Fuzzy MP - A Fuzzy Digital Signature Scheme with Biometrics

Tiong-Sik Ng and Andrew Beng-Jin Teoh(✉)

School of Electrical and Electronic Engineering,
Yonsei University, Seoul, South Korea
{ngtiongsik,bjteoh}@yonsei.ac.kr

Abstract. The combination of biometrics and cryptography have consistently remained an open problem due to the probabilistic nature of the former and the deterministic nature of the latter. Various fuzzy cryptosystem schemes have arisen in hopes of combining primitives for both disciplines, mostly for practical purposes. In cryptographic primitives, particularly the digital signature, most of the fuzzy signature schemes rely on the fuzzy extractor to generate keys for the algorithm. It is claimed that said schemes can be readily used with biometric inputs due to their fuzzy nature. In our proposed work which we coin the Fuzzy MP scheme, we use an approach different from the fuzzy extractor for a fuzzy signature scheme. The Fuzzy MP selects the keys independently from the biometric input, while the biometric inputs are actively involved in the signature generation. As a proof-of-concept of using biometric inputs, experiments on face biometrics have been conducted. The Fuzzy MP relies on a biometric template protection scheme, and a cryptographic attribute-based credential system based on the monic polynomial for signature generation. The signature and verification processes both are dependent on the pairing protocol of Elliptic Curve Cryptography.

Keywords: Digital signature · Biometrics · Fuzzy cryptography

1 Introduction

Biometrics such as face, fingerprint, and iris, are unique to every individual, such that no two person have exactly the same biometric data [15]. Due to the uniqueness of the biometric data, it can be said that each person is able to claim that their biometric templates truly belongs to them alone, which also signifies a non-repudiation property. However, one major limitation of biometrics is that once if a person's biometric template (a reference stored in the database) is compromised, the replacement is nearly impossible.

Cryptography on the other hand, is targeted towards the protection and security of a deterministic data, *i.e.*, a message. Two de-facto schemes from cryptography are encryption and digital signature, wherein the former's goal is to encrypt a data using a public key to produce a ciphertext [18], such that if an

C. T. Calafate et al. (Eds.): MONAMI 2021, LNICST 418, pp. 299–315, 2022.
https://doi.org/10.1007/978-3-030-94763-7_23

unauthorized party obtains the ciphertext, the said party is not able to obtain the message easily without the secret key (which is only known by the authorized party). As for the latter, the process is reversed, such that a digital signature is generated using a secret key and an input data [10]. The signature can be verified using the public key, which signifies the purpose of a digital signature; a non-repudiatable authentication system, since the public and secret key pairs involved in the signature can only belong to the signer.

It is well known that cryptography and biometrics are not able to integrate well due to the different natures of two fields: the former being exact, while the latter being fuzzy. Researchers have always coveted the idea of integrating biometrics into cryptography, particularly for the purpose of protecting the biometric data, *i.e.*, biometric template protection. However, it is not a simple feat to achieve. Fuzzy inputs to cryptography, especially biometric data, are considered as noise due to its stochasticism, which most cryptographic primitives lack the robustness to handle such data. However, if combined decently, both are able to complement each other to produce an adequate biometric cryptosystem, be it for biometric template protection using cryptography, or for cryptographic applications using biometric inputs.

1.1 Related Works

The usage of biometrics for cryptography, especially for cryptographic key generation is not new, as the proposal of the idea can be dated back up to 1998 [14]. However, the usage of biometrics back then was still rather unstable, considering the high Equal Error Rate (EER). In 2001, a biometric-based digital signature scheme was proposed [8] based on the RSA [18] and DSA [10] algorithm, which demonstrates their algorithm as an extension to the successful iris recognition technology at that time. The signature generation and verification hinges on the usage of the closest number in the template which is relative to the RSA prime $\phi(\mathbb{N})$. However, due to the nature of the RSA and DSA cryptosystem, a large key size is involved during the computation.

In 2005, Sahai and Waters formalized the fuzzy identity-based encryption [19], which gives rise to various fuzzy cryptosystems. This leads to the introduction of the Fuzzy Identity-Based Signature (IBS) by Yang *et al.* in 2008 [26]. The Fuzzy IBS is considered "fuzzy" in the sense that the signature is generated from a private key extracted from a biometric template. This method of extracting a private key from biometric template is similar in spirit to the Fuzzy Extractor [1].

A few years later, Wang and Kim [24] then took upon the liberty to further formalize the security notion of the Fuzzy IBS based on the discrete logarithm (DL) hard problem. This led to a definition of a Fuzzy IBS scheme with provable security [25]. To put it simply, provable security [17] is the linking of a cryptographic scheme to a hard problem (which is considered hard to brute force and is nearly impossible to inverse) via security proofs and mathematical equations, to show that the scheme is secure. The scheme is said to be tightly secure if

the probability of breaking the scheme is equals to the probability of solving the hard problem.

Different from the Fuzzy IBS schemes, Takahashi *et al.* introduced the fuzzy signature [22], which is said to be a new notion of digital signatures. The fuzzy signature adopts the generation of a fuzzy signing key instead of a fuzzy verification key. The advantage of Takahashi *et al.*'s scheme is that a helper data (HD) - additional credentials that are involved in biometric authentication, is not required to generate a signature, but the signing is solely dependent on the fuzzy input. The linear sketch was also proposed in their work for error correction purposes, which is necessary at times when biometric templates are involved.

