2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Towards Protecting Biometric Templates Without Sacrificing Performance

Jing Li
School of Computing
National University of Singapore
lijing@comp.nus.edu.sg

Yongkang Wong
Interactive and Digital Media Institute
National University of Singapore
yongkang.wong@nus.edu.sg

Terence Sim
School of Computing
National University of Singapore
tsim@comp.nus.edu.sg

***Abstract*—The ideal biometric template protection scheme possesses the properties of irreversibility, revocability, unlinkability, and good performance. These properties protect the security of the biometrics system as well as users' privacy. Practical systems, however, fall short of this ideal. In this paper, we present a novel protection scheme that achieves this ideal under the circumstance that a subject's token and his biometric template are not concurrently exposed. Moreover, our scheme can add template protection to *any* face verifier. We do this by rendering virtual faces, rather than by devising new biometric features, which is the more common approach. Experimental evaluations using two public face recognition systems show that accuracy is not adversely affected with our scheme.**

I. Introduction

Like Personal Identification Numbers (PINs, by which we also include passwords), biometrics has become a popular method for identity authentication in order to control access to a protected system or resource. Unlike PINs, however, biometrics generally cannot be revoked, in the sense that one cannot change one's fingerprint when the fingerprint is stolen, the way a PIN can. What can be revoked, however, is the *biometric template* of the fingerprint that is stored in a biometric authentication system. If the template is stolen, it should be possible to invalidate the old template and replace it with a new one so that security is not compromised: an attacker cannot use the stolen template to gain access to the protected resource, while the user should not be denied access with the new template.

This research area, called Biometric Template Protection, is seeing increased activity in recent years [1]. Note that the biometric template typically does not store the actual biometric sample (*e.g.* fingerprint image) of the user, but rather the features extracted from the sample. These features are subsequently used for matching against a database of features from known individuals. Two recent papers [1,2] provide good summaries of the key issues and open research questions. As these authors have pointed out, the theoretical ideal template protection scheme possesses these properties:

1) **Irreversibility**: the stolen template should not reveal the identity, nor the biometric sample, of the user. This protects the identity of the user.
2) **Revocability**: it should be relatively easy to invalidate a stolen template and replace it with a new one. This prevents the stolen template from being used to gain access to the protected resource, while allowing the legitimate user continued access.

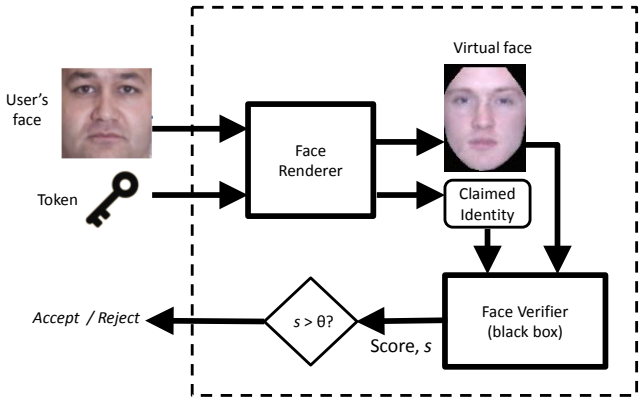


Fig. 1: Our template protection scheme adds a Face Renderer to *any* black box Face Verifier to render a virtual face, guided by the user's token. This virtual face is easily revoked by changing the token should the Verifier's template become compromised.

3) **Unlinkability**: it should be computationally difficult to determine whether two or more templates were derived from the same user. This protects the privacy of the user, who may be using the same biometric across different applications.
4) **Good performance**: The accuracy of the authentication system should not be decreased by the protection of biometric templates.

Practically, achieving *all* the above properties simultaneously is difficult. Most existing systems that protect biometric templates achieve irreversibility and revocability at the expense of good performance. Moreover, these systems typically devise new and sophisticated templates (and hence features) to achieve the said goals. In other words, it is not possible with these approaches to add template protection to an authentication system without changing its features. And since authentication accuracy is determined by the discriminability of the features, these approaches invariably affect accuracy. This trade-off between irreversibility, revocability, and good performance thus appears insurmountable.

Note that one naive way to protect templates is to encrypt them. But this shifts the difficulties from biometrics to crypto-key management, which may not be any easier to solve. In this paper, we present a novel way to add template protection to *any* authentication system by synthesizing virtual biometric samples with the assistance of individual tokens. Using virtual biometric enables our method to attempt to

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

decouple performance from the other three properties. Our method achieves irreversibility, revocability, unlinkability and good performance, when a token and its corresponding virtual biometric sample are never exposed at the same time. Since most authentication system is performed offline, the pairwise exposure of token and virtual biometric sample can be easily prevented by timely removing any virtual data.

