A Signature Scheme with a Fuzzy Private Key

Kenta Takahashi[1(✉)], Takahiro Matsuda[2], Takao Murakami[2], Goichiro Hanaoka[2], and Masakatsu Nishigaki[3]

[1] Hitachi, Ltd., Yokohama, Japan
kenta.takahashi.bw@hitachi.com
[2] National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan
{t-matsuda,takao-murakami,hanaoka-goichiro}@aist.go.jp
[3] Shizuoka University, Hamamatsu, Japan
nisigaki@inf.shizuoka.ac.jp

Abstract. In this paper, we introduce a new concept that we call *fuzzy signature*, which is a signature scheme that uses a noisy string such as biometric data as a private key, but *does not require auxiliary data* (which is also called helper string in the context of fuzzy extractors), for generating a signature. Our technical contributions are three-fold: (1) We first give the formal definition of fuzzy signature, together with a formal definition of a "setting" that specifies some necessary information for fuzzy data. (2) We give a generic construction of a fuzzy signature scheme based on a signature scheme with certain homomorphic properties regarding keys and signatures, and a new tool that we call *linear sketch*. (3) We specify a certain setting for fuzzy data, and then give concrete instantiations of these building blocks for our generic construction, leading to our proposed fuzzy signature scheme.

We also discuss how fuzzy signature schemes can be used to realize a biometric-based PKI that uses biometric data itself as a cryptographic key, which we call the *public biometric infrastructure (PBI)*.

Keywords: Fuzzy signature · Public biometric infrastructure

1 Introduction

1.1 Background and Motivation

The public key infrastructure (PKI), which enables authentication and cryptographic communication, plays a significant role as an infrastructure for information security, and is expected to be used for personal use (e.g. national ID, e-government service) more and more widely. In the PKI, private and public keys are generated for each user at the time of registration, and a certificate authority (CA) guarantees the link between the public key and the user's identity by issuing a public key certificate. The user can publish his/her digital signature by using the private signing key. However, since the user has to manage his/her

T. Malkin et al. (Eds.): ACNS 2015, LNCS 9092, pp. 105–126, 2015.
DOI: 10.1007/978-3-319-28166-7_6

private key in a highly secure manner [6], it is not very convenient in some situations. For example, the user is required to possess a hardware token (e.g. smart card, USB token) that contains his/her private key, and memorize a password to activate the key. Such limitations reduce usability, and especially, carrying a dedicated device can be a burden to users. This becomes more serious for elderly people in an aging society.

A feasible approach for solving this problem fundamentally is to use *biometric data* (e.g. fingerprint, iris, and finger-vein) as a cryptographic key. Namely, since biometric data is a part of human body, it can offer a more usable way to link the private key and the individual. Moreover, a multibiometric sensor that simultaneously acquires multiple biometric information (e.g. iris and face [1]; fingerprint and finger-vein [15]) has been recently developed to obtain enough entropy at one time, and we can also expect that longer strings will be produced from various biometric data in the near future.

However, since biometric data is noisy and fluctuates each time it is captured, it cannot be used directly as a key. Intuitively, it seems that this issue can be immediately solved by using a *fuzzy extractor* [4], but this is not always the case. More specifically, for extracting a string by a fuzzy extractor, an auxiliary data called a helper string is necessary, and therefore, the user is still enforced to carry a dedicated device that stores it. (We discuss the limitations of the approaches with helper data (i.e. the fuzzy-extractor-based approaches) in more detail in Appendix A.) Hence, it is considered that the above problem cannot be straightforwardly solved by using the fuzzy extractor, and another cryptographic technique by which noisy data can be used as a cryptographic private key without relying on any auxiliary data, is necessary.