1.2 Motivations

Based on the schemes discussed in Sect. 1.1, it is said that the fuzzy signature schemes proposed so far are able to handle biometric data as input. Such schemes utilize the fuzzy extractor notion [1], such that a cryptographic key is generated from a given biometric input.

Though this notion negates the needs of key and parameter management, it poses the risk of false accepts, where similar biometric templates of different identities are distinguished as the same identity, which in turn leads to the false acceptance of a mismatched template. In addition to that, Error Correction Codes (ECC) are usually employed to mitigate the fuzziness of two biometric instances of the same person, which further increases security issues, wherein the parity storage creates another point of attack for an adversary to obtain the original data [21].

Besides that, the fuzzy signature schemes discussed thus far have not considered biometric template protection notion, where an enrolled biometric input is altered or transformed to "protect" the raw input, such that the protected template can be stored or used instead [15]. This is particularly vital considering that biometric templates cannot be replaced once compromised.

1.3 Our Contribution

In this paper, we propose the Fuzzy MP scheme, where we make use of face biometric as a proof-of-concept for a digital signature scheme. To avoid the storage of the raw templates due to security concerns, biometric template protection scheme is considered. To fulfill this, we apply the Distance Recoverable Encryption (DRE) scheme [12] and the Index-of-Max (IoM) hashing [9]. The DRE serves as a first layer of security by encrypting the face templates, which is followed up by a second layer of protection, which would be the IoM hashing.

Besides protecting the raw face templates (face features) from being revealed, the hashed templates from the IoM hashing are also cancelable, such that the hashed templates can be revoked and renewed if compromised. Due to the performance preserving properties of the DRE and the IoM hashing, there is no glaring degradation in the verification accuracy of the encrypted vectors and the hashed codes respectively. Thus, the usage of the ECC module can be avoided, considering the security risks it poses.

The Fuzzy MP scheme works in a way such that the public and secret keys are selected independently from the biometric data. Instead, we used the face template as part of the digital signature. Though there still lies the issue of key management, our proposed method reduces the possibility of false accepts. Moreover, the biometric templates are protected by the secret key, where an adversary is not able to distinguish between the template and the key. Likewise, the secret key in turn, also protects the biometric templates.

The digital signature algorithm of the Fuzzy MP relies on a modified version of the MoniPoly Attribute-Based Credential (ABC) scheme [23], which is tightly reduced to the $q-$strong co-Diffie-Hellman (co-SDH) hard problem. Given that the scheme is reliant on a credential generation using attributes, we use the biometric data as the attributes to generate the signature. We further provide proof-of-concept experiments on the proposed scheme.

1.4 Organization

This paper is organized as follows. In Sect. 2, we first define the various tools and methods that play a part in constructing our scheme. In Sect. 3, we then define the overview of our scheme, alongside the details in each modules involved. Section 4 then illustrates the experiments and simulation conducted, alongside the security analysis of our scheme. We conclude our findings in Sect. 5.

2 Preliminaries

2.1 Digital Signature Scheme

A digital signature consists of three polynomial-time algorithms: **Key Generation**, **Sign**, and **Verify**. The first two algorithms are probabilistic. The algorithms are described as follows:

1. **Key Generation** $(1^k) \rightarrow (pk, sk)$: A pair of public and secret keys are generated based on the security parameter input 1^k. The public key pk can be transmitted openly, while the secret key sk is kept secret by the user.
2. **Sign** $(m, sk) \rightarrow \sigma$: The user uses the secret key sk to sign on a message m to generate a signature, which is denoted as σ.
3. **Verify** $(m, \sigma, pk) \rightarrow 1/0$: The verifier takes the public key pk and σ as the input to ensure that the signature is genuinely signed by the user. If the signature is authentic, the algorithm returns "1", and "0" otherwise.

2.2 Bilinear Pairing

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be groups of prime order q based on the curve E over the finite field $\mathbb{F}_p$ where $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Let g_1 be a generator of $\mathbb{G}_1$ and g_2 be a generator of $\mathbb{G}_2$. Bilinear pairing is a function which maps elements from group $\mathbb{G}_1$ and group $\mathbb{G}_2$ to group $\mathbb{G}_T$, *i.e.*, $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. The bilinear pairing function e requires the following properties:

1. Bilinearity: $e({g_1}^a, {g_2}^b) = e(g_1, g_2)^{ab}$.
2. Non-degeneracy: $e(g_1, g_2) \neq 1$
3. e is efficiently computable, which means there is an algorithm to compute $e(g_1, g_2)$ for any $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$.

2.3 $q-$strong co-Diffie-Hellman (co-SDH) Problem

Definition 1. *The $q-$strong co-Diffie-Hellman (co-SDH) Problem [2] is based on the $q-$strong Diffie-Hellman (SDH) Problem [3], with the difference that the co-SDH problem uses the Type-3 pairing [16]. It is said that a polynomial-time algorithm $\mathcal{S}$ $(t_{co-SDH}, \varepsilon_{co-SDH})$-solves the co-SDH problem for $\mathcal{S}$ running for a time of at most t_{co-SDH} and furthermore:*

$$\left|\Pr[a, b \leftarrow \mathbb{Z}_q^* : \mathcal{S}(g_1, {g_1}^x, ..., {g_1}^{x^q}, g_2, {g_2}^x, ..., {g_2}^{x^q}) = (g^{\frac{1}{x+c}}, c)]\right| \geq \varepsilon_{co-SDH}$$

We assume the co-SDH problem to be $(t_{co-SDH}, \varepsilon_{co-SDH})$-hard in $\mathbb{G}_1$ and $\mathbb{G}_2$ if $\Pr[\mathcal{S}$ solves co-SDH$] \leq \varepsilon_{co-SDH}$ for any $\mathcal{S}$ that runs in time t_{co-SDH}.