More concretely, we demonstrate our protection scheme by adding a Face Renderer to a Face Verifier, which we treat as a black box, and by adjusting the threshold for deciding between Accept and Reject (Fig. 1). The Face Verifier never sees the user's real face, only the virtual face, which is rendered according to some parameters stored in the user's token. Should the Verifier's template be compromised, it is easy to revoke the virtual face by changing the token. Also, by carefully controlling how virtual faces are rendered, we can achieve all the desired properties. As for the Face Verifier, we only require it to output a similarity score, $s \in [0, \tau]$, where τ, the maximum possible score, means that the input face is "perfectly similar", and 0 means "completely dissimilar", to the claimed identity. For many practical systems, this requirement is easily satisfied.

Our template protection scheme makes two contributions: (a) it possesses the properties of irreversibility, revocability, unlinkability and good verification performance, in the case of non-pairwise exposure of token and virtual biometric data; (b) it may be added on to *any* Face Verifier, because it treats the Verifier as a black box, requiring only that the Verifier outputs a score between 0 and some maximum value τ.

II. Related Work

According to the ISO/IEC Standard 24745 on biometric information protection, a protected biometric template is typically divided into *Pseudonymous Identifier* ($\mathbf{PI}$) and *Auxiliary Data* ($\mathbf{AD}$) [1]. Based on these components, existing template protection schemes can be categorized into feature transformation approaches [3]–[5] and biometric cryptosystems [6]–[12].

For the feature transformation approaches, a given biometric template, $\boldsymbol{x}$, is first transformed using a non-invertible function with known transformation parameters (*i.e.* $\mathbf{AD}$), and stored in the database as $\mathbf{PI}$. During the authentication stage, the query sample, $\boldsymbol{x}'$, is transformed with the corresponding $\mathbf{AD}$ to construct $\mathbf{PI}'$, which is then compared with the claimed $\mathbf{PI}$. Example of such schemes include cancellable biometric [3,4] and Biohashing [5]. In [3], Bolle *et al.* introduced the concept of cancellable biometrics, and demonstrated it with signal domain distortion and feature domain distortion. Savvides *et al.* [4] proposed a cancellable biometrics scheme which encrypts facial images with correlation filters. They show that convolution with any random kernel does not change the correlation output peak-to-sidelobe ratios. Although this approach preserves authentication performance [4], it can be jeopardized if the convolution kernel is leaked. In BioHashing [5], a template is protected via iterative inner products with token-derived random sequences and multispace quantization.

In biometric cryptosystems, $\boldsymbol{x}$ is first registered to an authentication system with a *cryptographic key* (*i.e.* $\mathbf{PI}$) to generate a user-specific *helper data* (*i.e.* $\mathbf{AD}$). Given a query sample $\boldsymbol{x}'$, $\mathbf{AD}$ is used to reconstruct a new cryptographic key, $\mathbf{PI}'$, that is identical to $\mathbf{PI}$ if $\boldsymbol{x}'$ is sufficiently similar to $\boldsymbol{x}$. Example of such schemes include the Fuzzy commitment scheme [6], Helper Data System (HDS) [7,8], Fuzzy vaults [9,11,13],and Secure sketches [12]. Juels and Wattenberg [6] introduced the Fuzzy commitment schemes, where an error-correction code is adapted to protect against fuzzy variability or random noise present in $\boldsymbol{x}$. The matching of $\mathbf{PI}$ and $\mathbf{PI}'$ was performed in the hashing domain for enhanced template protection. In line with [6], Michiel *et al.* [7] proposed HDS for face biometric, where six facial feature objects were selected to construct a biometric template. Lu *et al.* [8] extended HDS for a self-exclusion scenario. In the Fuzzy vault scheme [9], the author introduced *order invariance* features where key extraction can be achieved with unordered data sets. Different from Fuzzy commitment scheme and HDS, the Secure sketch method [12,14] produces a sketch (*i.e.* $\mathbf{AD}$) of a given $\boldsymbol{x}$ *without* an assigned key. The sketch allows reliable reconstruction of the genuine biometric template if the query belongs to the same individual. The key drawback for these approaches is that the protected biometric template is restricted to a particular authentication algorithm with a specified feature.

