Robust Biometric Scheme Against Replay Attacks Using One-Time Biometric Templates

Tanguy Gernot, Christophe Rosenberger

Robust Biometric Scheme Against Replay Attacks Using One-Time Biometric Templates

Tanguy Gernot, Christophe Rosenberger

Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France
firstname.surname@ensicaen.fr

Abstract

User authentication is an important issue on the Internet and usually solved through static and often unique passwords. Another method is to use biometrics, but biometric data are sensitive and need to be protected. Protection schemes such as cancelable biometric template generation has appeared, but they are sensitive to replay attacks. In this paper, we propose an original method to generate one-time biometric templates for user authentication applications. This proposed scheme limits replay attacks, consisting of an attacker maliciously retransmitting an intercepted user's identity proof. Our method is generic: any biometric modality could be used, the identity verification is realized by the service/identity provider to be realistic. Biometric features are extracted from captures using deep learning and then protected with biohashing, a cancelable biometric scheme. Finally, a step consisting of cryptographic hashing and symmetric encryption guarantees the generation of a one-time, non-replayable template. We have tested our scheme on two common biometric databases, from faces and fingerprints, and the results confirm its efficiency and robustness to attacks given a rigorous threat model.

Keywords— Authentication, Access Control, Deep learning, Face, Fingerprint, Template protection, Replay attack, Biometrics.

1 Introduction

User authentication is a key security component for logical access control to any digital services on the Web. Many applications are concerned such as social networks, online gaming, emails or e-payment. As for example, from September 14th 2019, the online Payment Services Directive (PSD2 - DSP2) is now in place at the EU level. The goal is to increase the level of security while providing a usable solution for consumers. This tendency tends to be generalized to any digital services.

Biometrics is increasingly used to enhance security at the level of confidence of user authentication, many solutions are now available on smartphones or laptops (finger-

print, face, behavior...). A biometric authentication scheme takes place in two steps: the first one involves registering the user (enrollment) in the biometric system on the basis of a reference biometric data, and the second step involves comparing a newly acquired biometric data with the previously stored reference (verification). It is important to remember that biometric data is personal data, and great care must be taken to protect it.

Concerning security, many attacks exist mainly to impersonate a specific user. Replay attacks are realized by resending a previous used bona fide biometric sample to the system. Presentation attacks consist in presenting to the biometric sensor a designed fake biometric sample. Many strategies are well known as for example for the face biometric modality such as using a picture of user's face to impersonate, makeup, use of masks [1]. We detail later in the paper other attacks but this shows the importance of biometric probe authenticity before making a verification decision on users' identity.

Considering privacy, biometric data could reveal a lot of information on an individual (traits, link between different digital identities...). The protection of biometric data must be permanent and cover the entire cycle of use of such data. Biometric reference storage must be secure, as must process and comparison of two biometric data sets. Cryptographic hashing, as used for passwords, cannot be used because biometric data differ slightly from one capture to the next. Encryption cannot be used either, as it would imply decryption for comparison purposes, and thus vulnerability and periodical data clarification. Biometric template protection schemes have been proposed to solve this problem [2, 3]. We believe that the use of biometric data necessarily requires the application of such template protection schemes (software). as illustration, many smartphone manufacturers propose a fingerprint sensor where the biometric templates are stored in a secure enclave in the chip of the mobile (hardware protection). Most of biometric-based authentication schemes are defined in a generic context and are not really usable for practical applications. They only use classical cryptography on biometric data or are not applicable on many biometric modalities. Moreover, their efficiency and security are not evaluated on real devices [4].

Identity verification can be done by two main approaches. In the first case, the decision is made offline by a terminal (biometric authentication on a smartphone, payment terminal...). The service/identity provider has to trust the decision result (security audit, device certification). In the second case, the service/identity provider makes the decision based on a risk policy. This is not so easy when using biometrics considering security and privacy trends as mentioned previously. Nevertheless, we focus in this paper on this last approach. The main contribution of this paper is to propose an original method to generate one-time features for biometric recognition. All biometric features are protected by a template protection scheme (privacy protection). This method permits to avoid many security attacks such as identity theft or replay attack, while meeting many privacy trends, such as unlinkability and revocability of biometric data as in [5, 6]. This generic scheme can be applied to any biometric modality and can be easily implemented on mobile devices or any physical hardware. The proposed use case in this paper handles face and fingerprint recognition using deep learning features. This solution makes it possible to acquire a new property for an existing protection method. It is simple to implement and does not interfere with other security properties already studied for the protection algorithm used. The biometric protection template can be computed in a secure element (SIM card of

the smartphone, TPM on a laptop), ensuring an end-to-end security of the biometric data. We believe this solution tends to propose at the same time a secure, privacy compliant and usable user authentication system and could meet FIDO requirements[1].

This paper is organized as follows. Section 2 describes the security and privacy threat model we designed for this work. We present in Section 3 the related works in this topic. Section 4 is dedicated to the proposal of the generation method of one-time biometric protected features. Section 5 concerns the experimental protocol and presents the used biometric datasets. Section 6 concerns the validation of the proposed method through a quantitative and qualitative analysis. Experimental results have been obtained on two significant biometric datasets. Finally, we conclude and give some perspectives in section 7.

2 Security and privacy threat model

In this paper, we consider the same environment as the framework proposed in [7], with authentication between a client and a service/identity provider. The authentication is then realized online and not locally on a physical device possessed by the client. This requirement implies a biometric database, which is complex to manage considering privacy. We do not want to encourage the use of a centralized biometric database, but we think that this assumption is realistic. In this case, a strong attention is required on the security of biometric data, against external attackers but also against a much too curious or corrupted service provider.

The following requirements are expected in most of the strong authentication schemes. We distinguish security requirements on the authentication (S_1 to S_6) and security requirements on biometric data protection (B_1 to B_6), because this later could be more important than the former, depending on the application that requires the authentication process.

- S_1 - Protection against masquerade attack: An attacker eavesdropping all the communications is not able to gain unauthorized access.
- S_2 - Protection against the replay attack: An attacker should not be able to reuse a previous authentication proof.
- S_3 - The knowledge of temporary information as time-stamp or counter has no effect on security.
- S_4 - Forward secrecy: the leakage of secret keys at a time should not cause security failure on previous authentication.
- S_5 - Weak passwords: The use of weak passwords has no effect on the security of the authentication.
- S_6 - Password change and choose easily phase for the user, without any change for the service provider.
- B_1 - Revocability: it should be possible to revoke the biometric data if they are compromised.
- B_2 - Unlinkability: an impostor should not be able to link different biometric data or accounts.

[1] `https://fidoalliance.org/specs/biometric/requirements/`

- B_3 - Non invertibility: it is not possible to recover the raw biometric data (even for the service provider).
- B_4 - Generalization: any biometric data could be used within the proposed scheme.
- B_5 - Sensitive biometric data should not be recovered if a weak password is used and the device stolen.
- B_6 - Biometric authentication is realized by the service provider and not by the client.

The proposed scheme should meet all these requirements.