2.4 ArcFace Model

The ArcFace model [4] is a face feature extractor based on deep neural networks. To be precise, the ArcFace is a convolutional neural network (CNN) with the ResNet-50 [6] as its backbone architecture. The ArcFace loss function is given as:

$$L = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s(\cos(\theta_{y_i} + m))}}{e^{s(\cos(\theta_{y_i} + m))} + \sum_{j=1, j \neq y_i}^{n} e^{s \cos \theta_j}}$$

where θ_j represent the angles between the weights and the vectors which are distributed on a hypersphere with radius s, and m represents the margin penalty. The ArcFace model uses face images of size 112×112 as the input, such that a face feature consisting of real values are generated with a dimension of 512 for each image.

2.5 Distance Recoverable Encryption (DRE)

First introduced in [12], the Distance Recoverable Encryption (DRE) preserves the distance between two encrypted vectors without compromising the raw vector (*i.e.*, the raw biometric template). To be specific, with an encryption function E, the encryption of the DRE scheme on two vectors $\vec{\mathcal{X}}$ and $\vec{\mathcal{Y}}$ produces $E(\vec{\mathcal{X}})$ and $E(\vec{\mathcal{Y}})$ such that the distances $\text{Dist}(\vec{\mathcal{X}}, \vec{\mathcal{Y}}) = \text{Dist}(E(\vec{\mathcal{X}}), E(\vec{\mathcal{Y}}))$. The DRE scheme is defined as follows.

Given an orthogonal matrix M where the inverse and transpose of M are equal, such that $M^{-1} \cdot M = M^T \cdot M = I$, where I is the identity matrix. With a user secret key u_{sk}, the encryption is performed by utilizing the following functions:

- Generate a pseudorandom orthogonal matrix $M \in \mathbb{R}^{n \times n}$ from pseudorandom function $\text{PRF}_M(u_{sk})$.
- Generate a pseudorandom vector $\vec{v} \in \mathbb{R}^n$ from pseudorandom function $\text{PRF}_V(u_{sk})$.
- Perform a pseudorandom permutation $\pi(\vec{\mathcal{X}})$ from pseudorandom permutation $\text{PRP}(u_{sk}, \vec{\mathcal{X}})$.

The encryption of the vector is defined as $\vec{c}_{\mathcal{X}} = (\pi(\vec{\mathcal{X}}) + \vec{v})M$.

2.6 Index-of-Max (IoM) Hashing

The Index-of-Max (IoM) hashing [9] is a ranking based hashing proposed as a means of biometrics template protection. The IoM hashing is said to preserve performance in terms of accuracy, where the distance of the "hashed codes" which are generated in integers, are rather similar to the distance of the same raw templates. In short, the IoM hashing is mainly used for the purpose of cancelable biometrics. The IoM hashing scheme, particularly the Gaussian Random Projection (GRP) variant which we will be using, is described as follows.

1. For a feature vector $\mathbf{x} \in \mathbb{R}^d$ and user secret key u_{sk}, set the random seed using u_{sk}, and generate a random Gaussian matrix with $\mathbf{W}$, such that $\mathbf{W} \in \mathbb{R}^{d \times q \times m}$.
2. Multiply $\mathbf{x}$ with $\mathbf{W}$ to obtain $z \in \mathbb{R}^{1 \times q \times m}$.
3. For each z_i such that $i = 1, ..., m$, record the maximum index in the q-th axis to obtain the hash codes $q \in \mathbb{R}^m$.

2.7 MoniPoly Commitment Scheme

The MoniPoly Attribute-Based Credential (ABC) scheme [23] is proposed for the privacy of multiple attributes in mind. In this section, we define the MoniPoly commitment scheme[1], which gives rise to the MoniPoly SDH-CL signature and the MoniPoly ABC scheme, which we will define later on.

[1] We did not include the **OpenDifference** and **VerifyDifference** algorithms in this definition, as the **Intersection** algorithms are sufficient for our case.

1. **Setup** $(1^k, n) \rightarrow (pk, sk)$. Construct cyclic groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ based on prime order p and bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Then, select random generators $g_1 \overset{R}{\in} \mathbb{G}_1, g_2 \overset{R}{\in} \mathbb{G}_2$ and generate random number $x \overset{R}{\in} \mathbb{Z}_p^*$. For number of attributes n, compute values $\{a_0, a_1, ..., a_n\} = \{g_1, {g_1}^x, ..., {g_1}^{x^n}\}$ and $\{X_0, X_1, ..., X_n\} = \{g_2, {g_2}^x, ..., {g_2}^{x^n}\}$. The public key and secret key[2] are: $pk = (e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \{a_i, X_i\}_{0 \leq i \leq n})$, $sk = (x)$.
2. **Commit** $(pk, A, o) \rightarrow (C)$. For a set of message $A = \{m_1, ..., m_{n-1}\} \in \mathbb{Z}_p^*$ and a random open value $o \overset{R}{\in} \mathbb{Z}_p^*$, generate a commitment value C. Given $\{\mathsf{m}_j\} = \mathsf{MPEncode}[3](A \cup \{o\})$, the value C is given as:

$$C = {a_0}^{(x+o)\prod_{j=1}^{n-1}(x+m_j)} = \prod_{j=0}^{n} {a_j}^{\mathsf{m}_j}$$