Fuzzy Signature: A Signature Scheme with a Fuzzy Private Key. In this paper, we introduce a new concept of digital signature that we call *fuzzy signature*. Consider an ordinary digital signature scheme. The signing algorithm Sign is defined as a function that takes a signing key sk and a message m as input, and outputs a signature $\sigma \leftarrow \mathsf{Sign}(sk, m)$[1]. Thus, it is natural to consider that its "fuzzy" version Sign should be defined as a function that takes a noisy string x and a message m as input, and outputs $\sigma \leftarrow \mathsf{Sign}(x, m)$. In this paper, we refer to such digital signature (i.e. digital signature that allows to use a noisy string itself as a signing key) as *fuzzy signature*. It should be noted that some studies proposed a fuzzy identity based signature (FIBS) scheme [7,20,21,23,24], which uses a noisy string as a verification key. However, fuzzy signature is a totally different concept since it does not allow a fuzzy verification key, but allows a *fuzzy signing key* (i.e. *fuzzy private key*).

Figure 1 shows the architecture of fuzzy signature in the left, and that of digital signature using a fuzzy extractor in the right. In fuzzy signature, the key generation algorithm $\mathsf{KG}_{\mathsf{FS}}$ takes a noisy string (e.g. biometric feature) x as input, and outputs a verification key vk.; The signing algorithm $\mathsf{Sign}_{\mathsf{FS}}$ takes

[1] Strictly speaking, in this paper we adopt the syntax in which Sign also takes a public parameter as input (see Sect. 2.2). In this section, we omit it for simplicity.

another noisy string x' and a message m as input, and outputs a signature σ.; The verification algorithm $\mathsf{Ver}_{\mathsf{FS}}$ takes vk, m, and σ as input, and verifies whether σ is valid or not. If x' is close to x, σ is verified as valid (the formal definitions of these algorithms are given in Sect. 3). We emphasize that a fuzzy signature scheme cannot be constructed based on a fuzzy extractor, since it requires a helper string P along with a noisy string x' to make a signature σ on a message m. Hence, to date, the realization of fuzzy signature has been an open problem.

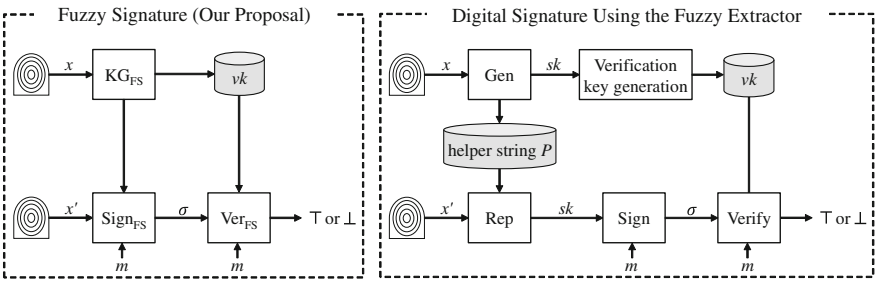

Fig. 1. Architecture of fuzzy signature (our proposal) (left), and that of digital signature using a fuzzy extractor (right) (x, x': noisy string, sk: signing key, vk: verification key, σ: signature, m: message, $\top$: valid, $\bot$: invalid).

1.2 Our Contributions

In this paper, we show that under the assumption that a noisy string is uniform and has enough entropy, a secure fuzzy signature scheme can be indeed realized. More specifically, our technical contributions are three-fold:

1. **Formal Definition of Fuzzy Signature (Sect. 3):** We first formalize a *fuzzy key setting* that specifies some necessary information for fuzzy data (e.g. a metric space to which fuzzy data belongs, and a distribution of fuzzy data over it, etc.). We then give a formal definition of a fuzzy signature scheme that is associated with a fuzzy key setting.
2. **Generic Construction (Sect. 4):** In order to better understand our ideas and the security arguments for our proposed scheme clearly and in a modular manner, we give a generic construction of a fuzzy signature from an ordinary signature scheme with certain homomorphic properties regarding keys and signatures (which is formally defined in Sect. 2.2), and a new technical tool that we call *linear sketch* that incorporates a kind of encoding and error correction processes. (We explain how it works and is used informally in Sect. 1.3, and give a formal definition in Sect. 4.1.)
3. **Concrete Instantiation (Sect. 5):** We specify a concrete fuzzy key setting in which fuzzy data is distributed uniformly over some metric space, and then show how to realize the underlying signature scheme and a linear sketch scheme that can be used in the generic construction for this fuzzy key setting.