3 Related works

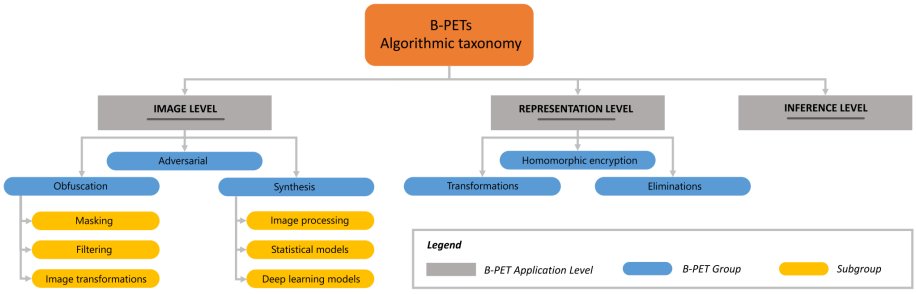


Figure 1: Taxonomy of privacy preserving algorithmic solutions for face recognition (source [8]).

User authentication schemes have been extensively studied over the last two decades, as for example in [9, 10] with various security assumptions, different cryptographic algorithms (from symmetric cryptography to elliptic curves) and a lot of cryptanalysis on them. In [4], authors classify attacks on user authentication in the following categories:

1. Brutal-force and Guessing Attacks: the attacker tries to impersonate the user by testing different random values of identity proof (such as a password),
2. Observation Attacks: any observation could help the attacker to guess the identity proof,
3. Impersonation Attacks: the attacker tries to generate an identity proof (called presentation attack in biometrics),
4. Side-channel Attacks: some information leakage of authentication systems can be exploited in order to generate a fake identity proof,
5. Authentication Model Attacks via Adversarial Examples: this attack has for objective to generate an adversarial example, which is not the user's identity proof but can still be recognized by the system and granted the access.

Many methods have been proposed for biometric authentication using homomorphic

encryption (generally Paillier encryption) for biometric data protection [3, 11]. In the case of an external mobile authentication, the authentication is realized in a secure multi-party computation protocol between the mobile, the server and possibly a third party. The most proposed use case is the computation of a distance between behavioral biometric data (or location data/usage profile) with garbled circuits and oblivious transfers. We refer the reader to previous references for technical details. One of the main drawbacks of all these papers is that the mobile is assumed to be secure. The implementation of these algorithms on a secure element is not possible.

Authors in [12] define a face security authentication system based on deep learning and homomorphic encryption. In this case, the identity verification is realized by the identity/service provider while obtaining on the LFW dataset an identification accuracy of 98.7%. Meden et al. [8] propose a very interesting comprehensive review on privacy–related research in the area of Privacy–Enhancing Techniques (PET) applied to face biometrics. Figure 1 provides a taxonomy of existing algorithmic solutions for preserving privacy for face recognition. Liu et al. [13] proposed privacy by design solutions for face identification. A cryptographic protocol is used for making privacy query to a database. The identification performance on small face datasets is good (97% of correct recognition) even used face features are eigenfaces (that are well known and provide poor results in general). Performance for user authentication is not given in the paper. Authors in [14] propose a privacy preserving solution for face recognition based on federated learning. Given a pre-trained face model, authors aim to simultaneously improve the generic face representation at the server side, and produce an optimized model for each client without transmitting any private identities' images or features from local devices. Wang et al. [15] propose a new approach called FaceMAE dealing privacy with deep architectures. This framework generates synthetic samples that reduce the privacy leakage. It also maintains recognition performance simultaneously. Performance on face recognition is very good (99.2% on the LFW dataset). Recently, authors in [16] proposed a privacy compliant face feature generator based on fuzzy vault. The performance on FERET and FRGCv2 databases (subsets) for authentication is very good ($< 1\%$ FNMR at $< 0.01\%$ FMR). The computation time (decoding) could be an issue in a mobile platform.

Most works in the literature consider the biometric identification problem while preserving privacy. We are more interested in our work on the authentication problem. We intend to generate a privacy biometric proof on user's identity. Second, the replay attack is also rarely considered in the literature, it can be considered as a naive presentation attack. None of the previous listed works consider this aspect that is crucial for many applications such as e-payment.

4 Proposed method

In this section, we consider a client (Bob) with a terminal having a webcam for the face capture (smartphone, laptop, tablet) or a fingerprint sensor. We present in the following the used biometric features and the different solutions used to guarantee the security and privacy that are very important issues. Then, we detail the enrollment and verification processes (see Figures 2 and 3). Some steps can optionally be done inside a secure element (SIM card, Trusted Platform Module) for more security.

4.1 Features extraction

Nowadays, biometric features are mainly computed through the application of a deep network. We can cite face descriptors [17], ear [18], palm print [19] ... In this paper, we consider having a CNN that generates fixed size feature vectors as a representation of the biometric modality.

4.2 Privacy protection

We designed the proposed solution by considering privacy as we use biometric data. We use the representation-level approach for privacy in this work (see Figure 1).

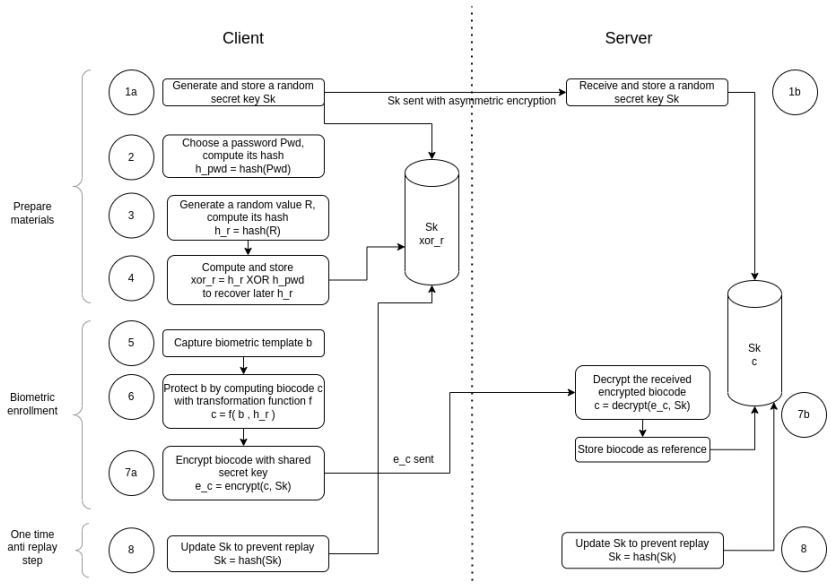


Figure 2: Enrollment step.

4.2.1 Transformation based template protection

We focus in this part on biometric template protection schemes based on the transformation of biometric data, called feature transformation [20]. For more details, we refer the reader to the surveys [21, 22]. A feature transformation is a function f using a secret key k applied to a biometric template b. During enrollment, the transformed template $f(b, k)$ is stored in a centralized database or in a decentralized database as a personal device. During the authentication, the same transformation is performed to the query fresh template b' with the same key k. Ultimately, a comparison is realized between $f(b, k)$ and $f(b', k)$ by verifying if $D_T(f(b, k), f(b', k)) \leq \epsilon$, where D_T is a distance function in the transformed domain, ϵ is the threshold value set by the system administrator. In this paper, $f(b, k)$ is called the reference BioCode and $f(b', k)$ is called the captured BioCode.