3. **Open** $(pk, C, A, o) \rightarrow 1/0$. The prover (person who committed earlier) reveals message A' and opening value o'. Compute the value $C' = \prod_{j=0}^{n} {a_j}^{\mathsf{m}'_j}$, where $\mathsf{m}'_j = \mathsf{MPEncode}(A' \cup \{o'\})$. If the value of $C' = C$, return 1 to accept; else return 0 to reject.
4. **OpenIntersection** $(pk, C, A, o, (A', \ell)) \rightarrow (I, W)/\perp$. For a threshold ℓ and given $\{\mathsf{w}_j\} = \mathsf{MPEncode}((A - I) \cup o)$, return $\perp$ if the ℓ is not met; otherwise return an intersection set $I = A' \cap A$ and witness W if $|A' \cap A| \geq \ell$ holds such that:

$$\begin{aligned} C &= \left(\prod_{j=0}^{n-\ell} {a_j}^{\mathsf{w}_j}\right)^{\prod_{\mathsf{m}_j \in I} x+\mathsf{m}_j} \\ &= W^{\prod_{\mathsf{m}_j \in I} x+\mathsf{m}_j} \end{aligned}$$

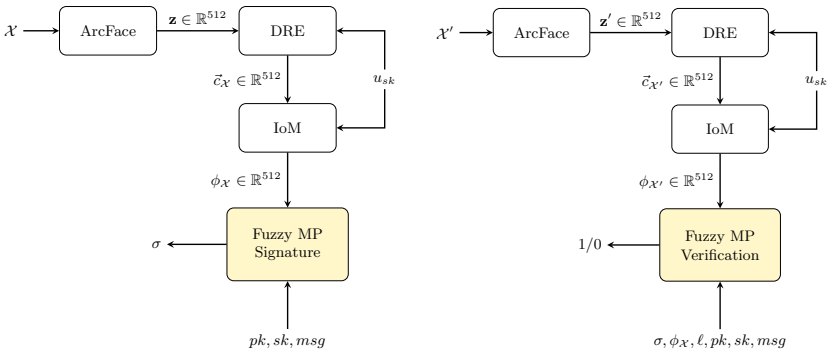
5. **VerifyIntersection** $(pk, C, (I, W), (A', \tilde{\ell})) \rightarrow 1/0$. Given $\{\mathsf{i}_j\} = \mathsf{MPEncode}(I)$, $\{\mathsf{m}_{1,j}\} = \mathsf{MPEncode}(A')$, and $\{\mathsf{m}_{2,j}\} = \mathsf{MPEncode}(A' - I)$, return 1 iff the following check equation as the following is met.

$$e\left(C \prod_{j=0}^{|A'|} {a_j}^{\mathsf{m}_{1,j}}, X_0\right) = e\left(W \prod_{j=0}^{|A'|-\ell} {a_j}^{\mathsf{m}_{2,j}}, \prod_{j=0}^{\ell} {X_j}^{\mathsf{i}_j}\right)$$

[2] sk can be discarded if n is fixed.
[3] $\mathsf{MPEncode}$ is the mapping of $\mathbb{Z}_p^n$ to $\mathbb{Z}_p^{n+1}$.

3 Proposed Scheme

3.1 Overview of our Proposed Scheme



(a) Signing Process (b) Verification Process

Fig. 1. Overview of fuzzy MP signature scheme

The overview of our scheme is presented in Fig. 1. To begin with the signing process, the raw biometric inputs $\mathcal{X}$ are extracted using the ArcFace, which result in (face) biometric features $\mathbf{z}$ with a dimension of 512. The feature is then encrypted using the DRE to obtain the encrypted vector $\vec{c}_{\mathcal{X}}$, with the help of a (user) secret key u_{sk}. With the same u_{sk}[4], the hash codes of the encrypted vectors $\phi_{\mathcal{X}}$, are generated using the IoM hashing. Given the public and secret keys (pk, sk) with the hashed codes $\phi_{\mathcal{X}}$ as inputs, the digital signature σ is generated.

Similar to the signing process, the verification of the generated digital signature σ is initiated with the feature extraction, encryption, and hashing process. This is used to show that the prover (*i.e.*, the user who wants prove that σ is his signature) is the authentic signer using his biometric template. By the comparison of an honest prover's extracted template $\phi_{\mathcal{X}'}$ with $\phi_{\mathcal{X}}$, the signature will be proven authentic if a certain threshold set by the verifier ℓ is met. We now further elaborate on each modules shown in Fig. 1.

3.2 ArcFace Feature Extractor

For face feature extraction, which is to be used as raw face template after enrollment or as query during signature verification, we used a pretrained ArcFace network. The face images were resized to 112×112 pixels prior to feature extraction. For each of the image, an output face feature vector $\mathbf{z} \in \mathbb{R}^{512}$ is generated.

[4] The u_{sk} may be selected by the user, such as a user-specific password to be kept by the user. Though it is shown that the u_{sk} is the same one used for the DRE, the u_{sk} may be a different user secret key too. This will be further elaborated later on.

3.3 Distance Recoverable Encryption (DRE)

Given the output of the ArcFace feature extractor $\mathbf{z} \in \mathbb{R}^{512}$ and a user secret key $u_{sk} \in \mathbb{Z}_q^*$ as inputs, an encrypted feature vector $\vec{c}_{\mathcal{X}} \in \mathbb{R}^{512}$ is generated. As the randomness of the vectors and permutation are greatly affected by the state of randomness, the u_{sk} plays a role as a seed to generate the random vectors and perform the random permutation. With the same u_{sk}, a query biometric template would be permuted and operated on with the same random vectors. Thus, this ensures the distance preservation of the generated vectors.