The method of Lee *et al.* [15] is closest to ours in that it also uses virtual faces for recognition, achieved by altering PCA and ICA coefficients. However, their virtual faces exhibit low pixel resolution and obvious visual artifacts. Furthermore, there is no mechanism to ensure all users are associated with unique virtual faces. In contrast, our method protects facial biometrics by generating a virtual distinguishable identity for each individual, and which guarantees irreversibility, revocability, and unlinkability for *any* Face Verifier.

III. Method

Our protection scheme takes the approach of "feature transformation". More precisely, from the original input image of a user's face, $\boldsymbol{x}$, we render (synthesize) a virtual face image as $\mathbf{PI}$, which is then input into the Face Verifier for actual classification (Fig. 1). In effect, our Renderer creates a virtual identity for each user, thereby protecting the user's real identity. This pairing of real and virtual identities is maintained as $\mathbf{AD}$ in a user-specific token (until the next revocation), so that every authentication attempt by one user always results in the same virtual identity being rendered. When the virtual identity needs to be revoked, it is first marked as invalid, and then a new virtual identity is created, far away (in the discriminative sense) from all other virtual identities in the system. This guarantees that virtual identities never collide, and verification accuracy remains high.

To achieve the desired rendering, we make use of an orthogonal subspace decomposition technique called MMDA [16], assisted by parameters stored in a user-specific token (which maintains the said real-virtual pairing), see Fig. 2. The advantage of using MMDA is its ability to separate identity and non-identity variations, such as illumination, into orthogonal subspaces. In turn, this permits independent manipulation of subspace coefficients, from which a new face may be rendered by doing MMDA back-projection.

A. Multimodal Discriminant Analysis (MMDA)

MMDA is a method that decomposes a dataset of face images containing multiple appearance variations, also called

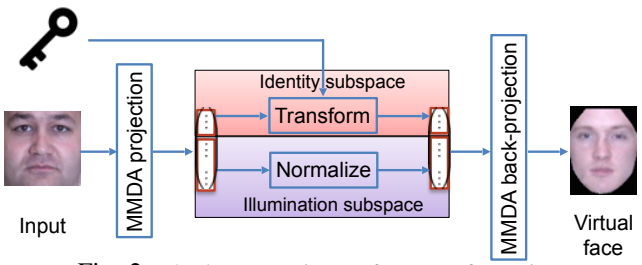


Fig. 2: The key steps in our face transformation.

modes, (such as identity, illumination) into orthogonal subspaces, and comprises these steps:

1) The dataset is firstly centered by their mean vector: $\boldsymbol{X} = [\boldsymbol{x_0} - \boldsymbol{\mu},\ \boldsymbol{x_1} - \boldsymbol{\mu},\ \ldots, \boldsymbol{x_n} - \boldsymbol{\mu}]$, $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=0}^{n} \boldsymbol{x_i}$;
2) The data is whitened so that its scatter matrix $\boldsymbol{S}$ equals the identity matrix $\boldsymbol{I}$: $\hat{\boldsymbol{X}}\hat{\boldsymbol{X}}^{\boldsymbol{T}} = \boldsymbol{I}$. The whitening procedure is: $\hat{\boldsymbol{X}} = \boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{X}$, where $\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\boldsymbol{T}}$ and $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{D}^{-\frac{1}{2}}$;
3) Linear Discriminative Analysis (LDA) is then applied onto each mode;
4) Then a matrix is formed as $\boldsymbol{V} = [\boldsymbol{V_1}, \boldsymbol{V_2}, \ldots \boldsymbol{V_n}]$, where $\boldsymbol{V_i}$ are the eigenvectors (with unity eigenvalues) for mode i, by LDA;
5) To decompose a vector, we use:

$$\boldsymbol{y} = \boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{x} \tag{1}$$

The vector $\boldsymbol{y}$ contains the coefficients of each mode.
6) To render (back-project), we use:

$$\boldsymbol{x} = \boldsymbol{P_r}\boldsymbol{V}\boldsymbol{y} \tag{2}$$

where $\boldsymbol{P_r} = \boldsymbol{U}\boldsymbol{D}^{1/2}$ undoes the whitening of $\boldsymbol{P}$.