The performance of authentication systems is generally estimated with False Match Rate (FMR) computing the ratio of successful impostor verification and with the False Non-Match Rate (FNMR) calculating the ratio of false rejection of legitimate users. We assume that the feature transformation used in our scheme should not decrease the performance of the system. Experiments are performed at the end of this paper for the verification of this assumption.

It is generally considered that, given several templates $f(b_1, k), \ldots, f(b_n, k)$ and the corresponding key k, where $b_1, \ldots, b_n$ come from the same user, it is possible to reconstruct a close approximation of the original template or at least execute a pre-image attack. It is not certain if the reconstruction of the original template is systematically possible, given one template $f(b, k)$ and the key k, and it strongly depends on the used biometric modality and the transformation. Nevertheless, it is sometimes possible to recover an approximation of b, in the sense that it is possible to create b' such that $f(b', k)$ is close to $f(b, k)$. In this paper, we assume that, given $f(b, k)$, the biometric data b can be recovered if and only if k is known. This assumption is only used to analyze the security of our scheme if the server or the secret keys are compromised, because all transformed biometric data are encrypted in the database or during the communications between the client and the server.

4.2.2 BioHashing algorithm

The BioHashing algorithm is a feature transformation defined in [23]. The distance D_T used to evaluate the similarity between two transformed templates (the reference BioCode and the captured BioCode) is the Hamming distance. This algorithm is generic and can be applied to any biometric modalities, as long as features can be extracted and represented as a fixed-size vector. It transforms the template b of length n in a binary template of length m as following:

1. Generate m pseudo-random orthonormal vectors $v_1, \ldots, v_m$ of length n, from the secret key k (typically with the Gram Schmidt algorithm).
2. For $i = 1, \ldots, m$, compute the scalar product:
   $x_i =< b, v_i >$.
3. Output the binary template $(t_1, \ldots, t_m)$ with the quantization process:

$$t_i = \begin{cases} 0 & \text{if} \quad x_i < \tau \\ 1 & \text{if} \quad x_i \geq \tau, \end{cases}$$

   where τ is a given threshold, generally equal to 0.

The performance of this algorithm is ensured by the scalar products with the orthonormal vectors, as detailed in [24]. The projection of the template in a lower dimension ($n \geq m$) and the threshold binarization ensures the non-invertibility of the data, especially if the secret is unknown. None attack on this feature transformation has been reported if the secret key is not compromised. Nevertheless, given one or several BioCodes with the corresponding secret key, some attacks have been presented in [22, 25, 26], for the reconstruction of an approximation on the biometric data. It corresponds to our security assumption of the previous section on the feature transformation. The main idea of this work is to change the secret key (one-time use) at any transaction to avoid these attacks. In the following, we detail the proposed authentication cryptographic protocol for each step (enrollment and verification).

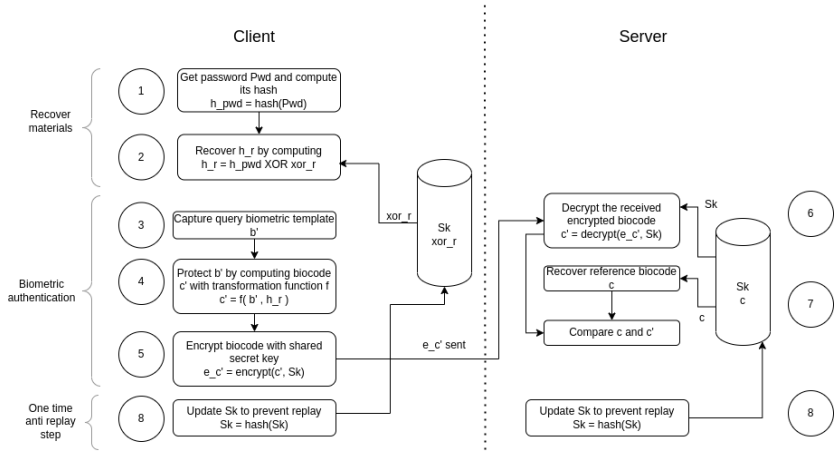


Figure 3: Verification step.

4.3 Enrollment

The objective of the enrollment step is the generation of materials for future authentication of Bob. Figure 2 presents the different steps of the enrollment process we detail in the following:

1. First, we prepare some materials to secure biometric data and to avoid a replay attack:
   - As the preamble of the enrollment process, the client and the server should share a symmetric secret key Sk in the steps $1a$ and $1b$, and they store it.
   - The client chooses in step 2 a password Pwd and computes its hash h_pwd. It will be used to securely store the transformation's seed.
   - The client generates a random value R in step 3. He computes its hash h_r which will be used as a seed for the BioHashing algorithm.
   - In step 4, the client computes xor_r the xor between h_r and h_pwd. It is stored to lately recover h_r. In fact, the seed h_r of the transformation must not be corrupted, as this allows reconstruction attacks of the BioCode.
2. The second step involves biometric enrollment:
   - In step 5, the client captures a biometric template b.
   - In step 6, the reference BioCode c is computed from b transformed by the function f (like Biohashing) with the seed h_r.
   - Lastly, in the step $7a$, the client encrypts the Biocode c with the shared symmetric key Sk. It sends the result e_c to the server which decrypts its with Sk and stores the BioCode c as reference in step $7b$. All encryption schemes used in this paper are realized in a randomized mode with authentication (as Cipher Block Chaining (CBC) with Message authentication code (MAC)). Note that the server is not able to recover the biometric template b because it does not know the secret key h_r.

3. Ultimately, the client and the server update Sk by hashing it in step 8 to prevent replay attack. This step ensures that an encrypted biocode captured by an adversary and then replayed cannot be authenticated. This is because the server-side decryption with the updated Sk key will fail.

4.4 Verification

The objective of this step is the verification of Bob's identity by providing a captured BioCode that is compared at the end with the supposed client's reference BioCode. Figure 3 presents the different steps of the authentication process:

1. First, we recover the data required for biometric processing:
   - In step 1, the client gives his password Pwd, that is supposed to be the same as in enrollment step. He computes the hash of the password h_pwd.
   - In step 2, the client recovers the seed h_r by computing the xor between h_pwd and the stored xor_r. If the password is incorrect, the recovered seed will be incorrect, and so the biocode transformed from it will not allow authentication.
2. The second step involves biometric authentication:
   - The client captures, in step 3 the query template b'.
   - In step 4, the client protects b' with the transformation function f. The seed's used is the recover seed h_r from xor_r and h_pwd.
   - Lastly, in the step 5, the client encrypts the Biocode c' with the shared symmetric key Sk. It sends the result e_c' to the server which decrypts its with Sk in step 6.
   - In step 7, the server recovers the reference biocode c and the servers compares c and c'. If the Hamming distance between the two BioCodes is lower than a given threshold, the client is authenticated and rejected in the other case. The value of the decision threshold depends on the security policy defined by the server.
3. Ultimately, the client and the server update Sk by hashing it in step 8 to prevent replay attack.