3.4 Index-of-Max (IoM) Hashing

In consideration that the IoM hashing utilizes the generation of a random Gaussian matrix, the mean and covariance values (μ, σ) are crucial. Apart from that, the randomness of the matrix are also reliant on the random seed, similar to the DRE. Thus, a user specific key can be defined for this purpose. Though it is desirable to use the same u_{sk} as the one in the DRE to avoid key management issues, it is suggested to use a different key instead for better security. Regardless, an appropriate secure key is sufficient to perform the hashing, since the generated hash codes $(\phi_{\mathcal{X}})$ are considered to be cancellable templates.

3.5 Fuzzy MP Signature

The definition of the Fuzzy MP signature scheme is two-fold. The main essence of the signature is based on the MoniPoly variant of the SDH-CL signature [20], where a message is signed on a set of attributes. However, since the MoniPoly SDH-CL is unable to perform verification for fuzzy inputs, we make use of the show proofs defined in the MoniPoly ABC scheme, particularly the ANY/OR proof for the verification.

Apart from handling fuzzy inputs, the verification used for the proposed Fuzzy MP scheme enables the prover (in this case, the signer) to achieve anonymity during the signature verification. To be specific, the intersection of the signing and proving templates are not known to the verifier. A more detailed definition of our scheme is shown in Algorithms 1 and 2, which corresponds to Fig. 1. It is noted that both algorithms demonstrate the honest signature and verification process, where the signer and prover are the same person wherein both templates are within a set threshold ℓ, (*i.e.*, $u_{sk} = u'_{sk}, sk = sk'$, and $(\mathcal{X} \approx \mathcal{X}' \big| |\mathcal{X} \cap \mathcal{X}'| \geq \ell)$).

It is noted that if a message msg is to be signed, it is appended to $\phi_{\mathcal{X}}$ such that an extra public key value is generated for each element in $\phi_{\mathcal{X}}$, including the message. This msg value corresponds to an extra attribute in the scheme. The same procedure is repeated during the signature verification, where msg is appended to the prover's feature vectors $\phi_{\mathcal{X}'}$.

[5] At this point, it can be said that the pairing here is already sufficient to prove the authenticity of the signature, considering that $\phi_{\mathcal{X}}$ is stored. However, an extra pairing process is used during the verification algorithm to prevent replay attacks.

Algorithm 1: Fuzzy MP Signing Algorithm

Input: pk, sk
Output: Signature σ
Data: Signer template $\mathcal{X}$, user secret key u_{sk}
Function `ExtractFeatures`$(\mathcal{X}, u_{sk})$**:**
  ArcFace$(\mathcal{X}) \rightarrow \mathbf{z}$
  DRE$(\mathbf{z}, u_{sk}) \rightarrow \vec{c}_{\mathcal{X}}$
  IoM$(\vec{c}_{\mathcal{X}}, u_{sk}) \rightarrow \phi_{\mathcal{X}}$
  return $\phi_{\mathcal{X}}$

Function `KeyGen`(1^k)**:**
  $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$
  $g_1, b, c \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2, X = {g_2}^{x'}$, $x, x', op \overset{R}{\in} \mathbb{Z}_p^*$
  $\{a_0, a_1, ..., a_n\} = \{g_1, {g_1}^x, ..., {g_1}^{x^n}\}$, $\{X_0, X_1, ..., X_n\} = \{g_2, {g_2}^x, ..., {g_2}^{x^n}\}$
  $pk = (e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, b, c, X, \{a_i, X_i\}_{0 \leq i \leq n})$, $sk = (x, x', op)$
  return pk, sk

Function `Sign`$(pk, sk, \phi_{\mathcal{X}})$**:**
  $s_1, s_2, t \overset{R}{\in} \mathbb{Z}_p^*$, set $\phi_{\mathcal{X}} = (\phi_{\mathcal{X}} \cup \{op\})$
  MPEncode$(\phi_{\mathcal{X}}) \rightarrow \Phi_{\mathcal{X}} = (\Phi_0, \Phi_1, ..., \Phi_n)$
  $M = \prod_{j=0}^{n} {a_j}^{\Phi_j} b^{s_1}$, $s = s_1 + s_2$, $v = (Mb^{s_2}c)^{1/(x'+t)}$
  if $e(v, X) = e\left(Mb^{s_2}cv^{-t}, X_0\right)$ **then**
    return[5] $\sigma = (t, s, v)$
  end

4 Experiments and Analysis

4.1 Experiment Setup

The implementation and experiments for our scheme are conducted using Python. The main reason for doing so is due to the procedure of extracting face features from the biometric inputs using the ArcFace feature extractor. For the machine, Linux Ubuntu 16.04 is used for the OS, which runs on an NVIDIA RTX 2080 Ti D6 GPU and an Intel i9-9900K CPU with 3.60 GHz and 8 cores. The pre-trained ArcFace feature extractor is applied using TensorFlow v1.14.0 and the GPU.

As for the dataset, we utilize an unconstrained face dataset, namely the Learning From Wild (LFW) [7]. To be precise, we conduct the signing and verification using the standard LFW evaluation protocol which consists of 3000 pairs of matched and non-matched identities each.

