A Generic Construction of Fuzzy Signature

Jie Song[1] and Yunhua Wen[1,2](✉)

[1] School of Computer Science and Technology, Donghua University, Shanghai 200240, China
2202559@mail.dhu.edu.cn, yhwen@dhu.edu.cn
[2] State Key Laboratory of Information Security (Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093)

Abstract. *Fuzzy signature* is a signature scheme in which the signing key is no longer uniformly generated nor precisely reproducible but a fuzzy string with enough entropy such as biometric information. In this paper, we give a variant definition of fuzzy signature and propose a generic construction of fuzzy signature which uses a fuzzy extractor and a signature scheme with simple key generation process as building blocks. Meanwhile, we give two instantiations of our generic construction. The first instantiation results in a fuzzy signature scheme which is secure under the computational Diffie-Hellman (CDH) assumption over bilinear groups in the standard model. The second instantiation results in a fuzzy signature that is secure in the random oracle model under the worst-case hardness of the $\widetilde{O}(n^{1.5})$-SIVP problem in general lattices. Moreover, compared with previous work, our fuzzy signatures have weaker requirements for the fuzzy signing key, which makes our fuzzy signatures more practical.

Keywords: Digital signature · Fuzzy extractor· Fuzzy Signature

1 Introduction

The security of cryptographic primitives depends on the security of private key. The private key generally needs to be uniformly distributed and carefully kept. Once the private key is leaked, there is no security at all. Private key is hard to remember, and securely keeping it puts a burden on the user. Since biometric information is inherent and unique, one of the promising approaches to fundamentally solve this problem is to use biometric information (e.g., fingerprint [11], face and iris [4]) as a cryptographic private key. However, since biometric data is not uniformly distributed and fluctuates each time when it is captured, it cannot be used directly as a cryptographic key.

A lot of works has been devoted to researching the application of biometric information in cryptography. For example, fuzzy extractor [6], fuzzy signature [14] and biometric-based remote user authentication [1, 2, 7].

In this paper, we will focus on the study of fuzzy signature [13]. *Fuzzy signature* is a digital signature in which the signing key need not to be uniformly distributed or accurate reproducible. We call such signing key as fuzzy signing

key. With fuzzy signature people can use his/her biometric characteristics (such as retina, iris, face and fingerprint) as the fuzzy signing key to sign messages. In fuzzy signature, the key generation algorithm $\mathsf{KeyGen_F}$ takes the fuzzy signing key sk (a sample of a user's biometric characteristic) as input and outputs a verification key vk. The signing algorithm $\mathsf{Sign_F}$ takes the fuzzy signing key sk' (another sample of the same biometric characteristic) and a message m as input, and outputs a signature σ. The verification algorithm $\mathsf{Verify_F}$ on input vk, m, σ outputs 0/1 meaning σ is invalid or valid. If the two fuzzy signing keys sk' and sk are close enough, the signature σ will be verified as valid by the verification key vk, where vk is generated by sk.

Takahashi et al. [13] gave the formal definition of fuzzy signature. They gave a generic construction of fuzzy signature based on a signature scheme with certain homomorphic properties regarding keys and signatures, and a tool called linear sketch. They showed a concrete instantiation of their generic construction based on the Waters signature scheme [15]. However, the resulting fuzzy signature has a weakness that the fuzzy signing key sk needs to be uniformly distributed. In another word, if a user wants to use his/her biometric characteristic such as fingerprint as the fuzzy signing key, then the sample of his/her fingerprint is assumed to be uniformly distributed. It seems impossible that samples from biometric characteristics follow the uniform distribution. In order to solve this problem, in paper [9], Matsuda et al. gave a new construction of fuzzy signature by relaxing the requirements of the building blocks in [13]. By instantiation, they got a fuzzy signature which does not need the fuzzy signing key uniform anymore, but the resulting fuzzy signature is secure only in the random oracle model.

Takahashi et al. also defined the security model of fuzzy signature [13]. Recall that the key generation algorithm of fuzzy signature $\mathsf{KeyGen_F}$ on input a fuzzy signing key sk outputs a verification key vk and the signing algorithm $\mathsf{Sign_F}$ on input the fuzzy signing key sk' and a message m outputs a signature σ, where sk and sk' are two different samples of the same biometric characteristic. In their security model, it is required that the error distribution e between the two different samples sk and sk' is independent of the biometric characteristic. This requirement makes the fuzzy signature far from practical application because it seems impossible that the error distribution between two different samples is independent of the biometric characteristic. A natural question arises:

Is it possible to design a fuzzy signature which has a more practical requirement for the fuzzy signing key?

1.1 Our Contributions

We answer the above question in the affirmative. Our contributions can be listed as follows:

- We give a formal definition of fuzzy signature which is a little different from that by Takahashi et al. [14]. The difference between our fuzzy signature and Takahashi et al.'s is that when generating a signature of a message m, our

signature algorithm needs both the fuzzy signing key sk and the verification key vk as input while Takahashi et al.'s does not need vk.

- We define a new security model of fuzzy signature which is called $\mathfrak{m}$-existentially unforgeable under chosen message attack ($\mathfrak{m}$-EUF-CMA) security. In this security model, we only require that the fuzzy signing key sk has entropy larger than $\mathfrak{m}$ and the error distribution between different samples sk and sk' can be arbitrary depend on the biometric characteristic.
- We provide a generic construction of fuzzy signature based on a fuzzy extractor (FE) and a signature scheme (SIG) with a simple key generation process. The simple key generation process says that the key generation algorithm first picks a secret key uniformly at random from the secret key space, then computes the corresponding verification key deterministically from the secret key.
- We give two instantiations of our generic construction.
  - When instantiating SIG with the Waters signature scheme [15] and FE constructed in [6], we obtain a fuzzy signature scheme which is secure in the standard model under the computational Diffie-Hellman (CDH) assumption in bilinear groups and only assumes that the fuzzy signing key has enough entropy.
  - When instantiating SIG with the lattice-based signature constructed in [8] and FE constructed in [6], we obtain a fuzzy signature scheme which is secure in the random oracle model based on the worst-case hardness of the $\widetilde{O}(n^{1.5})$-SIVP problem in general lattices and only assumes that the fuzzy signing key has enough entropy.

Our fuzzy signatures have a weaker requirement for the fuzzy signing key, which makes the fuzzy signature more practical. In Table 1, we compare our work with previous fuzzy signature schemes.

Table 1. Comparison with some known fuzzy signature schemes. Let SK be the distribution of the fuzzy signing key and $|\mathcal{SK}|$ be the size of key space $\mathcal{SK}$. "Entropy Requirement" asks what is the entropy requirement for the fuzzy signing key. "Correlation" asks the relationship of error distribution e and the biometric characteristic W, where e is the error distribution between the fuzzy signing key sk and sk' (two samples of the same biometric characteristic W). "Assumption" asks which assumption the fuzzy signature is based on. "Standard Model" asks whether the fuzzy signature is secure in the standard model.

Fuzzy Signature Schemes	Entropy Requirement	Correlation	Assumption	Stantard Model
TMMHN [13]	$H_\infty(SK) = log\|\mathcal{SK}\|$	independent	CDH	Yes
MTMH [9]	$H_\infty(SK) \geq \mathfrak{m}$	independent	DL	No
Our first instantiation	$H_\infty(SK) \geq \mathfrak{m}$	arbitrary	CDH	Yes
Our second instantiation	$H_\infty(SK) \geq \mathfrak{m}$	arbitrary	$\widetilde{O}(n^{1.5})$-SIVP	No

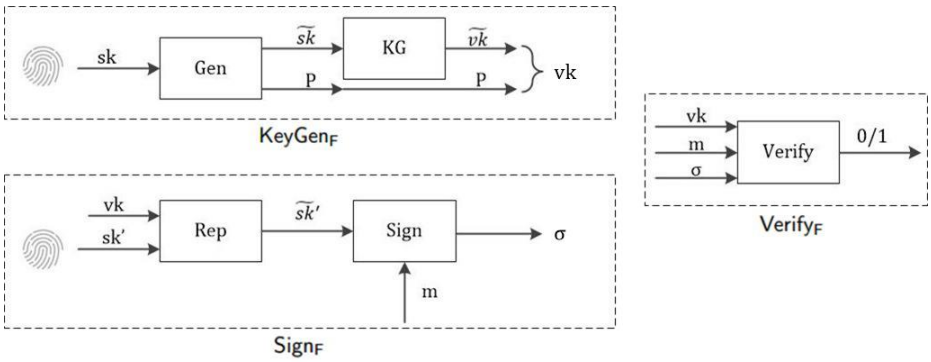

Fig. 1. The Construction of Fuzzy Signature

1.2 Our Approach

Our construction makes use of a fuzzy extractor and a signature scheme with a simple key generation process. Recall that a fuzzy extractor consists of two efficient algorithms $(\mathsf{Gen}, \mathsf{Rep})$. The generation algorithm Gen on input a sample w of a biometric characteristic $\mathcal{W}$ (such as retina, iris, face and fingerprint) outputs a public helper string P together with an extracted string R. The reproduction algorithm Rep on input w' and the public helper string P will reproduce R if w' is close enough to w. The security of fuzzy extractor guarantees that R is (pseudo-)random if $\mathcal{W}$ has enough entropy. A signature scheme with a simple key generation process says that there exists a deterministic PPT algorithm KG such that the key generation algorithm KeyGen can be written as follows:

$$\mathsf{KeyGen} : [\ sk \leftarrow_{\$} \mathcal{SK}; vk \leftarrow \mathsf{KG}(sk); \mathrm{Return}(vk, sk).\ ],$$

where $\mathcal{SK}$ is the secret key space of the signature.