5 Experimental protocol

We first detail the experimental protocol and we define the biometric databases and parameters settings for the validation of the proposed solution.

5.1 Datasets

In this work, we consider two well known and largely used biometric modalities: face and digital fingerprint. Note that the proposed solution could be applied by any biometric modality.

In this study, we used a large and significant biometric dataset called Labeled Faces in the Wild (LFW) [27]. This database of face photographs has been produced for

studying the problem of unconstrained face recognition. The data set contains more than 13,000 images of faces collected from the web. This dataset contains images from 5749 "famous" individuals. Image resolution is low as each sample is composed of 250×250 pixels. Figure 5 shows samples images for Bill Gates in the dataset.

Figure 4: Examples of images in the LFW dataset (samples for Bill Gates).

We used in this work a dataset from the FVC competition in 2002 [28]. This database contains digital fingerprint images from 100 users with 8 samples per user. Image resolution is medium as each sample is composed of 288×384 pixels. Figure 6 shows samples images 3 different users.

5.2 Biometric features

In order to generate face features, we use the following processing chain. The first step is to detect the face in the image. We use the MTCNN algorithm [29] based on deep learning. After detection, we then resize the face patch to 112×112 pixels. In this paper, we use a very efficient face recognition system based again on deep learning [17]. Figure 2 presents the deep network architecture. The feature vector size as output is 1024.

To our knowledge, we do not know any deep architecture for the generation of features for digital fingerprints. We used a simple CNN architecture, namely AlexNet (see Figure 5.2) and we trained it (transfer learning) with synthetic fingerprints generated by the well-known SFinge fingerprint generator [30]. We used 20,000 images from 2000 simulated users. After training, we obtain a feature vector of size 2000 for any fingerprint image.

Figure 5: Examples of images in the FVC 2002 DB4 dataset (samples for 3 different users).

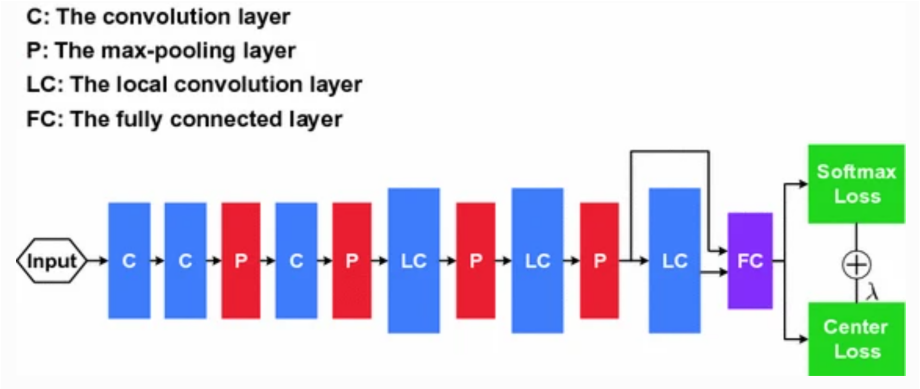


Figure 6: Deep architecture for face feature extraction (source [17]).

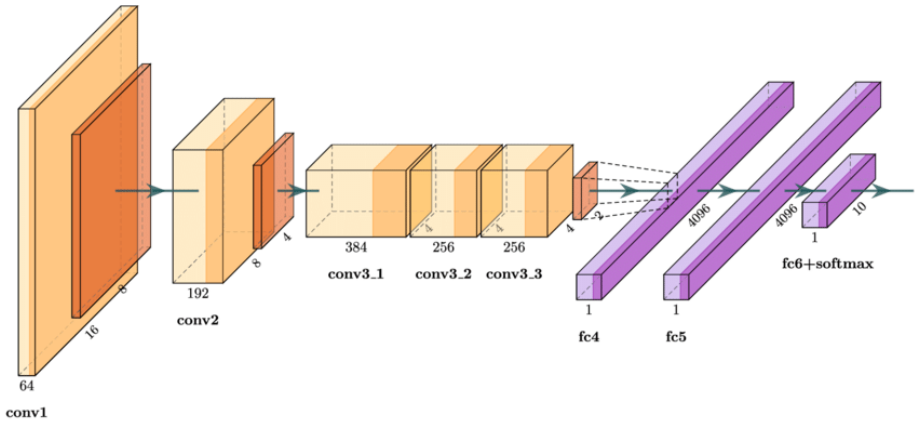


Figure 7: Alexnet architecture for fingerprint features generation (source [31]).

5.3 Protection settings

In this work, we use BioCodes of size $m = 256, 512, 768$ bits. As $n = 1000$ (feature size), the BioCode size m has to be lower than n for security reasons (irreversibility). We used the following methodology to evaluate the proposed scheme:

- The first capture of each user is taken as the reference biometric data from which we calculate the Reference Biocode.
- We compute legitimate scores and impostor ones (comparison between reference templates and probes for the same user (resp. other users)) using raw features and protected ones,
- The comparison is realized with the cosine distance for raw features and the Hamming one for protected features,
- Based on these two sets of scores, we analyze the imposters' ability to impersonate legitimate users according to different thresholds, and calculate the equal error rate (EER), an indicator widely used in biometrics to assess the robustness of a biometric system when $FMR = FNMR$ [32].
- We considered two scenarios for testing the robustness of the proposed method. In scenario 1, we suppose the impostor does not know the secret/password (requested to compute A value and the *Captured BioCode*) and uses its own biometric data (Zero effort attack) to impersonate Bob (impersonation attack). In scenario 2, we assume the impostor also knows the password (side-channel attack).
- We run a simulation with 1000 different values of K to demonstrate the variability of BioCodes for the same individual (privacy protection).

6 Experimental results

In this section, we present the obtained results by analyzing the performance of raw and protected features. We then realize a qualitative analysis of the propose solution

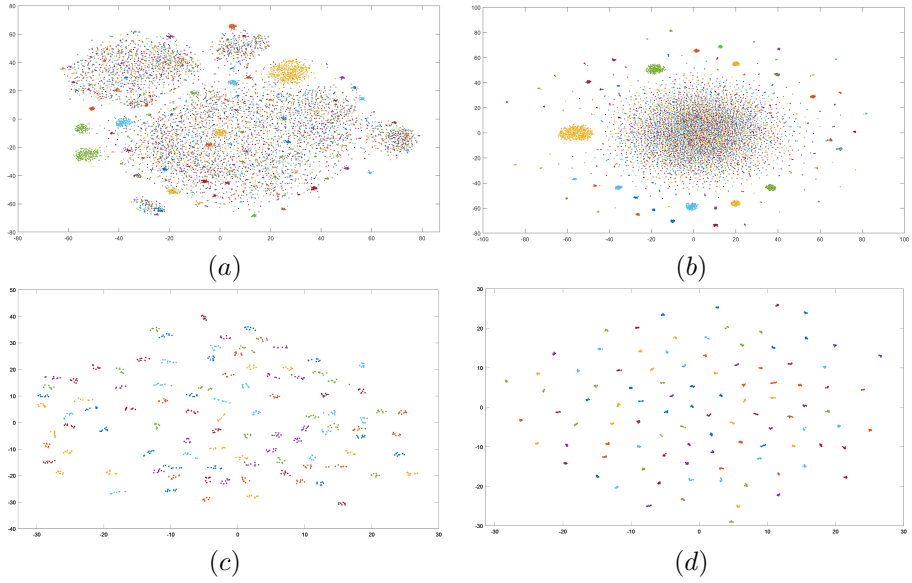


Figure 8: Projection of biometric features on the two principal axis: (a) Raw face features, (b) Protected face features on the LFW dataset, (c) Raw fingeprint features, (d) Protected fingerprint features on the FVC2002 DB4 dataset.

with the threat model we defined in section 2. We finally compare the proposed solution with other works from the literature.