Since the signature and verification algorithms of the Fuzzy MP scheme are dependent on bilinear pairing, we make use of the MIRACL [13] library, a pairing-friendly cryptography library, particulary the Python variation. We utilize the

Algorithm 2: Fuzzy MP Verification Algorithm

Input: $pk, sk', \phi_{\mathcal{X}}, \sigma$
Output: 1/0
Data: Prover template $\mathcal{X}'$, user secret key u'_{sk}, threshold ℓ
Function `ExtractFeatures`$(\mathcal{X}, u'_{sk})$**:**
  ArcFace$(\mathcal{X}') \rightarrow \mathbf{z}'$
  DRE$(\mathbf{z}', u'_{sk}) \rightarrow \vec{c}_{\mathcal{X}'}$
  IoM$(\vec{c}_{\mathcal{X}'}, u'_{sk}) \rightarrow \phi_{\mathcal{X}'}$
  return $\phi_{\mathcal{X}'}$

Function `Verify`$(pk, sk', \phi_{\mathcal{X}}, \phi_{\mathcal{X}'}, \sigma, \ell)$**:**
  Prover:
  Compute $I = |\phi_{\mathcal{X}'} \cap \phi_{\mathcal{X}}| \geq \ell$, where $\phi_{\mathcal{X}'} = (\phi_{\mathcal{X}'} \cup \{op\})$
  $\{\mathsf{w}'_j\}_{0 \leq j \leq n-\ell} = \mathsf{MPEncode}(\phi_{\mathcal{X}} - I), \{\mathsf{m}_{2,j}\}_{0 \leq j \leq k-\ell} = \mathsf{MPEncode}(\phi_{\mathcal{X}'} - I)$
  $W = \left(\prod_{j=0}^{n-\ell} a_j^{\mathsf{w}'_j}\right), W' = \left(\prod_{j=0}^{k-\ell} a_j^{\mathsf{m}_{2,j}}\right)$
  Verifier:
  $\{\mathsf{m}_{1,j}\}_{0 \leq j \leq k} = \mathsf{MPEncode}(\phi_{\mathcal{X}'}), \{\iota_j\}_{0 \leq j \leq \ell} = \mathsf{MPEncode}(I)$
  if $e\left(W'W, \prod_{j=0}^{\ell} X_j^{\iota_j}\right) e\left(\prod_{j=0}^{k} a_j^{-\mathsf{m}_{1,j}} b^s c v^{-t}, X_0\right) = e(v, X)$ **then**
    return 1
  end
  else
    return 0
  end

BN-462 elliptic curve for that purpose. The encryption, hashing, and signature/verification are carried out using the CPU. It is noted that the DRE and IoM hashing are also run using the CPU.

For the performance evaluation, we apply the Equal Error Rate (EER) and the Receiver Operating Characteristic (ROC) curve as a performance measurement for the encryption and hashing (*i.e.*, DRE and IoM hashing) procedures. To that end, we employ the cosine distance as a similarity metric. On the other hand, we set the threshold for the signature and verification algorithms using the Hamming distance between the signer and prover's feature vectors in computing the EER for the signature verification.

4.2 Performance Measurement

For the performance comparison described in this section, the following procedure is referred to.

$$\text{ArcFace}(\mathbf{z}) \Rightarrow \text{DRE}(\vec{c}_{\mathcal{X}}) \Rightarrow \text{IoM}(\phi_{\mathcal{X}}) \Rightarrow \text{Fuzzy MP}(\sigma)$$

Parameter Analysis. Table 1 outlines the Equal Error Rate (EER) percentage for different q parameter values. As the face features $\mathbf{z}$ and the encrypted vectors $\vec{c}_{\mathcal{X}}$ are not affected by any parameter values, only the results of $\phi_{\mathcal{X}}$ and σ are tabulated in Table 1. In consideration that the encrypted vector $\vec{c}_{\mathcal{X}}$ has a dimension of 512 due to the ArcFace feature extractor, the parameter m for the IoM hashing is fixed to 512 as well.

Table 1. EER for different q parameters for $m = 512$

EER (%)	q				
	8	16	32	64	128
IoM ($\phi_{\mathcal{X}}$)	**4.23**	5.00	4.57	**4.23**	5.47
Fuzzy MP (σ)	**3.33**	3.50	**3.33**	3.53	3.40

In Table 1, notice that the EER for the Fuzzy MP signature is rather low in comparison to that of the IoM hashing. This is attributed to the usage of Hamming distance, in consideration that the Fuzzy MP signature relies on element-wise comparison between $\phi_{\mathcal{X}}$ and $\phi_{\mathcal{X}'}$ to set the threshold.

Performance of ArcFace, DRE, and IoM Hashing. Figure 2 illustrates the Receiver Operating Characteristic (ROC) curve for the ArcFace, DRE, and IoM hashing feature vectors, particularly for the one where the EER value of $\phi_{\mathcal{X}}$ is 4.23. To be precise, the ArcFace curve signifies $\mathbf{z}$, the DRE curve signifies $\vec{c}_{\mathcal{X}}$, and the IoM curve signifies $\phi_{\mathcal{X}}$. It is noted that the same u_{sk} is used for performing both the DRE and IoM hashing.

Based on Fig. 2, it can be observed that both the curves for the ArcFace extracted features and the DRE encrypted feature vectors are almost similar. This shows that the DRE is able to preserve the distance performance, which is particularly important in distinguishing genuine and impostor features. As for the IoM hashing, the performance is slightly degraded. The EER based on the curve shown in Fig. 2 is presented in Table 2.

Table 2. EER of feature vectors

Feature	ArcFace	DRE	IoM
EER (%)	3.50	3.10	4.23

Performance of Fuzzy MP Signature. The overall timing to run the algorithms as well as the asymptotic complexity are tabulated in Table 3. For the time taken, an average for a total of 100 rounds of each algorithm are taken for the simulation.