Our generic construction of fuzzy signature is shown in Fig. 1 in which we omit the public parameters. More precisely,

- The key generation algorithm $\mathsf{KeyGen_F}$ on input the fuzzy signing key sk which is a sample of a biometric characteristic (e.g., fingerprint), outputs the verification key vk. It proceeds as follow, the fuzzy signing key sk is fed to the generation algorithm Gen of fuzzy extractor. The generation algorithm Gen outputs a public helper string P and a uniformly random string which will be served as the signing key $\widetilde{sk}$ of the underlying signature scheme (not fuzzy). The deterministic PPT algorithm KG on input $\widetilde{sk}$ outputs $\widetilde{vk}$. The verification key is $vk = (\widetilde{vk}, P)$.
- The signature algorithm $\mathsf{Sign_F}$ on input a fuzzy signing key sk' and the verification key vk will invoke the reproduction algorithm Rep to reproduce the signing key $\widetilde{sk}$ of the underlying signature scheme(not fuzzy) with the help of P, then use $\widetilde{sk}$ to sign the message m.

– The verification algorithm $\mathsf{Verify_F}$ on input (vk, m, σ), parses $vk = (\widetilde{vk}, P)$ and invokes the verification algorithm of the underlying signature scheme $b \leftarrow \mathsf{Verify}(\widetilde{vk}, m, \sigma)$.

Correctness. The key generation algorithm on input sk outputs a verification key vk. We need to show that for any sk' close enough to sk, the signature generated by sk' will be verified as valid. Observe that the signature algorithm $\mathsf{Sign_F}$ on input a fuzzy signing key sk' and the verification key vk will invoke the reproduction algorithm Rep to reproduce the signing key $\widetilde{sk}$ and use $\widetilde{sk}$ as the signing key to sign the message m. By the correctness of fuzzy extractor, if sk and sk' are close enough, then $\widetilde{sk}$ can be accurately reproduced, then by the correctness of the underlying signature scheme, the signature by sk' can be verified valid.

Security. The security of our generic construction of fuzzy signature scheme can be reduced to the security of the underlying signature scheme and the underlying fuzzy extractor. The security of underlying fuzzy extractor guarantees that if the input sk has enough entropy, then the extracted signing key $\widetilde{sk}$ is uniformly distributed. By the correctness of fuzzy extractor, if sk' is close enough to sk, then $\widetilde{sk}$ will always be accurately reproduced. Then the security of the fuzzy signature can be reduced to the security of the underlying signature scheme.

2 Preliminaries

Let $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{R}$ denote the sets of natural numbers, integers and real numbers, respectively. For a natural number $n \in \mathbb{N}$, we define $[n] := \{1, \cdots, n\}$. The notion "$x \leftarrow y$" denotes that y is (deterministically) assigned to x. For a finite set S, the notion "$|S|$" refers to the size of S, and the notion "$x \leftarrow_{\$} S$" refers to that x is uniformly chosen from S. For two bit-strings x and y, the notion $|x|$ refers to the bit length of x, and the notion $x \parallel y$ refers to the concatenation of x and y. "PPT" is short for probabilistic polynomial-time. If $\mathcal{A}$ is a probabilistic algorithm, the notion $y \leftarrow \mathcal{A}(x; r)$ refers to that $\mathcal{A}$ runs with input x and randomness r and outputs y. For a primitive XX and a security notion YY, by $\mathsf{Expt}^{\mathrm{YY}}_{\mathrm{XX},\mathcal{A}}(k) \Rightarrow 1$, we mean that the security experiment outputs 1 after interacting with an adversary $\mathcal{A}$. By $\mathsf{Adv}^{\mathrm{YY}}_{\mathrm{XX},\mathcal{A}}(k)$, we denote the advantage of a PPT adversary $\mathcal{A}$ and define $\mathsf{Adv}^{\mathrm{YY}}_{\mathrm{XX}}(\lambda) := \max_{\mathrm{PPT}\,\mathcal{A}} \mathsf{Adv}^{\mathrm{YY}}_{\mathrm{XX},k}(k)$.

Definition 1 (Negligible function). *A function f is negligible if for every polynomial $p(\cdot)$ there exists an N such that for all integers $n > N$ it holds that $f(n) < \frac{1}{p(n)}$.*

Definition 2 (Metric spaces). *A metric space is a set $\mathcal{W}$ with a distance function $\mathsf{dis} : \mathcal{W} \times \mathcal{W} \to \mathbb{R}^+ = [0, \infty)$. For all $x, y, z \in \mathcal{W}$, the distance function should satisfy the following conditions:*

1. *Reflexivity :* $\mathsf{dis}(x, y) = 0$ *if and only if* $x = y$*;*

2. *Symmetry :* $\mathsf{dis}(x, y) = \mathsf{dis}(y, x)$;
3. *Triangle inequality :* $\mathsf{dis}(x, z) \leq \mathsf{dis}(x, y) + \mathsf{dis}(y, z)$.

We usually consider multi-dimensional metric spaces of form $\mathcal{W} = \mathcal{F}^n$ for some alphabet $\mathcal{F}$ (usually a finite filed $\mathcal{F}^p$) equipped with the Hamming distance. For any two element $x, y \in \mathcal{W}$, the Hamming distance $\mathsf{dis}(x, y)$ is the number of coordinates in which they differ. For an element $x \in \mathcal{W}$, let $\mathsf{dis}(x) := \mathsf{dis}(x, 0)$.

Definition 3 (Min-entropy). *[12] For a random variable* X*, the min-entropy of* X*, denoted by* $H_\infty(\mathsf{X})$*, is defined by*

$$H_\infty(\mathsf{X}) := -log_2(max_x \Pr[\mathsf{X} = x]).$$

Definition 4 (Average min-entropy). *[12] For two random variables* X *and* Y*, the average min-entropy of* X *given* Y*, denoted by* $\widetilde{H}_\infty(\mathsf{X}|\mathsf{Y})$*, is defined by*

$$\widetilde{H}_\infty(\mathsf{X}|\mathsf{Y}) := -log_2(\underset{y \leftarrow \mathsf{Y}}{E} [\underset{x \in \mathsf{X}}{max} \Pr[\mathsf{X} = x|\mathsf{Y} = y]]).$$

Definition 5 (Statistical distance). *The statistical distance between two probability distributions* A *and* B *is*

$$\mathbf{SD}(A, B) \ = \frac{1}{2} \sum_v \ |\Pr\ (A = v) - \Pr(B = v)|.$$

Definition 6 (Universal hash functions). *[3] A family of hash functions* $\mathcal{H}_\mathcal{I} = \{\mathsf{H_i} : \mathcal{X} \to \mathcal{Y}\}$ *is universal, if for all distinct* $x, x' \in \mathcal{X}$*,*

$$\Pr[\mathsf{H_i} : \mathsf{H_i}(x) = \mathsf{H_i}(x')\ ] \leq \frac{1}{|\mathcal{Y}|},$$

where i *is uniformly chosen from* $\mathcal{I}$*.*

Definition 7 (Strong extractor). *[10] A function* $\mathsf{Ext} : \mathcal{X} \times \mathcal{I} \to \mathcal{T}$ *is an* average-case $(\mathcal{X}, \mathfrak{m}, \mathcal{T}, \varepsilon)$-strong extractor *with seed* $I \in \mathcal{I}$*, if for any variable* X *over* $\mathcal{X}$ *and any variable* Y *such that* $\tilde{H}_\infty(X|Y) \geq \mathfrak{m}$*, we have*

$$\mathbf{SD}((\mathsf{Ext}(X, I), Y, I), (U, Y, I)) \leq \varepsilon,$$

where I *and* U *are uniformly distributed over* $\mathcal{I}$ *and* $\mathcal{T}$*, respectively.*

In particular, universal hash functions are average-case $(\mathcal{X}, \mathfrak{m}, \mathcal{T}, \varepsilon)$-strong extractors.