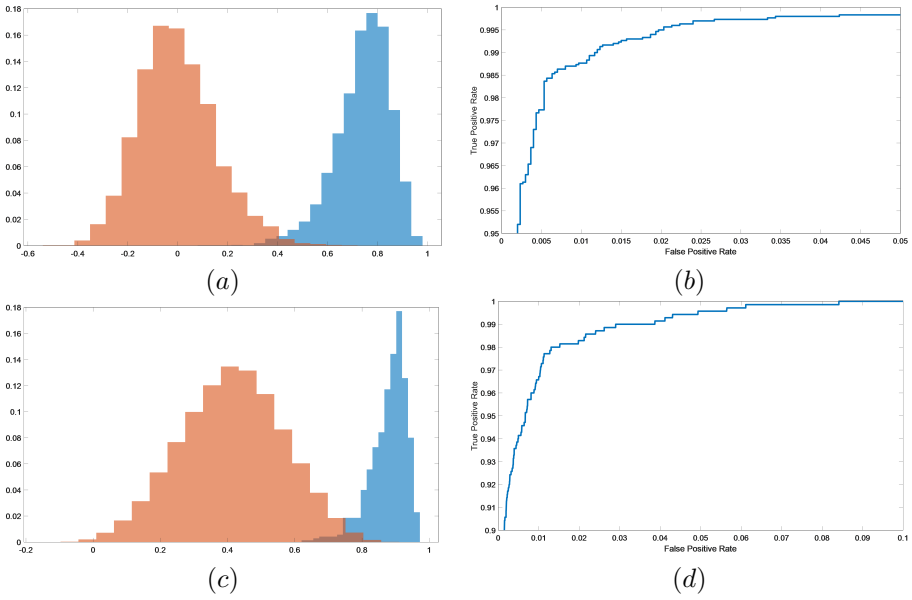
6.1 Performance with raw features

First, we analyze the performance of raw biometric features without any template protection in a static context. In order to show the capability to discriminate individuals with deep features, we plot in Figure 8 their projection in 2D. This projection has been obtained by using the t-Distributed Stochastic Neighbor Embedding algorithm [33].

Figure 8 (a) demonstrates the ability of used face features to distinguish the 5749 individuals in the LFW dataset. We compute first the accuracy of face recognition (identification context) when using deep face features. We use the cosinus distance for the comparison between the reference templates and the probe one. We obtain the accuracy of 98.97% showing the efficiency of used features. We now consider the authentication context by measuring the EER value equal to 1.1%. Figure 9(a) shows the distribution of legitimate and impostor scores. Figure 9(b) presents the obtained ROC curve. These results demonstrate the efficiency of used face features in this work.

Concerning fingerprint features, we plot in Figure 8 (c) their projection in 2D with the t-SNE method. This figure demonstrates the ability of used fingerprint features

to distinguish the 100 individuals in the FVC2002 DB4 dataset. We compute first the accuracy of fingerprint recognition (identification context) when using the proposed deep fingerprint features. We use the cosinus distance for the comparison between the reference templates and the probe one. We obtain the accuracy of 99.78% showing the significance of used features. We now consider the authentication context by measuring the EER value equal to 1.8%. Figure 9(c) shows the distribution of legitimate and impostor scores. Figure 9(d) presents the obtained ROC curve. These results demonstrate the efficiency of the proposed fingerprint features.



(a) (b) (c) (d)

Figure 9: Performance using raw features: (a) Distribution of legitimate and impostor scores for face features on the LFW dataset, (b) ROC curve by using face features, (c) Distribution of legitimate and impostor scores for fingerprint features, (d) ROC curve by using fingerprint features.

6.2 Performance with protected features

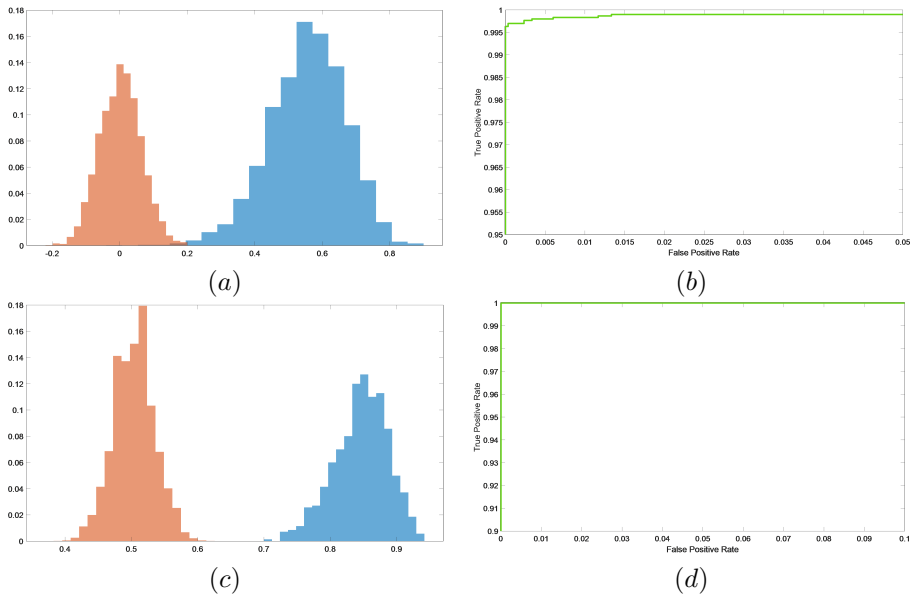
In this section, we quantify the performance of the biometric features after protection ($m = 256$ bits). In order to show the capability to discriminate users in each dataset with protected deep features, we plot in Figures 8 (b) and (d) their projection in 2D. These figures demonstrate that protected features keep their capacity to discriminate users (note that we used a different secret for each user).

6.2.1 Scenario 1: normal context

We compute first the accuracy of face recognition (identification context) when using protected deep face features in the scenario 1 context, i.e. when the impostor does

not know the secret. We use the Hamming distance for the comparison between the reference template and the probe one. We obtain the accuracy of 99.8% illustrating the efficiency of protected features. We now consider the authentication context by measuring the EER value equal to 0.2%. Figure 10(a) shows the distribution of legitimate and impostor scores. Figure 10(b) presents the obtained ROC curve. These results demonstrate the efficiency of protected face features.