MMDA has the beneficial property that bases of different modes are orthogonal to each other, that is, $\boldsymbol{V_i}^{\boldsymbol{T}}\boldsymbol{V_j} = \boldsymbol{0}$ (see [16] for more details and a proof). This property enables us to edit features in one mode without affecting other modes. In our proposed scheme, which works on frontal face images exhibiting neutral facial expression, we use only two modes: identity and illumination. Using Eqn. (1), we decompose any face image $\boldsymbol{x}$ into its identity and illumination subspaces. The coefficient vector $\boldsymbol{y}$ looks like: $\boldsymbol{y}^{\boldsymbol{T}} = [\ \boldsymbol{y}_{\text{id}}\ ,\ \boldsymbol{y}_{\text{illum}}\ ,\ \boldsymbol{r}\ ]$, which are vectors controlling the identity, illumination, and residual coefficients, respectively (see [16] for more details). In turn, these coefficients may be altered to create virtual identities or illuminations. The altered $\boldsymbol{y}$ is then synthesized into a face by the back-projecting Eqn. (2).

B. Virtual identity for authentication

1) Virtual identities: To better understand how virtual identities may be synthesized, we analyze $\boldsymbol{y}_{id}$ for a subset of 1928 face images from the Multi-PIE dataset. We discover that changing the magnitude of the identity vector does not change its perceived identity, but rotating the vector does, as exemplified in Fig. 3.

Inspired by the analysis, we generate a series of orthogonal cluster centers for different users as shown in Fig. 4. Each cluster center corresponds to one virtual identity. In the identity subspace where 2-norm is equal to α (0.4 in our experiments), cluster centers are generated one at a time by selecting a random vector from the left nullspace of existing cluster

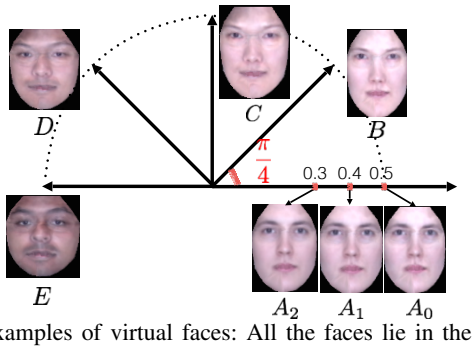


Fig. 3: Examples of virtual faces: All the faces lie in the same 2D plane in d-dimensional identity space. Faces A_0, A_1, A_2 differ only in their 2-norm (*i.e.* 0.5,0.4,0.3, respectively). Faces A_0, B, C, D lie on a hypersphere of radius 0.5, separated by an angle of $\frac{\pi}{4}$.

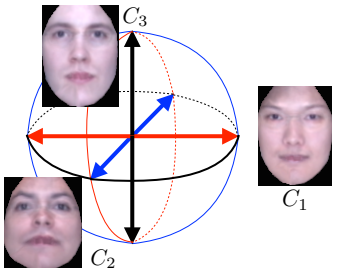


Fig. 4: Example of cluster centers in identity space: C_1, C_2, C_3 are three cluster centers that are orthogonal to each other. Their 2-norms are equal to 0.4.

centers. This guarantees that cluster centers (and hence virtual identities) are 90° apart, making them maximally discriminable, which is the ideal case.

In a d-dimensional identity subspace, there are at most d orthogonal directions, and hence at most d cluster centers that are maximally discriminable. Since each cluster center corresponds to a virtual identity, our scheme can only create d virtual identities in the ideal case. When this limit is reached, we may still continue to generate new cluster centers as follows: (i) Randomly select two existing orthogonal cluster centers $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$; (ii) Generate a new cluster center: $\boldsymbol{c} = -\frac{1}{\sqrt{2}}(\boldsymbol{c}_i + \boldsymbol{c}_j)$. This increases the number of possible cluster centers from d to $\frac{d(d+1)}{2}$. But now the angle between any two clusters may only be 60°, possibly reducing discriminability.

2) Real-virtual pairing: We had said that every real user is paired with a virtual identity. This is achieved by means of a user-specific rotation matrix $\boldsymbol{R}$, which is stored in the user's token, when enrolling or revoking that user. To enroll or revoke a user, do:

1) Decompose the user's gallery images using Eqn. (1).
2) Normalize all identity vectors, $\|\boldsymbol{y}_{\text{id}}\| = \alpha$.
3) Compute the mean identity vector, $\bar{\boldsymbol{y}}$, over all normalized identity vectors.
4) Determine a new cluster center $\boldsymbol{c}$ by the method described in the Section III-B1. Assign this cluster center to the user by marking it as "Used".
5) Compute the rotation matrix $\boldsymbol{R}$ such that $\boldsymbol{c} = \boldsymbol{R}\bar{\boldsymbol{y}}$.
6) Store $\boldsymbol{R}$ in the user's token.

Our Face Renderer remembers all the assigned cluster centers. Note that cluster centers are *not* stored in the user's token, but only in the Face Renderer.