Definition 8 (Secure sketch). *[6] A* $(\mathcal{W}, \mathfrak{m}, \widetilde{\mathfrak{m}}, t)$*-secure sketch consists of a pair of PPT algorithms* $(\mathsf{SS}, \mathsf{Rec})$ *with the following specifications:*

- $\mathsf{SS}(w)$ *on input* $w \in \mathcal{W}$ *outputs a sketch* s*.*

– $\mathsf{Rec}(w', s)$ *on input* $w' \in \mathcal{W}$ *and a sketch* s *outputs* $\widetilde{w}$.

It also satisfies the following properties:
Correctness. *If* $\mathsf{dis}(w, w') \leq t$, *then* $w = \mathsf{Rec}(w', \mathsf{SS}(w))$.
Privacy. *For any distribution* W *over* $\mathcal{W}$, *if* $H_\infty(W) \geq \mathfrak{m}$, *then* $\widetilde{H}_\infty(W|\mathsf{SS}(W)) \geq \widetilde{\mathfrak{m}}$.

An instantiation of secure sketch is the syndrome-based secure sketch [5]. Recall that an efficiently decodable $[n, k, 2t+1]_{\mathcal{F}}$-linear error correcting code $\mathcal{C}$ can correct up to t errors and it is a linear subspace of $\mathcal{F}^n$ of dimension k. The parity-check matrix of $\mathcal{C}$ is an $(n-k) \times n$ matirx H whose rows generate the orthogonal space $\mathcal{C}^\perp$. For any $v \in \mathcal{F}^n$, the syndrome of v is defined by $\mathsf{syn}(v) := Hv$. Note that $v \in \mathcal{C} \Leftrightarrow \mathsf{syn}(v) = 0$. For any $c \in \mathcal{C}$, $\mathsf{syn}(c+e) = \mathsf{syn}(c) + \mathsf{syn}(e) = \mathsf{syn}(e)$. A linear error-correcting code implies a syndrome-based secure sketch as shown below.

– $\mathsf{SS}(w) := \mathsf{syn}(w) = s$.
– $\mathsf{Rec}(w', s) := w' - \mathsf{Decode}(\mathsf{syn}(w') - s)$.

2.1 Bilinear Groups

We say that $\mathcal{BG} = (p, \mathbb{G}, \mathbb{G}_T, g, e)$ constitutes (symmetric) bilinear groups if p is a prime, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups with order p, g is a generator of $\mathbb{G}$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficiently (in $|p|$) computable mapping satisfying the following two properties:

– **Bilinearity:** For all $g \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, it holds that $e(g^a, g^b) = e(g, g)^{ab}$.
– **Non-degeneracy:** For all generators g of $\mathbb{G}$, $e(g, g) \in \mathbb{G}_T$ is not the identity element of $\mathbb{G}_T$.

For convenience, we denote by BGGen an algorithm (referred to as a "bilinear group generator") that, on input 1^k, outputs a description of bilinear groups $\mathcal{BG} = (p, \mathbb{G}, \mathbb{G}_T, g, e)$ such that $|p| = \Theta(k)$.

2.2 Signature Scheme

Definition 9 (Signature scheme). *A signature scheme* SIG *is a tuple of PPT algorithms* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *satisfying the following:*

– $\mathsf{Setup}(k) \to pp$. *The setup algorithm* Setup *takes a security parameter* k *as input, and outputs the public parameter* pp.
– $\mathsf{KeyGen}(pp) \to (vk, sk)$. *The key generation algorithm* KeyGen *takes the public parameter* pp *as input, and outputs a verification/signing key pair* (vk, sk).
– $\mathsf{Sign}(pp, sk, m) \to \sigma$. *The signing algorithm* Sign *takes the public parameter* pp, *the signing key* sk *and a message* m *from message space* $\mathcal{M}$ *as input, and outputs a signature* σ.

– $\mathsf{Verify}(pp, vk, m, \sigma) \to 1/0$. *The (deterministic) verification algorithm* Verify *takes the public parameter pp, a verification key vk, a message m and a signature σ as input, and outputs a bit b, with $b = 1$ meaning valid and $b = 0$ meaning invalid.*

Correctness. We require that for all $pp \leftarrow \mathsf{Setup}(k)$, $(vk, sk) \leftarrow \mathsf{KeyGen}(pp)$, and messages $m \in \mathcal{M}$, we have

$$\mathsf{Verify}(pp, vk, m, \mathsf{Sign}(pp, sk, m)) = 1.$$

Definition 10 (EUF-CMA security). *A signature scheme* SIG *is said to be existentially unforgeable under non-adaptive chosen message attack (EUF-CMA), if for all PPT adversaries $\mathcal{A}$,*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SIG},\mathcal{A}}(k) := \Pr[\ \mathsf{Expt}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SIG},\mathcal{A}}(k) \Rightarrow 1\ ] \leqslant \mathsf{negl}(k).$$

Here $\mathsf{Expt}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SIG},\mathcal{A}}(k)$ *is an experiment played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows.*

$\underline{\mathsf{Expt}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{SIG},\mathcal{A}}(k):}$

1. *The challenger $\mathcal{C}$ invokes $pp \leftarrow \mathsf{Setup}(k)$, generates $(vk, sk) \leftarrow \mathsf{KeyGen}(pp)$ and initializes the set of chosen-message queries $\mathcal{Q} = \emptyset$ issued by the adversary. Subsequently, it returns pp and vk to $\mathcal{A}$.*
2. *The adversary $\mathcal{A}$ may adaptively make signing oracle queries of the following form*
   – *$\mathcal{A}$ sends a message $m_i \in \mathcal{M}$ to the challenger $\mathcal{C}$.*
   – *$\mathcal{C}$ invokes $\sigma_i \leftarrow \mathsf{Sign}(pp, sk, m_i)$, adds m_i to the set $\mathcal{Q}$ and returns σ_i to $\mathcal{A}$.*
3. *Finally, $\mathcal{A}$ submits a message-signature pair (m^*, σ^*). The experiment outputs 1 if $m^* \notin \mathcal{Q} \wedge \mathsf{Verify}(pp, vk, m^*, \sigma^*) = 1$ and 0 otherwise.*

Simple key generation process. We will use signature schemes with a structural property which is called the *simple key generation process* property. Simple key generation process property says that the key generation algorithm KeyGen first picks a secret key sk uniformly at random from the secret key space, then computes the corresponding verification key vk deterministically from sk.

Definition 11 (Simple key generation process, [14]). *Let* SIG=($\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}$) *be a signature scheme. We say* SIG *has a simple key generation process if each pp output by* Setup *specifies the secret key space $\mathcal{SK}_{pp}$, and there exists a deterministic PPT algorithm* KG *such that the key generation algorithm* $\mathsf{KeyGen}(pp)$ *can be written as follows:*

$$\mathsf{KeyGen}(pp) : [\ sk \leftarrow_{\$} \mathcal{SK}_{pp}; vk \leftarrow \mathsf{KG}(pp, sk); Return\ (vk, sk).\ ].$$

2.3 Fuzzy Extractor

Definition 12 (Fuzzy extractor). *[6] An $(\mathcal{W}, \mathtt{m}, \mathcal{R}, t, \epsilon)$-fuzzy extractor $\mathsf{FE} = (\mathsf{Init}, \mathsf{Gen}, \mathsf{Rep})$ consists of three PPT algorithms:*

- $\mathsf{Init}(k) \to pp$. *The initialization algorithm* Init *takes a security parameter k as input, and outputs the public parameter pp.*
- $\mathsf{Gen}(pp, w) \to (P, R)$. *The generation algorithm* Gen *takes the public parameter pp and $w \in \mathcal{W}$ as input, and outputs a public helper string P and an extracted string $R \in \mathcal{R}$.*
- $\mathsf{Rep}(pp, P, w') \to R/\perp$. *The reproduction algorithm* Rep *takes pp, P and $w' \in \mathcal{W}$ as input, and outputs an extracted string R or $\perp$.*

Correctness. If $\mathsf{dis}(w, w') \leq t$, then for all $(R, P) \leftarrow \mathsf{Gen}(pp, w)$, it holds that $R \leftarrow \mathsf{Rep}(pp, P, w')$.

Security. Let W be a distribution on $\mathcal{W}$, if $H_\infty(W) \geq m$, then for all PPT adversaries $\mathcal{A}$,

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{FE},\mathcal{A}}(k) = |\Pr[\mathcal{A}(P, R) \Rightarrow 1] - \Pr[\mathcal{A}(P, U) \Rightarrow 1]| \leq \epsilon,$$

where $(P, R) \leftarrow \mathsf{Gen}(pp, w)$, $w \leftarrow W$ and $U \leftarrow_\$ \mathcal{R}$.

3 Fuzzy Signature

A fuzzy signature scheme is a special signature scheme whose signing key is no longer required to be uniformly random, but can be a noise random string such as biometric data. We call such signing key as fuzzy signing key. Here, we give the formal definition of fuzzy signature, which is a little different from the definition by Takahashi et al. [14]. The difference between our fuzzy signature and Takahashi et al.'s is that when generating a signature of a message m, our signature algorithm not only needs the fuzzy signing key but also needs vk as input while Takahashi et al.'s does not need vk.