We now consider fingerprint recognition when using protected deep fingerprint features. We use the Hamming distance for the comparison between the reference templates and the probe one. We obtain the accuracy of 100%. We now consider the authentication context by measuring the EER value equal to 0%. Figure 10(c) shows the distribution of legitimate and impostor scores. Figure 10(d) presents the obtained ROC curve. Once again, in this scenario, we obtain excellent results.

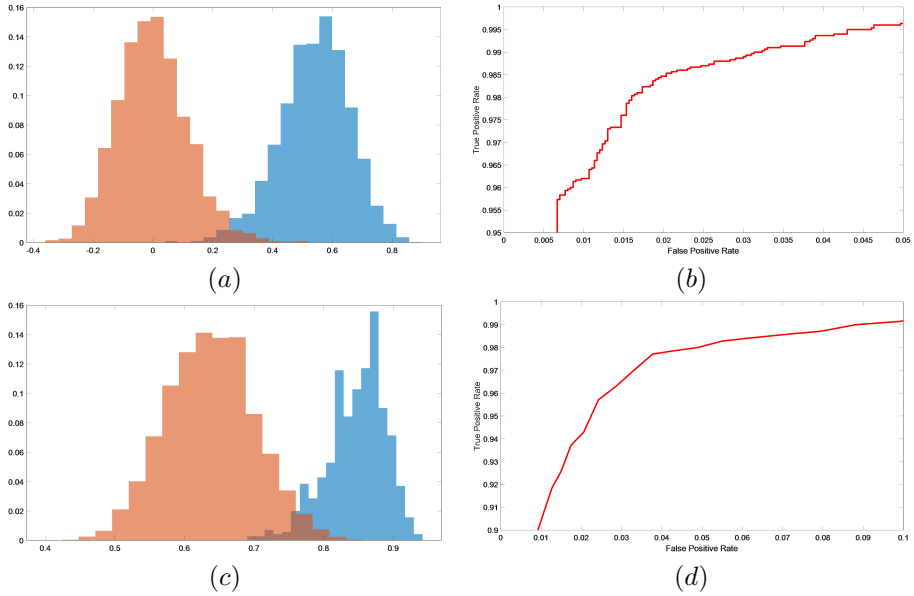


(a) (b) (c) (d)

Figure 10: Performance of protected biometric features in scenario 1 context (first row: face, second row: fingerprint) (a-c) Distribution of legitimate and impostor scores, (b-d) ROC curve.

6.2.2 Scenario 2: attack context

Now, we consider the scenario 2, i.e. the impostor knows the secret. In this case, we do the same experiments with the same secret for all users in the dataset. The accuracy for face identification remains high with 98.2% and the EER value is 1.76% in this context. for fingerprints, the accuracy for identification remains high with 99.6% and the EER value is 3.3% in this context. The proposed solution remains efficient even in case of attack.



(a) (b) (c) (d)

Figure 11: Performance of protected biometric features in scenario 2 context (first row: face, second row: fingerprint): (a-c) Distribution of legitimate and impostor scores, (b-d) ROC curve.

6.2.3 Impact of bioCode size

We now study the impact of the BioCode size on the performance and robustness. Table 1 presents the obtained accuracy and EER values for the two scenarios. Note that with unprotected biometric data, the accuracy was 98.97% and the EER value equal to 1.1%. We can see clearly that the higher is the BioCode size, the better are the performance and robustness. In order to have a better irreversibility, choosing a BioCode size of 512 bits is a good compromise. Note that in the proposed method, the value K (secret for the template protection scheme) also changes each time for irreversibility.

6.3 Qualitative analysis

The proposed scheme can use any biometric modality (requirement B_4), the only requirement is to have a fixed size representation of the biometric data. The verification is done by the server by design (requirement B_6).

The security of the scheme is not directly based on passwords because the password is only used locally in order to start the authentication process (requirement S_5). It is also used to mask the secret key K even if this last key is securely stored in the device. We argue that it is unlikely that a user creates a very weak password if it is combined with biometrics. Nevertheless, it should be possible to see directly the password typed

	BioCode size	256 bits	512 bits	768 bits
FACE	**Scenario 1 (normal)**			
	Accuracy (identification)	99.8%	99.93%	99.95%
	EER (authentication)	0.2%	0.1%	0.06%
	Scenario 2 (attack)			
	Accuracy (identification)	98.2%	98.7%	98.8%
	EER (authentication)	1.76%	1.37%	1.23%
FINGERPRINT	**Scenario 1 (normal)**			
	Accuracy (identification)	100%	100%	100%
	EER (authentication)	0%	0%	0%
	Scenario 2 (attack)			
	Accuracy (identification)	99.6%	99.7%	99.7%
	EER (authentication)	3.3%	2.6%	2.3%

Table 1: Impact of the BioCode size on performance and robustness.

during the authentication and consequently the security should not be based on it.

Mutual authentication and non-repudiation are not considered in this paper to simplify the design of the scheme. Nevertheless, if it is required, it could be added, for example, with a public cryptographic key (requirement S_1). Finally, it is recommended to the service provider to encrypt the entire biometric database with its own secret key and to decrypt the data only for authentication purpose. This problematic is related to database encryption and is not described here. Nevertheless, if the service provider is compromised, the secret key K' or the reference BioCode used during the verification step could be compromised (the raw biometric data cannot be recovered anyway).

6.3.1 Security with non compromised secret keys

The proposed authentication scheme combines many authentication factors i.e., two secret keys, a biometric data, a password and a device (a smartphone in our use case). We assume in a first time that the secret keys K and K' cannot be compromised. Thus, if the server database is compromised, then an attacker cannot use the reference BioCode, without the knowledge of the secret key, to recover the original biometric data in order to spoof the system. Moreover, the server is not able to recover the biometric data because it does not know the secret key used by the template transformation (requirement B_5). Furthermore, if the authentication data are eavesdropped by an attacker during the authentication step, these data cannot be decrypted by the attacker without the knowledge of the secret key K'. These data cannot be replayed by the attacker because the encryption algorithm is not used in a deterministic mode (requirement S_2).

Temporary information leakage property is verified because all these data (random value, counter, time-stamp or identifiers Id, Is and It) are considered as public in the proposed scheme (requirement S_3). Clearly, if K and K' are not compromised, revocability, non-invertibility and unlinkability of biometric data are verified (requirements B_1, B_2 and B_3). Considering unlikability, we show in Figure 12 a) the projection of

1000 BioCodes generated from the same biometric sample (face feature) with different values of K. This figure shows clearly (in 2D) that the BioHashing is capable of hiding the biometric sample in the feature space. Figure 12 b) shows the distribution of pseudo legitimate scores (computed from samples from the same individual with different values of K) and pseudo-impostor ones (calculated from samples from other individuals with different values of K). It clearly shows that it is not possible to decide if the comparison is made with samples from the same individual or with another one. The forward secrecy (requirements S_4) is guaranteed thanks to the update of the K' key and the use of a hash function in step 7 of the authentication process. Finally, the proposed scheme is not protected against desynchronization attacks on K'. It is for simplicity reasons and it can be easily taken into account by the scheme.

