz. Efficient, reusable fuzzy extractors from lwe. In *Cyber Security Cryptography and Machine Learning*. Springer International Publishing, 2017. 2

[4] L. Ballard, S. Kamara, and M. Reiter. The practical subtleties of biometric key generation. In *Proceedings of the 17th USENIX Conference on Security Symposium*, SS'08, pages 61–74, 2008. 1, 2

[5] E. Chang, R. Shen, and F. W. Teo. Finding the original point set hidden among chaff. In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, ASIACCS '06. Association for Computing Machinery, 2006. 2

[6] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, STOC '02, pages 380–388. ACM, 2002. 1

[7] B. Chen and V. Chandran. Biometric based cryptographic key generation from faces. In *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA)*, 2007. 2

[8] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000. 3

[9] X. Dong, Z. Jin, and A. B. J Teoh. A genetic algorithm enabled similarity-based attack on cancellable biometrics. *IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2019. 8

[10] J. Fagertun and M. B. Stegmann. The IMM frontal face database, 2005. 3

[11] Y. C. Feng and P. C. Yuen. Binary discriminant analysis for generating binary face template. *IEEE Transactions on Information Forensics and Security*, 7(2):613–624, 2012. 5, 6, 7

[12] Y. C. Feng, P. C. Yuen, and A. K. Jain. A hybrid approach for generating secure and discriminating face template. *IEEE Transactions on Information Forensics and Security*, 5(1):103–117, 2010. 2, 5, 6, 7

[13] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, STOC '08, page 197–206, 2008. 7

[14] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, STOC '96, page 212–219. Association for Computing Machinery, 1996. 8

[15] J. He, S. Chang, R. Radhakrishnan, and C. Bauer. Compact hashing with joint optimization of search accuracy and time. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 753–760, 2011. 7

[16] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. pages 604–613, 1998. 1

[17] Information technology — security techniques — biometric information protection. Standard, International Organization for Standardization, 2011. 2

[18] A. K. Jain, K. Nandakumar, and A. Nagar. Biometric template security. *EURASIP J. Adv. Signal Process*, 2008. 1, 2

[19] J. Ji, J. Li, S. Yan, B. Zhang, and Q. Tian. Super-bit locality-sensitive hashing. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'12, page 108–116. Curran Associates Inc., 2012. 7

[20] J. Kelsey, L. T. A. N. Brandao, R. Peralta, and H. Booth. A reference for randomness beacons, 2019. 4

[21] A. Kholmatov and B. Yanikoglu. Realization of correlation attack against the fuzzy vault scheme. 2008. 2

[22] Y. Kim and K. Toh. A method to enhance face biometric security. In *2007 First IEEE International Conference on Biometrics: Theory, Applications, and Systems*, 2007. 2

[23] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, NIPS'09, page 1042–1050. Curran Associates Inc., 2009. 7

[24] N. Limare, J. L. Lisani, J. M. Morel, A. B. Petro, and C. Sbert. Simplest Color Balance. *Image Processing On Line*, 1:297–315, 2011. 4

[25] H. Lu, K. Martin, F. Bui, K. N. Plataniotis, and D. Hatzinakos. Face recognition with biometric encryption for privacy-enhancing self-exclusion. In *2009 16th International Conference on Digital Signal Processing*, 2009. 2

[26] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology (EUROCRYPT)*, pages 1–23, 2010. 7

[27] L. Ma, T. Tan, Y. Wang, and D. Zhang. Efficient iris recognition by characterizing key local variations. *IEEE Transactions on Image Processing*, 13(6):739–750, 2004. 1

[28] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000. 2, 5

[29] R. Plamondon and S. N. Srihari. Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000. 1

[30] B. U. Kota R. K. Pandey, Y. Zhou and V. Govindaraju. Deep secure encoding for face template protection. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 77–83, 2016. 1, 2, 5, 6

[31] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, pages 1177–1184. Curran Associates, Inc., 2007. 2, 3

[32] C. Rathgeb and A. Uhl. A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, 2011. 1, 5

[33] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, STOC '05, page 84–93, 2005. 7

[34] M. Savvides, B. V. K. Vijaya Kumar, and P. K. Khosla. Cancelable biometric filters for face recognition. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, volume 3, 2004. 2

[35] W. J. Scheirer and T. E. Boult. Cracking fuzzy vaults and biometric encryption. In *2007 Biometrics Symposium*, 2007. 2

[36] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. 2, 4

[37] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression (pie) database. In *Proceedings of 5th IEEE International Conference on Automatic Face Gesture Recognition*, pages 53–58, 2002. 2, 5

[38] K. Simoens, P. Tuyls, and B. Preneel. Privacy weaknesses in biometric sketches. In *2009 30th IEEE Symposium on Security and Privacy*, 2009. 2

[39] Yagiz Sutcu, Qiming Li, and Nasir Memon. Design and analysis of fuzzy extractors for faces. *Proceedings of SPIE - The International Society for Optical Engineering*, 2009. 2

[40] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[41] A. B. J. Teoh, A. Goh, and D. C. L. Ngo. Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12), 2006. 2

[42] A. B. J. Teoh, D. C. L. Ngo, and A. Goh. Personalised cryptographic key generation based on facehashing. *Computers and Security*, 23(7), 2004. 2

[43] A. B. J. Teoh and C. T. Yuang. Cancelable biometrics realization with multispace random projections. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(5), 2007. 2

[44] C. E. Thomaz and G. A. Giraldi. A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing*, 28(6):902–913, 2010. 2, 5

[45] N. Nasrabadi V. Talreja, M. Valenti. Zero-shot deep hashing and neural network based error correction for face template protection. *IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2019. 1, 2, 5, 6, 7

[46] M. Veen, T. Kevenaar, G. Schrijen, A. H. M. Akkermans, F. Zuo, and P. Holstlaan. Face biometrics with renewable templates. *Proceedings of SPIE - The International Society for Optical Engineering*, 2006. 2

[47] J. Wang, S. Kumar, and S. Chang. Semi-supervised hashing for large-scale search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2393–2406, 2012. 7

[48] Y. Wu and B. Qiu. Transforming a pattern identifier into biometric key generators. In *2010 IEEE International Conference on Multimedia and Expo*, 2010. 2

[49] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 4

[50] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399–458, 2003. 1