Definition 13 (Fuzzy signature). *Let $\mathcal{SK}$ be the fuzzy signing key space and $\mathcal{M}$ be the message space. An $(\mathcal{SK}, \mathcal{M}, t)$-fuzzy signature scheme $\mathsf{SIG_F}$ consists of the following four PPT algorithms $(\mathsf{Setup_F}, \mathsf{KeyGen_F}, \mathsf{Sign_F}, \mathsf{Verify_F})$:*

- $\mathsf{Setup_F}(k) \to pp$. *The setup algorithm* $\mathsf{Setup_F}$ *takes a security parameter k as input, and outputs a public parameter pp.*
- $\mathsf{KeyGen_F}(pp, sk) \to vk$. *The key generation algorithm* $\mathsf{KeyGen_F}$ *takes the public parameter pp and a fuzzy signing key $sk \in \mathcal{SK}$ as input, and outputs a verification key vk.*
- $\mathsf{Sign_F}(pp, vk, sk', m) \to \sigma$. *The signing algorithm* $\mathsf{Sign_F}$ *takes the public parameter pp, the verification key vk, a new fuzzy signing key $sk' \in \mathcal{SK}$ and a message $m \in \mathcal{M}$ as input, and outputs a signature σ.*
- $\mathsf{Verify_F}(pp, vk, m, \sigma) \to 0/1$. *The verification algorithm* $\mathsf{Verify_F}$ *takes the public parameter pp, the verification key vk, the message m and the signature σ as input, and outputs a bit 1 (accept) or 0 (reject).*

Correctness. We require that for all $pp \leftarrow \mathsf{Setup_F}(k)$, $vk \leftarrow \mathsf{KeyGen_F}(pp, sk)$, and messages $m \in \mathcal{M}$, if $\mathsf{dis}(sk, sk') \leq t$, it holds that

$$\mathsf{Verify_F}(pp, vk, m, \mathsf{Sign_F}(pp, vk, sk', m)) = 1.$$

In [14], Takahashi et al. assumed that the error distribution between sk and sk' (two samples of the same biometric characteristic) are independent of the biometric characteristic. More precisely, they assume that for all objects W that produce the fuzzy data (which will be used as the fuzzy signing key), if W produces a data sk at the first measurement, and if the same object is measured next time, then the measured data sk' follows the distribution $\{e \leftarrow \Phi; sk' \leftarrow sk + e : sk'\}$, and error distribution Φ is independent of W. This requirement is too strong since it seems impossible that error distribution between different samples is independent of the biometric characteristic. So we try to relax the requirement in [14] and define the following security model of fuzzy signature.

Definition 14 ($\mathfrak{m}$-EUF-CMA security). *An $(\mathcal{SK}, \mathcal{M}, t)$- fuzzy signature scheme is said to be $\mathfrak{m}$-existentially unforgeable under non-adaptive chosen message attack ($\mathfrak{m}$-EUF-CMA), if for any distribution SK over metric space $\mathcal{SK}$ with $H_\infty(SK) \geq \mathfrak{m}$, for any PPT adversaries $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathfrak{m}\text{-}\mathit{EUF\text{-}CMA}}_{\mathsf{SIG_F},\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathfrak{m}\text{-}\mathit{EUF\text{-}CMA}}_{\mathsf{SIG_F},\mathcal{A}}(k) \Rightarrow 1] \leqslant \mathsf{negl}(k).$$

Here $\mathsf{Expt}^{\mathfrak{m}\text{-}\mathit{EUF\text{-}CMA}}_{\mathsf{SIG},\mathcal{A}}(k)$ is an experiment played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows.

$\mathsf{Expt}^{\mathfrak{m}\text{-}\mathit{EUF\text{-}CMA}}_{\mathsf{SIG},\mathcal{A}}(k)$:

1. *The challenger $\mathcal{C}$ invokes $pp \leftarrow \mathsf{Setup_F}(k)$, samples $sk \leftarrow SK$, generates $vk \leftarrow \mathsf{KenGen_F}(pp, sk)$ and initializes the set of chosen-message queries $\mathcal{Q} = \emptyset$ issued by the adversary. Subsequently, it returns pp and vk to $\mathcal{A}$.*
2. *The adversary $\mathcal{A}$ may adaptively make signing oracle queries of the following form*
   - *$\mathcal{A}$ sends a message $m_i \in \mathcal{M}$ and a function $f_i \in \Phi$ to challenger $\mathcal{C}$, where $\Phi := \{\forall sk \in \mathcal{SK}, f(sk) < t | f : \mathcal{SK} \to \mathcal{SK}\}$.*
   - *$\mathcal{C}$ invokes $\sigma_i \leftarrow \mathsf{Sign_F}(pp, vk, sk + f_i(sk), m_i)$, adds m_i to the set $\mathcal{Q}$ and returns σ_i to $\mathcal{A}$.*
3. *Finally, $\mathcal{A}$ submits a message-signature pair (m^*, σ^*). The experiment outputs 1 if $m^* \notin \mathcal{Q} \wedge \mathsf{Verify_F}(pp, vk, m^*, \sigma^*) = 1$ and 0 otherwise.*

Remark 1. In our security model, when adversary $\mathcal{A}$ making a signing oracle query of message m_i, $\mathcal{A}$ will send a function $f_i \in \Phi$ to challenger $\mathcal{C}$ as well. Note that $\Phi := \{\forall sk \in \mathcal{SK}, f(sk) < t | f : \mathcal{SK} \to \mathcal{SK}\}$. In this way, we model that the error distribution between different samples of the same biometric characteristic can be arbitrary depend on the biometric characteristic except the error is bound by t. It is reasonable to bound the error, because different samples of the same biometric characteristic are similar. Meanwhile, it is necessary to bound the error, otherwise there is no security at all.

4 Construction of Fuzzy Signature

Figure 2 illustrates our construction of fuzzy signature $\mathsf{SIG_F} = (\mathsf{Setup_F}, \mathsf{KeyGen_F}, \mathsf{Sign_F}, \mathsf{Verify_F})$ which makes use of the following building blocks:

- A signature scheme SIG=($\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}$) with a simple key generation process (i.e., there exists a deterministic PPT algorithm KG). Let its secret key space be $\widetilde{\mathcal{SK}}$ and message space be $\mathcal{M}$.
- An $(\mathcal{SK}, \mathfrak{m}, \widetilde{\mathcal{SK}}, t, \epsilon)$-fuzzy extractor FE=($\mathsf{Init}, \mathsf{Gen}, \mathsf{Rep}$).

$\underline{\mathsf{Setup_F}(k)}$:	$\underline{\mathsf{KeyGen_F}(pp, sk)}$:	$\underline{\mathsf{Sign_F}(pp, vk, sk', m)}$:	$\underline{\mathsf{Verify_F}(pp, vk, m, \sigma)}$:
$pp_1 \leftarrow \mathsf{Init}(k)$	$(pp_1, pp_2) \leftarrow pp$	$(pp_1, pp_2) \leftarrow pp$	$(pp_1, pp_2) \leftarrow pp$
$pp_2 \leftarrow \mathsf{Setup}(k)$	$(P, \widetilde{sk}) \leftarrow \mathsf{Gen}(pp_1, sk)$	$(\widetilde{vk}, P) \leftarrow vk$	$(\widetilde{vk}, P) \leftarrow vk$
$pp \leftarrow (pp_1, pp_2)$	$\widetilde{vk} \leftarrow \mathsf{KG}(pp_2, \widetilde{sk})$	$\widetilde{sk}' \leftarrow \mathsf{Rep}(pp_1, P, sk')$	$b \leftarrow \mathsf{Verify}(pp_2, \widetilde{vk}, m, \sigma)$
Return pp	$vk \leftarrow (\widetilde{vk}, P)$	$\sigma \leftarrow \mathsf{Sign}(pp_2, \widetilde{sk}', m)$	Return b
	Return vk	Return σ	

Fig. 2. Our generic construction of fuzzy signature scheme $\mathsf{SIG_F}$.

4.1 Correctness

The correctness of our fuzzy signature scheme $\mathsf{SIG_F}$ is guaranteed as follows.

Theorem 1. *The correctness of* $\mathsf{SIG_F}$ *follows from the correctness of the underlying signature scheme* SIG *and the underlying* $(\mathcal{SK}, \mathfrak{m}, \widetilde{\mathcal{SK}}, t, \epsilon)$*-fuzzy extractor* FE.