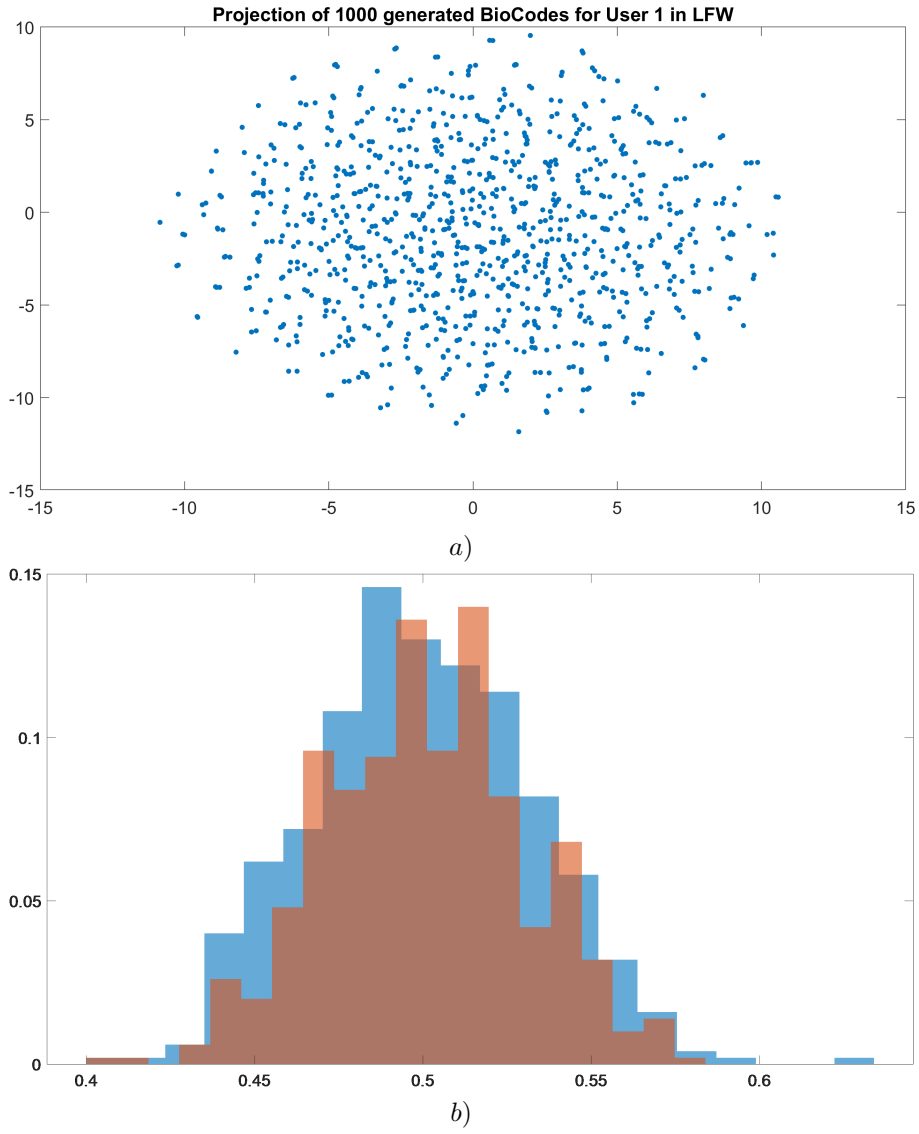


Figure 12: Unlinkability analysis: a) projection of 1000 generated Biocodes for user 1 (face data) as illustration (a Biocode is represented by a binary vector of size 256 bits), b) distribution of pseudo-legitimate and impostor scores.

Reference	*Method*	*Database*	*Performance*
[12]	*Homomorphic encryption*	*LFW, CASIA datasets*	*accuracy* 98.7%/93.4%
[13]	*Homomorphic encryption*	*FERET, ORL datasets*	*accuracy* 97.9%
[14]	*federated learning*	$MS-Celeb1$ *dataset*(*subset*)	*accuracy* 95.2%
[15]	*federated learning*	*LFW dataset*	*accuracy* 99.2%
[16]	*fuzzy vault*	*FERET, FRGCv2* (*subsets*)	$\leq 1\%$ FNMR at $\leq 0.01\%$ FMR
Proposed	*Biohashing, anti − replay attack protocol*	*LFW dataset*	*accuracy* 99.93%, $EER = 0.1\%$

Table 2: Comparison with other methods in the literature. We give results for the proposed method with $m = 512$ bits.

6.3.2 Security with compromised factors

The security of strong authentication schemes is generally evaluated with an attacker possessing all previous communications between the client and the server and with the knowledge (or the possession) of two and only two factors. The objective of the attacker is to be authenticated by the server or to recover the original biometric data of the client (if unknown).

If the attacker, with the knowledge of Pwd, steals the device and recovers the keys K' and K with side-channel attacks or physical attacks on the device, he/she cannot use K' to decrypt the previous exchanged data between the client and the server because K' has been updated in $H(K')$. Consequently, the attacker has no access to $f(b, K)$ or $f(b', K)$ and consequently cannot recover b (or b') and cannot be authenticated by the server (recalling that $f(b, K)$ is not stored on the device). Thus, if the smartphone of the client is stolen, the server realizes a biometric data revocation and both realize a new enrollment step.

If the attacker, with the knowledge of Pwd, steals a genuine biometric data b' of the client, he/she cannot use them for authentication without the knowledge of the secret keys K and K' (and consequently without the possession of the physical device). The authentication is even not possible if the attacker compromises the secret key K' of the server side because the secret key K is only stored in the physical device. If the attacker, without the knowledge of Pwd, possesses the genuine biometric data b' of the client and the physical device (i.e. with K' and $K \oplus H(Pwd \parallel I_s)$), she/he cannot recover the key K (without the knowledge of the password Pwd) (requirement S_6).

All requirements defined in section 2 are guaranteed by the proposed scheme (as it can be seen in Table 3).

S_1	S_2	S_3	S_4	S_5	S_6	B_1	B_2	B_3	B_4	B_5	B_6
$\sim$	✓	✓	✓	✓	✓	✓	✓	✓	$\sim$	✓	✓

Table 3: Summary of the security analysis.

6.4 Comparison to Related Works

In the following, we tried to position the proposed method compared to the literature in Table 2. Concerning performance, the proposed method provides very good results on the LFW dataset (composed of more than 5000 individuals) for identification and authentication applications. Another interesting contribution is the anti-replay attack protocol that is not proposed by other methods and could be useful to detect forged face features.

7 Conclusion and perspectives

We believe the proposed method can be used as a robust user authentication solution taking into account privacy by design. We showed in this paper that the obtained

performance is very good for identification and authentication contexts. The proposed method is also able to deal with replay attacks while letting the service/identity provider make the identity verification. This one-time identity proof from user's biometric data gives a better confidence on its authenticity.

As perspectives, we believe that, as mentioned in [34], approaches based on deep architectures enabling them to directly generate protected deep features are good solutions to explore. We also intend to generalize the proposed scheme to multi-biometrics and to continuous authentication schemes.