C. Discussion

In the following, we visit the properties of the theoretical idea template protection scheme.

1) Irreversibility: In our proposed scheme, it is clear that the token information, *i.e*. the rotation matrix, does not reveal the original identity; it is merely a $d \times d$ matrix, and not a face image. And as will be shown in Section IV, the virtual face does not resemble the original face from which it was derived. It is also clear that knowing one cluster center reveals only the virtual face, and not the original face. Furthermore, knowledge of one cluster center $\boldsymbol{c}$ does not reveal the other cluster centers, since there are many directions orthogonal to $\boldsymbol{c}$. Thus, our scheme is irreversible.

2) Revocability: Revocation in our proposed scheme is also straightforward. When a user wishes to revoke his virtual identity, we first mark as invalid his current cluster center. Then we generate a new cluster by the method described in Section III-B1, compute the new rotation matrix, and update the user's token. It is clear that revocation time is dominated by the generation of a new cluster center, *i.e*. calculating the left nullspace of a $d \times d$ matrix. This takes $\mathcal{O}(d^3)$ time.

Suppose N_{user} users have been enrolled. Then our Face Renderer can only handle at most $\frac{d(d+1)}{2} - N_{\text{user}}$ revocations. This constraint is reasonable, because it is expected that verification accuracy will decrease with increasing enrollment. One simple way to postpone hitting the revocation limit is to recycle invalid cluster centers, simply by marking them as valid. This is clearly feasible, but at the cost of some overhead to keep track of the status of cluster centers.

3) Unlinkability: Unlinkability means that the virtual identities for an individual across different applications should be different. This property can be easily achieved by using different cluster centers in different applications. Since the pairing of a cluster center to each user depends on the order of enrollment, and on how a vector from the left nullspace is chosen, the probability of having the same cluster center for the same user is very low. Moreover, if we train a different MMDA model with an identity space of dimension $d' \neq d$, then cluster centers across different applications are guaranteed to be different.

4) Performance: Our protection scheme strives to maintain verification accuracy by keeping cluster centers as far apart as possible from one another. This is done by keeping the angle between two cluster centers to 90° ideally, or at least 60° if more virtual identities are required. The experiments in the next section corroborate this point.

IV. Experiments

We evaluate our template protection scheme on two publicly available face recognition systems, OpenBR [17] and OpenFace [18], using images from the Multi-PIE dataset [19]. We select 249 individuals, each having several frontal face images under different illuminations. Out of these, 161 individuals with 6 different illuminations are used to train an MMDA model with identity and illumination modes. For the remaining 88 individuals, 5 images are used as galleries and 6 are used as probes. In order to increase the number of genuine probes, we augment 45 more probe images for each individual by taking a weighted sum of the MMDA feature vectors from any two existing probes, $\boldsymbol{v'} = \beta \boldsymbol{v_1} + (1-\beta)\boldsymbol{v_2}$ ($\beta \in \{0.25, 0.5, 0.75\}$), and then by back-projecting these new features into images.

TABLE I: Equal Error Rates of OpenBR and OpenFace in Normal and Stolen-token scenarios.

		OpenBR	OpenFace
Normal Scenario	Original Faces	0.024	0.020
	Virtual Faces	0.016	0.028
	Virt. *vs*. Orig. Faces	0.035	0.044
Normal Scenario (color background)	Original Faces		0.021
	Virtual Faces		0.023
	Virt. *vs*. Orig. Faces		0.047
Stolen-token Scenario		0.018	0.031

OpenBR [17] is an open source framework that supports face recognition, age estimation and gender estimation. It implements the 4SF [20] algorithm, and computes a similarity score when given two input images. In our experiments, the score, s, of a probe against an enrolled user is the average of the scores when the probe is compared with all 5 galleries of the user. This limits the score to the range $0 \leq s \leq 24$. OpenFace [18] is an implementation of face recognition using deep neural networks. In our experiments, we use its pre-trained model to extract a 128-dimensional feature vector for each face image. During enrollment, the template for each user is formed by averaging the feature vectors of its 5 galleries. Since the feature vectors by OpenFace are normalized, we evaluate the similarity of a probe $\boldsymbol{v}$ and an averaged template $\boldsymbol{t}$ by: $s(\boldsymbol{v}, \boldsymbol{t}) = 2 - \|\boldsymbol{v} - \boldsymbol{t}\|$. This means $0 \leq s \leq 2$.