Proof. By the correctness of FE, if $\mathsf{dis}(sk, sk') \leq t$, then $\widetilde{sk}' = \widetilde{sk}$, where $(P, \widetilde{sk}) \leftarrow \mathsf{Gen}(pp_1, sk)$, $\widetilde{sk}' \leftarrow \mathsf{Rep}(pp_1, P, sk')$. Note that $\widetilde{vk} \leftarrow \mathsf{KG}(pp_2, \widetilde{sk})$, by the correctness of SIG, for any message $m \in \mathcal{M}$, it follows that

$$\mathsf{Verify}(pp_2, \widetilde{vk}, m, \mathsf{Sign}(pp_2, \widetilde{sk}', m)) = 1.$$

More precisely,

$$\begin{aligned}
& \mathsf{Verify_F}(pp, vk, m, \mathsf{Sign_F}(pp, vk, sk', m)) \\
= \; & \mathsf{Verify}(pp_2, \widetilde{vk}, m, \mathsf{Sign_F}(pp, vk, sk', m)) && \text{(by the construction)} \\
= \; & \mathsf{Verify}(pp_2, \widetilde{vk}, m, \mathsf{Sign}(pp_2, \widetilde{sk}', m)) && \text{(by the construction)} \\
= \; & \mathsf{Verify}(pp_2, \widetilde{vk}, m, \mathsf{Sign}(pp_2, \widetilde{sk}, m)) && \text{(by the correctness of } \mathsf{FE}) \\
= \; & 1 && \text{(by the correctness of } \mathsf{SIG})
\end{aligned}$$

■

4.2 Security

The security of our fuzzy signature scheme $\mathsf{SIG_F}$ is guaranteed as follows.

Theorem 2. *If the underlying signature scheme* SIG *with secret key space* $\widetilde{\mathcal{SK}}$ *and message space* $\mathcal{M}$ *satisfies* ***EUF-CMA*** *security and the underlying* FE *is an* $(\mathcal{SK}, \mathfrak{m}, \widetilde{\mathcal{SK}}, t, \epsilon)$*-fuzzy extractor, our construction* $\mathsf{SIG_F}$ *is an* $(\mathcal{SK}, \mathcal{M}, t)$*-fuzzy signature that satisfies* $\mathfrak{m}$*-**EUF-CMA** security.*

Proof. Let $\mathcal{A}$ be an arbitrary PPT algorithm adversary that attacks the $\mathfrak{m}$-EUF-CMA security of $\mathsf{SIG_F}$. We will consider three indistinguishable games, where the first game $\mathsf{Game\ 0}$ is the original game $\mathsf{Expt}^{\mathfrak{m}\text{-}\mathsf{EUF\text{-}CMA}}_{\mathsf{SIG_F},\mathcal{A}}(k)$. For $i \in \{0, 1, 2\}$, denote by S_i the event that $\mathcal{A}$ wins (i.e., the experiment returns 1) in $\mathsf{Game\ i}$. Our goal is to show that $\mathsf{Adv}^{\mathfrak{m}\text{-}\mathsf{EUF\text{-}CMA}}_{\mathsf{SIG_F},\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathfrak{m}\text{-}\mathsf{EUF\text{-}CMA}}_{\mathsf{SIG_F},\mathcal{A}}(k) \Rightarrow 1]$ is negligible.

Game 0. $\mathsf{Game\ 0}$ is just the experiment $\mathsf{Expt}^{\mathfrak{m}\text{-}\mathsf{EUF\text{-}CMA}}_{\mathsf{SIG_F},\mathcal{A}}(k)$. More precisely,

1. The challenger $\mathcal{C}$ invokes $pp_1 \leftarrow \mathsf{Init}(k)$ and $pp_2 \leftarrow \mathsf{Setup}(k)$, samples $sk \leftarrow SK$, invokes $(P, \widetilde{sk}) \leftarrow \mathsf{Gen}(pp_1, sk)$ and $\widetilde{vk} \leftarrow \mathsf{KG}(pp_2, \widetilde{sk})$, sets $pp \leftarrow (pp_1, pp_2)$ and $vk \leftarrow (\widetilde{vk}, P)$ and initializes the set of chosen-message queries $\mathcal{Q} = \emptyset$ issued by the adversary. Subsequently, it returns pp and vk to $\mathcal{A}$.
2. The adversary $\mathcal{A}$ may adaptively make signing oracle queries of the following form
   - $\mathcal{A}$ sends a message $m_i \in \mathcal{M}$ and a function $f_i \in \Phi$ to challenger $\mathcal{C}$, where $\Phi := \{\forall sk \in \mathcal{SK}, f(sk) < t | f : \mathcal{SK} \to \mathcal{SK}\}$.
   - $\mathcal{C}$ parses pp as pp_1 and pp_2, parses vk as $\widetilde{vk}$ and P, invokes $\widetilde{sk'}_i \leftarrow \mathsf{Rep}(pp_1, P, sk + f_i(sk))$, $\sigma_i \leftarrow \mathsf{Sign}(pp_2, \widetilde{sk'}_i, m_i)$, adds m_i to the set $\mathcal{Q}$ and returns σ_i to $\mathcal{A}$.
3. Finally, $\mathcal{A}$ submits a message-signature pair (m^*, σ^*). Then, challenger $\mathcal{C}$ parses pp as pp_1 and pp_2, parses vk as $\widetilde{vk}$ and P, invokes $b \leftarrow \mathsf{Verify}(pp_2, \widetilde{vk}, m^*, \sigma^*)$. The experiment outputs 1 if $m^* \notin \mathcal{Q} \wedge b = 1$ and 0 otherwise.

Obviously,

$$\mathsf{Adv}^{\mathfrak{m}\text{-}\mathsf{EUF\text{-}CMA}}_{\mathsf{SIG_F},\mathcal{A}}(k) = \Pr[\mathsf{Expt}^{\mathfrak{m}\text{-}\mathsf{EUF\text{-}CMA}}_{\mathsf{SIG_F},\mathcal{A}}(k) \Rightarrow 1] = \Pr[S_0].$$

Game 1. This game is identical to $\mathsf{Game\ 0}$, except that in step 2, when the challenger answers the signing oracle queries, it uses $\widetilde{sk}$ as the signing key of the underlying signature scheme other than $\widetilde{sk'}_i \leftarrow \mathsf{Rep}(pp_1, P, sk'_i)$. More precisely,

1. The challenger $\mathcal{C}$ invokes $pp_1 \leftarrow \mathsf{Init}(k)$ and $pp_2 \leftarrow \mathsf{Setup}(k)$, samples $sk \leftarrow SK$, invokes $(P, \widetilde{sk}) \leftarrow \mathsf{Gen}(pp_1, sk)$ and $\widetilde{vk} \leftarrow \mathsf{KG}(pp_2, \widetilde{sk})$, sets $pp \leftarrow (pp_1, pp_2)$ and $vk \leftarrow (\widetilde{vk}, P)$ and initializes the set of chosen-message queries $\mathcal{Q} = \emptyset$ issued by the adversary. Subsequently, it returns pp and vk to $\mathcal{A}$.
2. The adversary $\mathcal{A}$ may adaptively make signing oracle queries of the following form
   - $\mathcal{A}$ sends a message $m_i \in \mathcal{M}$ and a function $f_i \in \Phi$ to challenger $\mathcal{C}$, where $\Phi := \{\forall sk \in \mathcal{SK}, f(sk) < t | f : \mathcal{SK} \to \mathcal{SK}\}$.

- $\mathcal{C}$ parses pp as pp_1 and pp_2, invokes $\underline{\sigma_i \leftarrow \mathsf{Sign}(pp_2, \widetilde{sk}, m_i)}$, adds m_i to the set $\mathcal{Q}$ and returns σ_i to $\mathcal{A}$.

3. Finally, $\mathcal{A}$ submits a message-signature pair (m^*, σ^*). Then, challenger $\mathcal{C}$ parses pp as pp_1 and pp_2, parses vk as $\widetilde{vk}$ and P, invokes $b \leftarrow \mathsf{Verify}(pp_2, \widetilde{vk}, m^*, \sigma^*)$. The experiment outputs 1 if $m^* \notin \mathcal{Q} \wedge b = 1$ and 0 otherwise.

Lemma 1. $\Pr[S_1] = \Pr[S_0]$.

Proof. The only difference between Game 0 and Game 1 is the signing key used in the signing oracle. In Game 1 the challenger uses $\widetilde{sk}$ as the signing key, while in Game 0 the challenger uses $\widetilde{sk'}_i$ where $\widetilde{sk'}_i \leftarrow \mathsf{Rep}(pp_1, P, sk + f_i(sk))$.