Table 3. Simulation and complexity for fuzzy MP signature

Algorithm	Asymptotic complexity	Time (ms)
Key Gen	$\mathcal{O}(n)$	109,755
Sign	$\mathcal{O}(2n)$	39,777
Verify	$\mathcal{O}(n + 2k)$	28,461

Note: $k = |\phi_{\mathcal{X}'}| \leq n = |\phi_{\mathcal{X}}|$

It can be deduced that the values shown in Table 3 is not optimal, as the simulation is carried out using Python. Some possible ways of improving the timing is by implementing a parallelized version of the algorithms, and also by utilizing a GPU for the computations.

4.3 Unlinkability Analysis

In [23], a detailed unlinkability analysis is shown such that the unlinkability of the values in $\phi_{\mathcal{X}}$ with the generated σ implies the anonymity of $\phi_{\mathcal{X}}$. Therefore, we chose not to emphasize further the unlinkability for the Fuzzy MP signature with $\phi_{\mathcal{X}}$, since it is already proven.

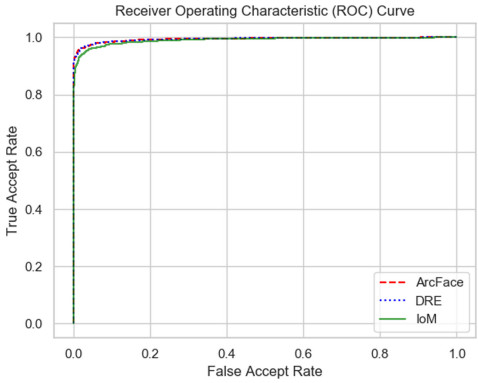
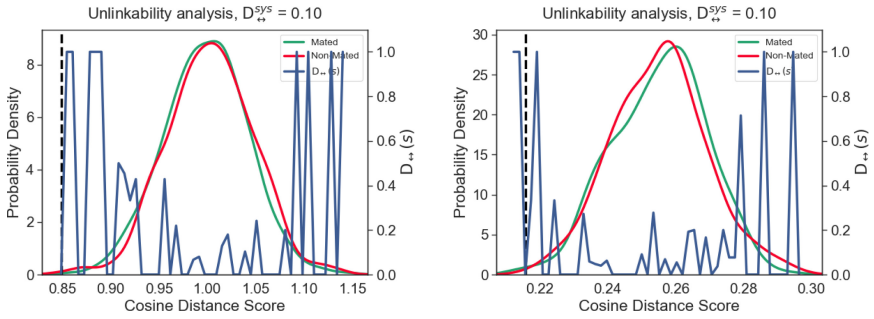


Fig. 2. ROC curve of feature vectors

Instead, in consideration that there is also a lack of unlinkability analysis for the DRE [12], we evaluate the unlinkability of the feature vectors that are generated before the Fuzzy MP signing process. The unlinkability of the feature vectors are equally important to prevent cross-matching attacks, where an adversary is able to obtain the raw templates through compromising the matching scores, taking into account the storage of $\phi_{\mathcal{X}}$.



(a) DRE ($\vec{c}_{\mathcal{X}}$) (b) IoM ($\phi_{\mathcal{X}}$)

Fig. 3. Unlinkability analysis

In order to do so, we conduct the evaluation based on the protocol by [5]. The mated sample requires the generation of protected templates with *different seeds* for the *same instances*, while the non-mated sample requires the generation of protected templates with *different seeds* for *different instances*. We carry out the protocol using 10 random seed instances. The results of the unlinkability analysis are disclosed in Fig. 3.

Following the proposed unlinkability protocol [5], the local and global measures are given as $D_{\leftrightarrow}(s) \in [0, 1]$ and $D_{\leftrightarrow}^{sys} \in [0, 1]$ respectively. For the local measure $D_{\leftrightarrow}(s)$, the score $s = 0$ signifies that an adversary is not able to decide if a protected template is from the same person, and 1 for vice versa. As for the global measure $D_{\leftrightarrow}^{sys}$, the score 0 signifies full unlinkability for the mated subjects, and 1 for vice versa.

In Fig. 3, it is shown that both $\vec{c}_{\mathcal{X}}$ and $\phi_{\mathcal{X}}$ has a global measure of 0.10, which shows a sufficiently good score of unlinkability. However, there is still room for improvement in the unlinkability analysis, especially for $\phi_{\mathcal{X}}$. One possible improvement is to replace the IoM hashing with the Softmax-Out Transformation Permutation Network (SOTPN) [11], which was dubbed as the neural network version of the URP-IoM [9].

4.4 Security Attacks

In this section, we examine the durability of our scheme in terms of security. We begin by formalizing an adversary $\mathcal{A}$ that intends to attack the scheme from multiple angles. For each of the attacks defined, we first state $\mathcal{A}$'s goals, and how our scheme is able to handle such attacks.

Collision Attack. Also known as the brute force attack in this case, $\mathcal{A}$'s goal in conducting collision attacks is to be able to forge the signature using numerous repetition of biometric templates and user keys, to cause a collision of signature (*i.e.*, a forger) with one of the eventual combinations. However, the likeliness of

a collision between $(\phi_{\mathcal{X}}, op)$ and $(\phi_{\mathcal{X}'}, op')$ is unlikely if he does not know the value of op [23]. This means that $\mathcal{A}$ is not able to forge the signature σ using $(\phi_{\mathcal{X}'}, op')$ via collision unless he is able to solve the co-SDH hard problem.