We emphasize again that our protection scheme treats OpenBR and OpenFace as black boxes, *i.e*. we do not require knowledge of the Verifiers' internal descriptors or algorithms, nor do we modify any code therein. But *if* we do know the Verifiers' inner workings, we may exploit such knowledge to further reduce their Equal Error Rate (EER).

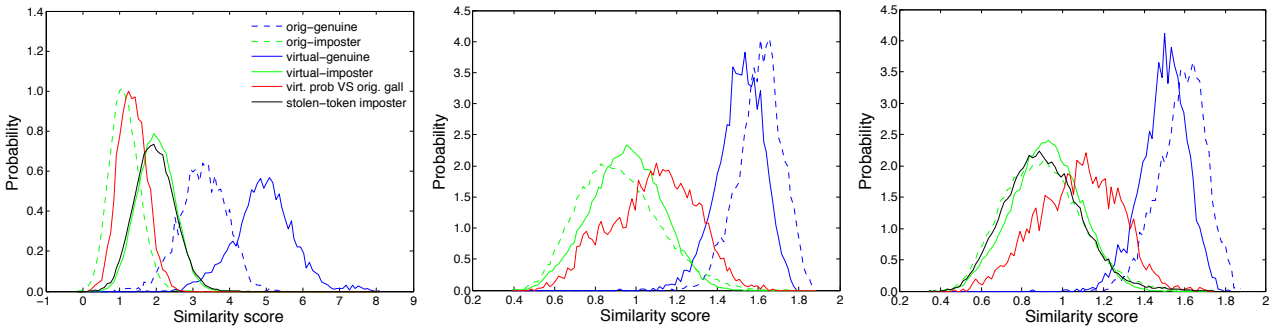
A. Evaluations

We evaluate our protection scheme under three scenarios:

1) Normal Scenario: In this scenario, each user is allocated a personal token during enrollment, in which is stored the user-specific rotation matrix $\boldsymbol{R}$. This matrix is used to transform both gallery and probe images of the user (see Section III).

During verification, each original image is used both as genuine and imposter probes, while each augmented image serves only as genuine probes. This results in 4488 genuine and 45936 imposter comparisons for our experiments, from which we determine the pdfs of the similarity scores for both genuine and imposter probes. In addition, we also calculate the EER, as follows: (1) A series of thresholds θ are set to determine corresponding False Accept Rates (FAR) and False Reject Rates (FRR). The FAR and FRR are obtained by: $\text{FAR}(\theta) = \frac{N_{\text{imp}}(s \geq \theta)}{N_{\text{imp}}}$ and $\text{FRR}(\theta) = \frac{N_{\text{gen}}(s < \theta)}{N_{\text{gen}}}$, where N_{imp} and N_{gen} are the number of imposter comparisons and genuine comparisons respectively; (2) EER is the value of FAR when it equals to FRR for the same θ.

As a baseline, we seek to know the performance of both Face Verifiers on original (*i.e*. not virtual) faces. The baseline EERs are 0.024 and 0.020 for OpenBR and OpenFace (Table I), respectively, and the corresponding score pdfs are



(a) OpenBR (b) OpenFace (c) OpenFace - color background

Fig. 5: The pdfs of similarity scores for (a) OpenBR, and (b) OpenFace, in Normal and Stolen-token scenarios. The blue dotted lines show the pdfs for the scores on genuine probes using original face images, while the green dotted lines show the score pdfs for original imposter probes. The blue and green solid lines show the corresponding pdfs using virtual faces as probes. The red lines show the comparison between virtual faces and their corresponding original faces. The black lines show the scores for imposter probes in the Stolen-token scenarios.

Fig. 6: **Top row:** Original faces of Fig. 8 with colored background; **Bottom row:** Corresponding virtual face with colored background.

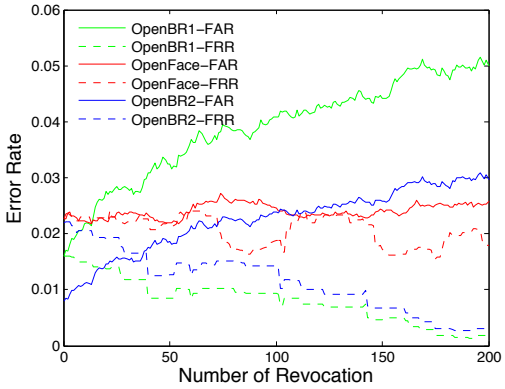


Fig. 7: Error rates *vs.* revocations

plotted in dotted lines (blue for genuine probes, green for imposter probes) in Fig. 5(a) and (b). When we apply our template protection to OpenBR and OpenFace, all galleries and probes are transformed into virtual faces, and this shifts the pdfs, as may be seen in the solid blue and green lines in Fig. 5(a) and (b). The EERs also change to 0.016 and 0.028 (Table I). This is a 33% performance improvement for OpenBR, but a 40% degradation for OpenFace. Although this seems large, the absolute value of EER is still under 0.03, *i.e.* 3%, which is still considered low.