Since the function f_i is chosen by the adversary $\mathcal{A}$ from the set Φ, where $\Phi := \{\forall sk \in \mathcal{SK}, f(sk) < t | f : \mathcal{SK} \to \mathcal{SK}\}$, we have $\mathsf{dis}(sk, sk + f_i(sk)) \leq t$. By the correctness of the underlying $(\mathcal{SK}, \mathfrak{m}, \widetilde{\mathcal{SK}}, t, \epsilon)$-fuzzy extractor FE, we have $\widetilde{sk'}_i = \widetilde{sk}$. Therefore, the change between Game 0 and Game 1 is just conceptual. It follows that

$$\Pr[S_1] = \Pr[S_0].$$

□

Game 2. This game is identical to Game 1, except that the signing key $\widetilde{sk}$ is uniformly chosen from $\widetilde{\mathcal{SK}}$ other than $(P, \widetilde{sk}) \leftarrow \mathsf{Gen}(pp_1, sk)$. More precisely,

1. The challenger $\mathcal{C}$ invokes $pp_1 \leftarrow \mathsf{Init}(k)$ and $pp_2 \leftarrow \mathsf{Setup}(k)$, samples $sk \leftarrow SK$, invokes $(P, \widetilde{sk}) \leftarrow \mathsf{Gen}(pp_1, sk)$, $\underline{\text{samples } U \leftarrow_{\$} \widetilde{\mathcal{SK}}, \text{ set } \widetilde{sk} \leftarrow U}$, invokes $\widetilde{vk} \leftarrow \mathsf{KG}(pp_2, \widetilde{sk})$, sets $pp \leftarrow (pp_1, pp_2)$ and $vk \leftarrow (\widetilde{vk}, P)$ and initializes the set of chosen-message queries $\mathcal{Q} = \emptyset$ issued by the adversary. Subsequently, it returns pp and vk to $\mathcal{A}$.
2. The adversary $\mathcal{A}$ may adaptively make signing oracle queries of the following form
   - $\mathcal{A}$ sends a message $m_i \in \mathcal{M}$ and a function $f_i \in \Phi$ to challenger $\mathcal{C}$, where $\Phi := \{\forall sk \in \mathcal{SK}, f(sk) < t | f : \mathcal{SK} \to \mathcal{SK}\}$.
   - $\mathcal{C}$ parses pp as pp_1 and pp_2, invokes $\sigma_i \leftarrow \mathsf{Sign}(pp_2, \widetilde{sk}, m_i)$, adds m_i to the set $\mathcal{Q}$ and returns σ_i to $\mathcal{A}$.
3. Finally, $\mathcal{A}$ submits a message-signature pair (m^*, σ^*). Then, challenger $\mathcal{C}$ invokes $b \leftarrow \mathsf{Verify}(pp_2, \widetilde{vk}, m^*, \sigma^*)$. The experiment outputs 1 if $m^* \notin \mathcal{Q} \wedge b = 1$ and 0 otherwise.

Lemma 2. $|\Pr[S_1] - \Pr[S_2]| \leq \epsilon$.

Proof. The only difference between Game 1 and Game 2 is the way of generating the signing key $\widetilde{sk}$ (not fuzzy). In Game 1, $\widetilde{sk}$ is generated by $(P, \widetilde{sk}) \leftarrow \mathsf{Gen}(pp_1, sk)$, while in Game 2, $\widetilde{sk}$ is uniformly chosen from $\widetilde{\mathcal{SK}}$.

By the security of the underlying $(\mathcal{SK}, \mathfrak{m}, \widetilde{\mathcal{SK}}, t, \epsilon)$-fuzzy extractor FE, if SK is a distribution over $\mathcal{SK}$ with $H_\infty(SK) \geq \mathfrak{m}$, then

$$|\Pr[\mathcal{A}(P, \widetilde{sk}) \Rightarrow 1] - \Pr[\mathcal{A}(P, U) \Rightarrow 1]| \leq \epsilon,$$

where $(P, \widetilde{sk}) \leftarrow \mathsf{Gen}(pp_1, sk)$, $sk \leftarrow SK$ and $U \leftarrow_{\$} \widetilde{\mathcal{SK}}$. Therefore, we have

$$|\Pr[S_1] - \Pr[S_2]| \leq \epsilon.$$

□

Lemma 3. $\Pr[S_2] \leq \mathsf{Adv}_{\mathsf{SIG}}^{\mathit{EUF\text{-}CMA}}(k)$.

Proof. We will reduce the EUF-CMA security of the underlying signature scheme SIG to the altered game as described in Game 2. To this end, we assume a PPT adversary $\mathcal{A}$ winning Game 2 and show how to construct a PPT adversary $\mathcal{B}$ attacking the underlying EUF-CMA secure signature SIG.

On receiving pp_2 and $\widetilde{vk}$ from its own challenger, adversary $\mathcal{B}$ invokes $pp_1 \leftarrow \mathsf{Init}(k)$, samples $sk \leftarrow SK$, invokes $(P, \widetilde{sk}) \leftarrow \mathsf{Gen}(pp_1, sk)$, sets $pp \leftarrow (pp_1, pp_2)$ and $vk \leftarrow (\widetilde{vk}, P)$. Finally, $\mathcal{B}$ returns pp and vk to $\mathcal{A}$.

Upon receiving the signing oracle queries $m_i \in \mathcal{M}$ and a function $f_i \in \Phi$ from $\mathcal{A}$, adversary $\mathcal{B}$ answers $\mathcal{A}$'s query as follows:

1. $\mathcal{B}$ sends m_i to its own challenger.
2. Upon receiving σ_i from its own challenger, $\mathcal{B}$ returns σ_i to $\mathcal{A}$.

Finally, $\mathcal{A}$ submits a message-signature pair (m^*, σ^*) to $\mathcal{B}$. $\mathcal{B}$ submits (m^*, σ^*) to its own challenger and returns what its challenger returns.

Note that $\mathcal{B}$ simulates Game 2 perfectly. If $\mathcal{A}$ wins in Game 2, then $\mathcal{B}$ wins in the EUF-CMA game. By the security of the underlying EUF-CMA secure signature scheme SIG, we have

$$\Pr[S_2] = \mathsf{Adv}_{\mathsf{SIG},\mathcal{B}}^{\mathsf{EUF\text{-}CMA}}(k) \leq \mathsf{Adv}_{\mathsf{SIG}}^{\mathsf{EUF\text{-}CMA}}(k).$$

□

From lemma 1, 2 and 3, we have

$$\mathsf{Adv}_{\mathsf{SIG_F}}^{\mathsf{m\text{-}EUF\text{-}CMA}}(k) \leq \mathsf{Adv}_{\mathsf{SIG}}^{\mathsf{EUF\text{-}CMA}}(k) + \epsilon,$$

and Theorem 2 follows. ■

5 Instantiation

Let us recall the fuzzy extractor in [6] which uses a $(\mathcal{W}, \mathfrak{m}, \widetilde{\mathfrak{m}}, t)$-secure sketch $(\mathsf{SS}, \mathsf{Rec})$ and an average-case $(\mathcal{W}, \widetilde{\mathfrak{m}}, \mathcal{R}, \epsilon)$-strong extractor Ext with seed set $\mathcal{I}$ as building blocks. The construction is as follows.

- $\mathsf{Init}(k)$: $i \leftarrow_{\$} \mathcal{I}$, $pp = i$.
- $\mathsf{Gen}(pp, w)$: Compute $s = \mathsf{SS}(w)$, set $P = s$, $R = \mathsf{Ext}(w, i)$. Output (P, R).
- $\mathsf{Rep}(pp, w', P)$: Recover $w = \mathsf{Rec}(w', s)$ and output $R = \mathsf{Ext}(w, i)$.

It was shown by Dodis et.al. [6] the above construction is an $(\mathcal{W}, \mathfrak{m}, \mathcal{R}, t, \epsilon)$-fuzzy extractor.

5.1 First Instantiation

Our first concrete instantiation of a fuzzy signature scheme is based on the Waters signature scheme [15], and thus we review it here. We consider the version where the setup and the key generation for each user are separated so that the scheme fits our syntax. Let $l = l(k)$ be a positive polynomial, and let BGGen be a bilinear group generator. The Waters signature scheme $\mathsf{SIG}_{\mathsf{Wat}}$ is shown in Figure 3, which is $\mathsf{EUF\text{-}CMA}$ secure if the CDH assumption holds with respect to BGGen.