False Accept Attack. Also known as the dictionary attack, the false accept attack is similar to the collision attack, however to forge a signature with a predetermined set of "close enough" biometric templates. This leads to a false accept of the said biometric templates. As mentioned in the collision attacks, the signature cannot be forged if $\mathcal{A}$ does not have the value of op, as it is assumed that $\mathcal{A}$ has to solve the co-SDH problem without knowing the value of op.

Replay Attack. It is noted that the first pairing in the signing algorithm to authenticate σ is susceptible to replay attacks during the first verification process. In short, $\mathcal{A}$ is able to forge the σ via replay attacks during the pairing process. This leads to the usage of a user's biometric template for the verification process, as the one in Algorithm 2. With the freshness of $\phi_{\mathcal{X}'}$ during each verification process, $\mathcal{A}$ is not able to conduct a replay attack, since the returned values will be different each time.

5 Conclusion

In this paper, we presented a fuzzy digital signature scheme, namely the Fuzzy MP. Though the Fuzzy MP still requires key management, the keys used for our scheme are selected independently from the biometric templates, which reduces the risk of false accepts. Through the experiments and analysis conducted thus far using the LFW face dataset, it can be deduced that the Fuzzy MP is able to achieve a satisfactory performance in terms of the EER, though there is still room for improvement in the timing and complexity to run the algorithm.

Acknowledgements. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NO. NRF-2019R1A2C1003306). The authors would also like to thank Syh-Yuan Tan and Thomas Groß for their helpful advice during the implementation of the MoniPoly scheme.

References

1. Boyen, X.: Reusable cryptographic fuzzy extractors. In: Proceedings of the 11th ACM Conference on Computer and Communications Security, pp. 82–91 (2004)
2. Chatterjee, S., Menezes, A.: On cryptographic protocols employing asymmetric pairings-the role of ψ revisited. Discrete Appl. Math. **159**(13), 1311–1322 (2011)
3. Cheon, J.H.: Security analysis of the strong Diffie-Hellman problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 1–11. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_1

4. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: ArcFace: additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition – CVPR 2019, pp. 4690–4699 (2019)
5. Gomez-Barrero, M., Galbally, J., Rathgeb, C., Busch, C.: General framework to evaluate unlinkability in biometric template protection systems. IEEE Trans. Inf. Forensics Secur. **3**(6), 1406–1420 (2018)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition – CVPR 2016, pp. 770–778 (2016)
7. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. In: Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition (2008)
8. Janbandhu, P.K., Siyal, M.Y.: Novel biometric digital signatures for Internet based applications. Inf. Manage. Comput. Secur. (2001)
9. Jin, Z., Hwang, J.Y., Lai, Y.L., Kim, S., Teoh, A.B.J.: Ranking-based locality sensitive hashing-enabled cancelable biometrics: index-of-max hashing. IEEE Trans. Inf. Forensics Secur. **13**(2), 393–407 (2017)
10. Kerry, C.F., Director, C.R.: FIPS PUB 186-4 federal information processing standards publication digital signature standard (DSS). FIPS Publication (2013)
11. Lee, H., Low, C.Y., Teoh, A.B.J.: SoftmaxOut transformation-permutation network for facial template protection. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 7558–7565. IEEE (2021)
12. Loh, J.C., et al.: PBio: Enabling Cross-organizational Biometric Authentication Service through Secure Sharing of Biometric Templates. Cryptology ePrint Archive: Report 2020/1381 (2020). ia.cr/2020/1381
13. MIRACL: MIRACL Core (2021). https://github.com/miracl/core/tree/master/python
14. Nichols, R.K.: ICSA Guide to Cryptography. McGraw-Hill Professional, New York (1998)
15. Patel, V.M., Ratha, N.K., Chellappa, R.: Cancelable biometrics: a review. IEEE Sig. Process. Mag. **32**(5), 54–65 (2015)
16. Pereira, G.C., Simplício, M.A., Naehrig, M., Barreto, P.S.: A family of implementation-friendly BN elliptic curves. J. Syst. Softw. **84**(8), 1319–1326 (2011)
17. Pointcheval, D., Stern, J.: Security proofs for signature schemes. In: Maurer, U. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 387–398. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68339-9_33
18. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
19. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_27
20. Schäge, S.: Tight proofs for signature schemes without random oracles. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 189–206. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_12
21. Stoianov, A.: Security of error correcting code for biometric encryption. In: 2010 Eighth Annual International Conference on Privacy Security and Trust (PST), pp. 231–235 (2010)
22. Takahashi, K., Matsuda, T., Murakami, T., Hanaoka, G., Nishigaki, M.: A signature scheme with a fuzzy private key. In: Malkin, T., Kolesnikov, V., Lewko, A.B., Polychronakis, M. (eds.) ACNS 2015. LNCS, vol. 9092, pp. 105–126. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-28166-7_6

23. Tan, S.-Y., Groß, T.: MoniPoly—an expressive q-SDH-Based anonymous attribute-based credential system. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020. LNCS, vol. 12493, pp. 498–526. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64840-4_17
24. Wang, C.J., Kim, J.H.: Two constructions of fuzzy identity based signature. In: 2009 2nd International Conference on Biomedical Engineering and Informatics, pp. 1–5. IEEE (2009)
25. Wang, C.: A provable secure fuzzy identity based signature scheme. Sci. China Inf. Sci. **55**(9), 2139–2148 (2012)
26. Yang, P., Cao, Z., Dong, X.: Fuzzy Identity Based Signature. IACR Cryptology ePrint Archive (2008)