References

[1] Faseela Abdullakutty, Eyad Elyan, and Pamela Johnston. A review of state-of-the-art in face presentation attack detection: From early development to advanced deep learning and multi-modal fusion methods. Information fusion, 75:55–69, 2021.

[2] Josep Domingo-Ferrer, Qianhong Wu, and Alberto Blanco-Justicia. Flexible and robust privacy-preserving implicit authentication. In 29th IFIP 11 International Conference on Systems Security and Privacy Protection (SEC), pages 18–34, 2015.

[3] Paolo Gasti, Jaroslav Sedenka, Qing Yang, Gang Zhou, and Kiran S. Balagani. Secure, fast, and energy-efficient outsourced authentication for smartphones. IEEE Trans. Information Forensics and Security, 11(11):2556–2571, 2016.

[4] Chen Wang, Yan Wang, Yingying Chen, Hongbo Liu, and Jian Liu. User authentication on mobile devices: Approaches, threats and trends. Computer Networks, 170:107118, 2020.

[5] Aude Plateaux, Patrick Lacharme, Audun Jøsang, and Christophe Rosenberger. One-time biometrics for online banking and electronic payment authentication. In International Conference on Availability, Reliability, and Security, pages 179–193. Springer, 2014.

[6] Patrick Lacharme and Christophe Rosenberger. Synchronous one time biometrics with pattern based authentication. In 2016 11th International Conference on Availability, Reliability and Security (ARES), pages 260–265. IEEE, 2016.

[7] Xinyi Huang, Yang Xiang, Ashley Chonka, Jianying Zhou, and Robert H. Deng. A generic framework for three-factor authentication: Preserving security and privacy in distributed systems. Transactions on Parallel and Distributed Systems, 22(8):1390–1397, 2011.

[8] Blaž Meden, Peter Rot, Philipp Terhörst, Naser Damer, Arjan Kuijper, Walter J Scheirer, Arun Ross, Peter Peer, and Vitomir Štruc. Privacy–enhancing face biometrics: A comprehensive survey. IEEE Transactions on Information Forensics and Security, 2021.

[9] Aditya Sundararajan, Arif I Sarwat, and Alexander Pons. A survey on modality characteristics, performance evaluation metrics, and security for traditional and wearable biometric systems. ACM Computing Surveys (CSUR), 52(2):1–36, 2019.

[10] Matei-Sorin Axente, Ciprian Dobre, Radu-Ioan Ciobanu, and Raluca Purnichescu-Purtan. Gait recognition as an authentication method for mobile devices. Sensors, 20(15):4110, 2020.

[11] Mahesh Kumar Morampudi, Munaga VNK Prasad, and USN Raju. Privacy-preserving iris authentication using fully homomorphic encryption. Multimedia Tools and Applications, pages 1–23, 2020.

[12] Dechao Sun, Hong Huang, Dongsong Zheng, Haoliang Hu, Chunyue Bi, and Renfang Wang. Face security authentication system based on deep learning and homomorphic encryption. Security and Communication Networks, 2022, 2022.

[13] Meng Liu, Hongsheng Hu, Haolong Xiang, Chi Yang, Lingjuan Lyu, and Xuyun Zhang. Clustering-based efficient privacy-preserving face recognition scheme without compromising accuracy. ACM Transactions on Sensor Networks (TOSN), 17(3):1–27, 2021.

[14] Chih-Ting Liu, Chien-Yi Wang, Shao-Yi Chien, and Shang-Hong Lai. Fedfr: Joint optimization federated framework for generic and personalized face recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 1656–1664, 2022.

[15] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Jiankang Deng, Xinchao Wang, Hakan Bilen, and Yang You. Facemae: Privacy-preserving face recognition via masked autoencoders. arXiv preprint arXiv:2205.11090, 2022.

[16] Christian Rathgeb, Johannes Merkle, Johanna Scholz, Benjamin Tams, and Vanessa Nesterowicz. Deep face fuzzy vault: Implementation and performance. Computers & Security, 113:102539, 2022.

[17] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In European conference on computer vision, pages 499–515. Springer, 2016.

[18] Aman Kamboj, Rajneesh Rani, and Aditya Nigam. A comprehensive survey and deep learning-based approach for human recognition using ear biometric. The Visual Computer, 38(7):2383–2416, 2022.

[19] Selma Trabelsi, Djamel Samai, Fadi Dornaika, Azeddine Benlamoudi, Khaled Bensid, and Abdelmalik Taleb-Ahmed. Efficient palmprint biometric identification systems using deep learning and feature selection methods. Neural Computing and Applications, 34(14):12119–12141, 2022.

[20] N.K. Ratha, J.H. Connell, and R. Bolle. Enhancing security and privacy in biometrics-based authentication system. IBM Systems J., 37(11):2245–2255, 2001.

[21] C. Rathgeb and A. Uhl. A survey on biometric cryptosystems and cancelable biometrics. EURASIP J. on Information Security, 3, 2011.

[22] A. Nagar, K. Nandakumar, and A. K. Jain. Biometric template transformation: A security analysis. Proceedings of SPIE, Electronic Imaging, Media Forensics and Security XII, 2010.

[23] A.B.J. Teoh, D. Ngo, and A. Goh. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. Pattern recognition, 40, 2004.

[24] A. B.J. Teoh, Y. W. Kuan, and S. Lee. Cancellable biometrics and annotations on biohash. Pattern Recognition, 41:2034–2044, 2008.

[25] Patrick Lacharme, Estelle Cherrier, and Christophe Rosenberger. Preimage attack on biohashing. In Security and Cryptography (SECRYPT), 2013 International Conference on, pages 1–8. IEEE, 2013.

[26] Berkay Topcu, Cagatay Karabat, Matin Azadmanesh, and Hakan Erdogan. Practical security and privacy attacks against biometric hashing using sparse recovery. EURASIP J Adv. Sig. Proc., page 100, 2016.

[27] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[28] Dario Maio, Davide Maltoni, Raffaele Cappelli, James L. Wayman, and Anil K. Jain. Fvc2002: Second fingerprint verification competition. In International Conference on Pattern Recognition (ICPR'02), volume 3, pages 811 – 814, 2002.

[29] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE signal processing letters, 23(10):1499–1503, 2016.

[30] Raffaele Cappelli, Dario Maio, Davide Maltoni, et al. Sfinge (synthetic fingerprint generator). 2004.

[31] Nicola Strisciuglio, Manuel Lopez-Antequera, and Nicolai Petkov. Enhanced robustness of convolutional networks with a push–pull inhibition layer. Neural Computing and Applications, 32:17957–17971, 2020.

[32] BioAPI Consortium. Information technology – biometric performance testing and reporting – part 1: Principles and framework. Technical report, ISO/IEC 19795-1, 2006.

[33] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of machine learning research, 9(11), 2008.

[34] Vedrana Krivokuća Hahn and Sébastien Marcel. Biometric template protection for neural-network-based face recognition systems: A survey of methods and evaluation techniques. arXiv preprint arXiv:2110.05044, 2021.