$\underline{\mathsf{Setup}_{\mathsf{Wat}}(k)}$: $\mathcal{BG} := (p, \mathbb{G}, \mathbb{G}_T, g, e) \leftarrow \mathsf{BGGen}(1^k)$ $h, u', u_1, \cdots, u_l \leftarrow_\$ \mathbb{G}$ $pp \leftarrow (\mathcal{BG}, h, u', (u_i)_{i\in[l]})$ Return pp	$\underline{\mathsf{KeyGen}_{\mathsf{Wat}}(pp)}$: $sk \leftarrow_\$ \mathbb{Z}_p$ $vk \leftarrow g^{sk}$ Return (vk, sk)
$\underline{\mathsf{Sign}_{\mathsf{Wat}}(pp, sk, m)}$: Parse m as $(m_1 \| \cdots \| m_l) \in \{0,1\}^l$ $r \leftarrow_\$ \mathbb{Z}_p$ $\sigma_1 \leftarrow h^{sk} \cdot (u' \cdot \prod_{i\in[l]} u_i^{m_i})^r$, $\sigma_2 \leftarrow g^r$ Return $\sigma \leftarrow (\sigma_1, \sigma_2)$	$\underline{\mathsf{Verify}_{\mathsf{Wat}}(pp, vk, m, \sigma)}$: $(\sigma_1, \sigma_2) \leftarrow \sigma$ Parse m as $(m_1 \| \cdots \| m_l) \in \{0,1\}^l$ If $e(\sigma_2, u' \cdot \prod_{i\in[l]} u_i^{m_i}) \cdot e(vk, h) = e(\sigma_1, g)$ Then return 1 else return 0

Fig. 3. The Waters signature scheme $\mathsf{SIG}_{\mathsf{Wat}}$

Theorem 3. *[15] The signature scheme is (t, q, ϵ) existentially unforgeable assuming the decisional $(t, \frac{\epsilon}{16(n+1)q})$ BDH assumption holds, where $\lambda = \frac{1}{8(n+1)q}$.*

Obviously, the signature scheme $\mathsf{SIG}_{\mathsf{Wat}}$ in Figure 3 has the *simple key generation process* property in which $vk = g^{sk}$.

Note that the secret key space of Waters signature scheme is $\mathbb{Z}_p$. We want to construct a fuzzy extractor with extracted string space $\mathbb{Z}_p$. Let Ext be a universal hash function, i.e., $\mathsf{Ext}(\cdot, i) := \mathsf{H}_i(\cdot) : \mathbb{Z}_p^l \to \mathbb{Z}_p$, [3] where $i = (i_1, \cdots, i_l) \in \mathbb{Z}_p^l$, $x = (x_0, x_1, \cdots, x_l) \in \mathbb{Z}_p^{l+1}$, and

$$\mathsf{Ext}(x, i) = \mathsf{H}_i(x) = x_0 + i_1 x_1 + \cdots + i_l x_l, \tag{1}$$

and H is a family of universal hash functions according to [12]. If we instantiate the secure sketch by the syndrome-based secure sketch introduced in Sect. 2.3 and extractor by $\mathsf{Ext}(x, i)$(Eq. 1), then we get a fuzzy extractor FE_1. And the extracted string by FE_1 is uniformly distributed over $\mathbb{Z}_p$. Note that the secret key space of Waters signature $\mathsf{SIG}_{\mathsf{Wat}}$ is $\mathbb{Z}_p$.

By instantiating the signature SIG in Figure 2 with Waters signature $\mathsf{SIG}_{\mathsf{Wat}}$, the fuzzy extractor FE in Figure 2 with the above fuzzy extractor FE_1, we get

[3] One can always translate a binary string to an element in $\mathbb{Z}_p^l$ for a proper l.

a fuzzy signature scheme which is $\mathfrak{m}$-EUF-CMA secure under the computational Diffie-Hellman (CDH) assumption in bilinear groups in the standard model (See Fig. 4).

$\underline{\mathsf{Setup}_{\mathsf{F1}}(k)}$: $i \leftarrow_{\$} \mathbb{Z}_p^l$, $pp_1 \leftarrow i$ $\mathcal{BG} := (p, \mathbb{G}, \mathbb{G}_T, g, e) \leftarrow \mathsf{BGGen}(1^k)$ $h, u', u_1, \cdots, u_l \leftarrow_{\$} \mathbb{G}$ $pp_2 \leftarrow (\mathcal{BG}, h, u', (u_i)_{i\in[l]})$ $pp \leftarrow (pp_1, pp_2)$ Return pp	$\underline{\mathsf{KeyGen}_{\mathsf{F1}}(pp, sk)}$: $(pp_1, pp_2) \leftarrow pp$ $s \leftarrow \mathsf{SS}(sk)$, $P \leftarrow s$ $\widetilde{sk} \leftarrow \mathsf{Ext}(sk, i)$ $\widetilde{vk} \leftarrow g^{\widetilde{sk}}$ $vk \leftarrow (\widetilde{vk}, P)$ Return vk.
$\underline{\mathsf{Sign}_{\mathsf{F1}}(pp, vk, sk', m)}$: $(pp_1, pp_2) \leftarrow pp$, $(\widetilde{vk}, P) \leftarrow vk$ $\widetilde{sk}' \leftarrow \mathsf{Ext}(\mathsf{Rec}(sk', P), i)$ Parse m as $(m_1 \Vert \cdots \Vert m_l) \in \{0,1\}^l$ $r \leftarrow_{\$} \mathbb{Z}_p$ $\sigma_1 \leftarrow h^{\widetilde{sk}'} \cdot (u' \cdot \prod_{i\in[l]} u_i^{m_i})^r$, $\sigma_2 \leftarrow g^r$ Return $\sigma \leftarrow (\sigma_1, \sigma_2)$	$\underline{\mathsf{Verify}_{\mathsf{F1}}(pp, vk, m, \sigma)}$: $(pp_1, pp_2) \leftarrow pp$ $(\widetilde{vk}, P) \leftarrow vk$ $(\sigma_1, \sigma_2) \leftarrow \sigma$ Parse m as $(m_1 \Vert \cdots \Vert m_l) \in \{0,1\}^l$ If $e(\sigma_2, u' \cdot \prod_{i\in[l]} u_i^{m_i}) \cdot e(\widetilde{vk}, h) = e(\sigma_1, g)$ Then return 1 else return 0

Fig. 4. Our first instantiation $\mathsf{SIG}_{\mathsf{F1}}$

Corollary 1. *The construction in Figure 4 is an* $\mathfrak{m}$*-EUF-CMA secure fuzzy signature under the computational Diffie-Hellman (CDH) assumption in bilinear groups in the standard model.*

5.2 Second Instantiation

Our second concrete instantiation of a fuzzy signature scheme is based on the Lattice-based Signatures [8], and thus we review it here. Let $\mathcal{D}_\theta^m$ be the m-dimensional discrete Gaussian distribution for some standard deviation θ. The lattice-based signature scheme $\mathsf{SIG}_{\mathsf{Lat}}$ based on SIS is shown in Figure 5, which is secure in the random oracle model based on the worst-case hardness of the $\widetilde{O}(n^{1.5})$-SIVP problem in general lattices.

Theorem 4. *[8] If there is a polynomial-time forger, who makes at most s queries to the signing oracle and h queries to the random oracle* H*, who breaks the signature (with proper parameters) with probability δ, then there is a polynomial-time algorithm who can solve the l_2-$SIS_{q,u,v,\beta}$ problem for $\beta = (4\theta + 2dk)\sqrt{v} = \widetilde{O}(du)$ with probability $\approx \frac{\delta^2}{2(h+s)}$.*

$\mathsf{Setup}_{\mathsf{Lat}}(k)$:	$\mathsf{Sign}_{\mathsf{Lat}}(pp, \mathbf{sk}, m)$:	$\mathsf{Verify}_{\mathsf{Lat}}(pp, \mathbf{vk}, m, \sigma)$:
Random Oracle: $\mathsf{H} : \{0,1\}^* \to \{-1,0,1\}^k$ $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{u \times v}$ Return $pp \leftarrow (\mathsf{H}, \mathbf{A})$ $\mathsf{KeyGen}_{\mathsf{Lat}}(pp)$: $\mathbf{sk} \leftarrow_\$ \{-d, 0, d\}^{v \times k}$ $\mathbf{vk} \leftarrow \mathbf{A} \cdot \mathbf{sk}$ Return $(\mathbf{vk}, \mathbf{sk})$	$\mathbf{y} \leftarrow \mathcal{D}_\theta^v$ $\mathbf{c} \leftarrow \mathsf{H}(\mathbf{Ay}, m)$ $\mathbf{z} \leftarrow \mathbf{sk} \cdot \mathbf{c} + \mathbf{y}$ Return $\sigma \leftarrow (\mathbf{c}, \mathbf{z})$	$(\mathbf{c}, \mathbf{z}) \leftarrow \sigma$ If $\lVert \mathbf{z} \rVert \leq \eta\theta\sqrt{v}$ $\wedge\ \mathbf{z} = \mathsf{H}(\mathbf{Az} - \mathbf{vk} \cdot \mathbf{c}, m)$ Return 1, Else, Return 0

Fig. 5. Lattice-based Signatures $\mathsf{SIG}_{\mathsf{Lat}}$

Obviously, the signature scheme $\mathsf{SIG}_{\mathsf{Lat}}$ in Figure 5 has the *simple key generation process* property in which $\mathbf{vk} = \mathbf{A} \cdot \mathbf{sk}$.

Note that the secret key space of the signature scheme $\mathsf{SIG}_{\mathsf{Lat}}$ is $\{-d, 0, d\}^{v \times k}$. We want to construct a fuzzy extractor with extracted string space $\{-d, 0, d\}^{v \times k}$. Lyubashevsky [8] gave parameters choices for the lattice-based signature scheme in which d can be equal to 1 or 31. Here we consider the case $d = 1$.