Thus far we are treating both Verifiers as black boxes, not knowing their internal workings. This ignorance is sufficient to improve the EER of OpenBR, but alas, not for OpenFace. Might we do better with more knowledge? Indeed. To pursue this idea, we exploit the fact that the OpenFace algorithm is well known. In particular, we observe that OpenFace's cropping of face image is not sufficiently tight: some non-face background may be seen in the cropped image. We exploit this fact by tweaking our Face Renderer to render user-specific colored backgrounds. Examples are shown in Fig. 6.

We re-ran our experiments using colored backgrounds. This time, OpenFace's EER only increased from 0.021 to 0.023, a mere 10% degradation (see Fig. 5(c) and middle section of Table I). Moreover, when comparing such colored virtual faces against their originals, the EER increased (as it should) to 0.047, confirming that virtual faces do not resemble their originals.

One obvious question is: does our virtual face resemble its original face? Perceptually, there is no resemblance, as the examples in Fig. 8 show. This is also reflected in the red pdfs in Fig. 5(a), and (b). These pdfs show that virtual faces appear as imposters when compared against the original galleries. The EERs are also higher: 0.035 for OpenBR, and 0.044 for OpenFace, demonstrating that our virtual faces do not look like the original faces from which they were derived.

2) Stolen-token Scenario: This scenario is designed to evaluate the security of a user's virtual identity when his token is lost. A secure token should not leak any identity information, whether (i) the original face, or (ii) the virtual face. Since our tokens store only the user-specific rotation matrix and not the original face image, there is no leakage of (i). To evaluate (ii), we proceed as follows:

First, all users are enrolled and allocated with user-specific tokens, as in the Normal scenario. Then one user's token is exposed, and all other users attempt to use the exposed token to impersonate the victim. Every user takes turn at being victim. We record the similarity scores for all these impersonation tests, and plot the pdfs in black in Fig. 5(a) and (c). It is clear that both black curves approximate the solid green curves, *i.e.* stealing a token is the same as regular impersonation. This is also corroborated by the EERs: 0.018 for OpenBR, and 0.031 for OpenFace, which are comparable to the EERs for virtual-face impersonation (last row of Table I).

3) Revocation Scenario: Revocation happens when a user wishes to change his token, whether or not his token is compromised. A secure revocation should not only replace an old token with a new one, it must also ensure that the old token can no longer be used (whether by the same user, or by others).

Fig. 8: Face examples. The first two columns show the original and corresponding virtual faces, respectively. Subsequent columns show virtual faces in different revocation sessions.

To achieve this, we mark as invalid the cluster center pointed to by the old token (using the rotation matrix), generate a new cluster center, and then update or replace the token with the new rotation matrix. If the user attempts to use his old token, his input face will be projected to the invalid cluster center, and our system can flag this anomaly. If someone else tries to use the old token, this will show up as a Stolen-token impersonation attempt, which will not succeed, as discussed in the previous section.

To evaluate this scenario, we do the following. (1) Enroll all users as in the Normal scenario; (2) Randomly choose one user and revoke his token; (3) Test the system with genuine and imposter probes to determine the FAR and FRR. Steps (2) and (3) are repeated up to 200 times, after which we plot the FAR and FRR *vs*. the number of revocations (Fig. 7). The purpose is to assess how multiple revocations affect error rates. A user may be multiply revoked because of Step (2).

In the figure, the red lines show the FAR (solid) and FRR (dotted) for OpenFace. Both lines start at the EER (because we set the decision threshold θ to achieve this), and then fluctuate within a narrow range (about 0.01) as the number of revocations is increased. In fact, FRR decreases slightly. This indicates that the error rates for OpenFace are relatively stable and unaffected by revocations. Not so for OpenBR. In the same figure, the green lines show the FAR and FRR for OpenBR. Here, it is clear that FAR increases but FRR decreases with more revocations. We observe the same trend in the blue lines, which are the FAR (solid) and FRR (dotted) for OpenBR when we start at FAR=0.008 (by adjusting θ). We surmise that this behavior is because OpenBR is sensitive to cluster centers becoming closer to one another. The latter phenomena is due to the fact that our system has a capacity of 72 revocations, after which cluster centers are generated only 60° apart, down from 90° (see Section III-C).