For $x \in \mathbb{Z}_3^l$, $i \in \mathbb{Z}_3^{n \times l}$, define

$$\mathsf{H}_i(x) := ix,$$

then $\mathcal{H} = \{\mathsf{H}_i : \mathbb{Z}_3^l \to \mathbb{Z}_3^n | i \in \mathbb{Z}_3^{n \times l}\}$ is a family of universal hash functions. Let $n = v \times k$, we can readily interpret a vector in $\mathbb{Z}_3^n$ as a matrix in $\mathbb{Z}_3^{v \times k}$. Given $y \in \mathbb{Z}_3^{v \times k}$, we can get a matrix $y' \in \{-1, 0, 1\}^{v \times k}$ by subtracting 2 from each component of the matrix y. Define the above two operation as $f_1 : \mathbb{Z}_3^n \to \mathbb{Z}_3^{v \times k}$ and $f_2 : \mathbb{Z}_3^{v \times k} \to \{-1, 0, 1\}^{v \times k}$ separately. We can easily get a family of universal hash functions

$$\mathcal{H}' = \{f_2 \circ f_1 \circ \mathsf{H}_i : \mathbb{Z}_3^l \to \{-1, 0, 1\}^{v \times k} | i \in \mathbb{Z}_3^{n \times l}\}. \tag{2}$$

If we instantiate the secure sketch by the syndrome-based secure sketch introduced in Sect. 2.3 and extractor by the universal hash function $\mathcal{H}'$ (Eq. 2), then we get a fuzzy extractor FE_2. The extracted string by FE_2 is uniformly distributed over $\{-1, 0, 1\}^{v \times k}$.

By instantiating the signature SIG in Figure 2 with $\mathsf{SIG}_{\mathsf{Lat}}$ (with proper parameters), the fuzzy extractor FE in Figure 2 with above fuzzy extractor FE_2, we get a fuzzy signature scheme which is $\mathfrak{m}$-EUF-CMA secure in the random oracle model based on the worst-case hardness of the $\widetilde{O}(n^{1.5})$-SIVP problem in general lattices (See Fig. 6).

$\mathsf{Setup}_{\mathsf{F2}}(k)$:	$\mathsf{Sign}_{\mathsf{F2}}(pp, sk', m)$:	$\mathsf{Verify}_{\mathsf{F2}}(pp, vk, m, \sigma)$:
$i \leftarrow_{\$} \mathbb{Z}_3^{n\times l}$, $pp_1 \leftarrow i$ Random Oracle: $\mathsf{H} : \{0,1\}^* \to \{-1,0,1\}^k$ $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{u\times v}$, $pp_2 \leftarrow (\mathsf{H}, \mathbf{A})$ Return $pp \leftarrow (pp_1, pp_2)$ $\mathsf{KeyGen}_{\mathsf{F2}}(pp, sk)$: $(pp_1, pp_2) \leftarrow pp$ $s \leftarrow \mathsf{SS}(sk)$, $P \leftarrow s$ $\widetilde{\mathbf{sk}} \leftarrow \mathsf{Ext}(sk, i)$, $\widetilde{\mathbf{vk}} \leftarrow \mathbf{A} \cdot \widetilde{\mathbf{sk}}$ Return $vk \leftarrow (\widetilde{\mathbf{vk}}, P)$	$(pp_1, pp_2) \leftarrow pp$ $(\widetilde{\mathbf{vk}}, P) \leftarrow vk$ $\widetilde{\mathbf{sk}}' \leftarrow \mathsf{Ext}(\mathsf{Rec}(sk', P), i)$ $\mathbf{y} \leftarrow \mathcal{D}_\theta^v$, $\mathbf{c} \leftarrow \mathsf{H}(\mathbf{A}\mathbf{y}, m)$ $\mathbf{c} \leftarrow \widetilde{\mathbf{sk}}' \cdot \mathbf{c} + \mathbf{y}$ Return $\sigma \leftarrow (\mathbf{c}, \mathbf{z})$	$(\mathbf{c}, \mathbf{z}) \leftarrow \sigma$ $(\widetilde{\mathbf{vk}}, P) \leftarrow vk$ If $\lVert\mathbf{z}\rVert \le \eta\theta\sqrt{v}$ $\wedge\ \mathbf{z} = \mathsf{H}(\mathbf{A}\mathbf{z} - \widetilde{\mathbf{vk}} \cdot \mathbf{c}, m)$ Return 1, Else, Return 0

Fig. 6. Our second instantiation $\mathsf{SIG}_{\mathsf{F2}}$

Corollary 2. *The construction in Figure 6 is an* $\mathfrak{m}$*-EUF-CMA secure fuzzy signature based on the worst-case hardness of the* $\widetilde{O}(n^{1.5})$*-SIVP problem in general lattices in the random oracle model.*

Acknowledgements. We would like to thank the reviewers for their valuable comments. The authors are supported by the Shanghai Sailing Program (Grant No. 21YF1401200) Fundamental Research Funds for the Central Universities (Grant No. 2232020D-34) and Open Fund Program for State Key Laboratory of Information Security of China under Grant (Grant No. 2021-MS-05)

References

[1] Boyen, X.: Reusable cryptographic fuzzy extractors. In: Atluri, V., Pfitzmann, B., McDaniel, P.D. (eds.) Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washington, DC, USA, October 25-29, 2004. pp. 82–91. ACM (2004), https://doi.org/10.1145/1030083.1030096

[2] Boyen, X., Dodis, Y., Katz, J., Ostrovsky, R., Smith, A.D.: Secure remote authentication using biometric data. In: Cramer, R. (ed.) Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3494, pp. 147–163. Springer (2005), https://doi.org/10.1007/11426639_9

[3] Carter, L., Wegman, M.N.: Universal classes of hash functions. J. Comput. Syst. Sci. 18(2), 143–154 (1979), https://doi.org/10.1016/0022-0000(79)90044-8

[4] Connaughton, R., Bowyer, K.W., Flynn, P.J.: Fusion of face and iris biometrics. In: Burge, M.J., Bowyer, K.W. (eds.) Handbook of Iris Recognition, pp. 219–237. Advances in Computer Vision and Pattern Recognition, Springer (2013), https://doi.org/10.1007/978-1-4471-4402-1_12

[5] Cramer, R., Dodis, Y., Fehr, S., Padró, C., Wichs, D.: Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In: Smart, N.P. (ed.) Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings. Lecture Notes in

Computer Science, vol. 4965, pp. 471–488. Springer (2008), https://doi.org/10.1007/978-3-540-78967-3_27

[6] Dodis, Y., Reyzin, L., Smith, A.D.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J. (eds.) Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3027, pp. 523–540. Springer (2004), https://doi.org/10.1007/978-3-540-24676-3_31

[7] Li, N., Guo, F., Mu, Y., Susilo, W., Nepal, S.: Fuzzy extractors for biometric identification. In: Lee, K., Liu, L. (eds.) 37th IEEE International Conference on Distributed Computing Systems, ICDCS 2017, Atlanta, GA, USA, June 5-8, 2017. pp. 667–677. IEEE Computer Society (2017), https://doi.org/10.1109/ICDCS.2017.107

[8] Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7237, pp. 738–755. Springer (2012), https://doi.org/10.1007/978-3-642-29011-4_43

[9] Matsuda, T., Takahashi, K., Murakami, T., Hanaoka, G.: Fuzzy signatures: Relaxing requirements and a new construction. In: Manulis, M., Sadeghi, A., Schneider, S.A. (eds.) Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings. Lecture Notes in Computer Science, vol. 9696, pp. 97–116. Springer (2016), https://doi.org/10.1007/978-3-319-39555-5_6

[10] Nisan, N., Zuckerman, D.: Randomness is linear in space. J. Comput. Syst. Sci. 52(1), 43–52 (1996), https://doi.org/10.1006/jcss.1996.0004

[11] Raghavendra, R., Raja, K.B., Surbiryala, J., Busch, C.: A low-cost multimodal biometric sensor to capture finger vein and fingerprint. In: IEEE International Joint Conference on Biometrics, Clearwater, IJCB 2014, FL, USA, September 29 - October 2, 2014. pp. 1–7. IEEE (2014), https://doi.org/10.1109/BTAS.2014.6996225

[12] Shoup, V.: A computational introduction to number theory and algebra. Cambridge University Press (2006)

[13] 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.) Applied Cryptography and Network Security - 13th International Conference, ACNS 2015, New York, NY, USA, June 2-5, 2015, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9092, pp. 105–126. Springer (2015), https://doi.org/10.1007/978-3-319-28166-7_6

[14] Takahashi, K., Matsuda, T., Murakami, T., Hanaoka, G., Nishigaki, M.: Signature schemes with a fuzzy private key. Int. J. Inf. Sec. 18(5), 581–617 (2019), https://doi.org/10.1007/s10207-019-00428-z

[15] Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3494, pp. 114–127. Springer (2005), https://doi.org/10.1007/11426639_7