V. Conclusion

To the best of our knowledge, our template protection scheme is the first of its kind. This ability to guarantee irreversibility, revocability, and unlinkability for *any* Face Verifier, while maintaining good verification performance, has not been reported in the literature. We achieve this by rendering user-specific virtual faces, which are carefully placed far apart from one another in MMDA's identity subspace. While our experimental results on OpenBR and OpenFace are encouraging, it would be nice to provably guarantee that performance will not worsen for all Face Verifiers. We intend to pursue this in future work. Another area of improvement is to remove the capacity limit (see Section III-C) in our scheme, so that infinitely many revocations are permitted. Still another improvement is to guarantee irreversibility when both the token and virtual face are stolen.

Acknowledgment

This research is supported by the National Research Foundation, Prime Ministers Office, Singapore under its International Research Centre in Singapore Funding Initiative.

References

[1] K. Nandakumar and A. K. Jain, "Biometric template protection: Bridging the performance gap between theory and practice," *IEEE Signal Processing Magazine*, vol. 32, no. 5, pp. 88–100, 2015.

[2] C. Rathgeb and A. Uhl, "A survey on biometric cryptosystems and cancelable biometrics," *EURASIP Journal on Information Security*, vol. 2011, no. 1, pp. 1–25, 2011.

[3] R. M. Bolle, J. H. Connell, and N. K. Ratha, "Biometric perils and patches," *Pattern Recognition*, vol. 35, no. 12, pp. 2727–2738, 2002.

[4] M. Savvides, B. V. K. V. Kumar, and P. K. Khosla, "Cancelable biometric filters for face recognition," in *ICPR*, 2004, pp. 922–925.

[5] A. B. Teoh, A. Goh, and D. C. Ngo, "Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs," *PAMI*, vol. 28, no. 12, pp. 1892–1901, 2006.

[6] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *ACM CCS*, 1999, pp. 28–36.

[7] M. V. D. Veen, T. Kevenaar, G.-J. Schrijen, T. H. Akkermans, and F. Zuo, "Face biometrics with renewable templates," in *Electronic Imaging 2006*, 2006, pp. 60 720J–60 720J.

[8] H. Lu, K. Martin, F. Bui, K. Plataniotis, and D. Hatzinakos, "Face recognition with biometric encryption for privacy-enhancing self-exclusion," in *ICDSC*, 2009, pp. 1–8.

[9] A. Juels and M. Sudan, "A fuzzy vault scheme," *Des. Codes Cryptography*, vol. 38, no. 2, pp. 237–257, 2006.

[10] Y. Wu and B. Qiu, "Transforming a pattern identifier into biometric key generators," in *ICME*, 2010, pp. 78–82.

[11] K. Nandakumar, A. K. Jain, and S. Pankanti, "Fingerprint-based fuzzy vault: Implementation and performance," *TIFS*, vol. 2, no. 4, pp. 744–757, 2007.

[12] Y. Sutcu, Q. Li, and N. D. Memon, "Protecting biometric templates with sketch: Theory and practice," *TIFS*, vol. 2, no. 3, pp. 503–512, 2007.

[13] Y. Wang and K. Plataniotis, "Fuzzy vault for face based cryptographic key generation," in *Biometrics Symposium*, 2007, pp. 1–6.

[14] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[15] H. Lee, C. Lee, J.-Y. Choi, J. Kim, and J. Kim, "Changeable face representations suitable for human recognition," in *ICB*, 2007, pp. 557–565.

[16] T. Sim, S. Zhang, J. Li, and Y. Chen, "Simultaneous and orthogonal decomposition of data using multimodal discriminant analysis," in *IEEE International Conference on Computer Vision*, 2009, pp. 452–459.

[17] J. C. Klontz, B. F. Klare, S. Klum, A. K. Jain, and M. J. Burge, "Open source biometric recognition," in *BTAS*, 2013, pp. 1–8.

[18] B. Amos, B. Ludwiczuk, J. Harkes, P. Pillai, K. Elgazzar, and M. Satyanarayanan, "OpenFace: Face Recognition with Deep Neural Networks." [Online]. Available: http://github.com/cmusatyalab/openface

[19] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.

[20] B. Klare, "Spectrally sampled structural subspace features (4SF)," *Michigan State University Technical Report, MSU-CSE-11-16*, 2011.