Our signature scheme is based on the Waters signature scheme [22], which we modify so that it satisfies the homomorphic property required in our generic construction. Our linear sketch scheme is based on the Chinese reminder theorem and some form of linear coding and error correction.

In Sect. 1.3, we give an overview of how our proposed fuzzy signature scheme is constructed.

It is expected that our fuzzy signature scheme can be used to realize a biometric-based PKI that uses biometric data itself as a cryptographic key, which we call the *public biometric infrastructure (PBI)*. We discuss it in Sect. 6 in more detail. We would like to emphasize that although so far we have mentioned biometric feature as a main example of noisy data, our scheme is not restricted to it, and can also use other noisy data such as the output of a PUF (physically unclonable function) [12] as input, as long as it satisfies the requirement of a fuzzy key setting.

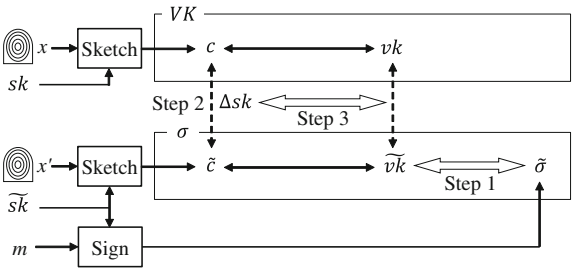

Fig. 2. An overview of our generic construction of a fuzzy signature scheme. The box "Sketch" indicates one of the algorithms of a primitive that we call "linear sketch," which is formalized in Sect. 4.1.

1.3 Overview of Our Fuzzy Signature Scheme

Our proposed fuzzy signature scheme Σ_{FS} is constructed based on an ordinary signature scheme (let us call it the "underlying scheme" Σ for the explanation here). In Fig. 2, we illustrate an overview of our construction of a fuzzy signature scheme. Our basic strategy is as follows: In the signing algorithm $\mathsf{Sign}_{\mathsf{FS}}(x', m)$ (where x' is a noisy string and m is a message to be signed), we do not extract a signing key sk (for the underlying scheme Σ) directly from x' (which is the idea of the fuzzy-extractor-based approach), but use a randomly generated key pair $(\widetilde{vk}, \widetilde{sk})$ of Σ, generate a signature $\widetilde{\sigma}$ using $\widetilde{sk}$, and also create a "sketch" $\widetilde{c}$ (via the algorithm denoted by "Sketch" in Fig. 2), which is a kind of "one-time pad" ciphertext of the signing key $\widetilde{sk}$ using x' as a "one-time pad key"[2], and let

[2] The procedure "Sketch" is actually not the one-time pad encryption, but more like a (one-way) "encoding," because we do not need to decrypt $\widetilde{c}$ to recover $\widetilde{sk}$. This is the main reason why we call $\widetilde{c}$ "sketch" (something that contains the information of $\widetilde{sk}$), not "ciphertext".

a signature σ consist of $(\widetilde{vk}, \widetilde{\sigma}, \widetilde{c})$. This enables us to generate a fresh signature $\widetilde{\sigma}$ without being worried about the fuzziness of x'. Here, however, since $\widetilde{\sigma}$ is a valid signature only under $\widetilde{vk}$, in order to generate a signature next time, we need to somehow carry the "encrypted" signing key $\widetilde{c}$. To avoid it, in the key generation algorithm $\mathsf{KG}_{\mathsf{FS}}(x)$ (where x is also a noisy string measured at the key generation), we also generate a "sketch" c of another fresh signing key sk using x as the "one-time pad key", and put it as a part of a verification key of our fuzzy signature scheme. Hence, a verification key VK in our fuzzy signature scheme Σ_{FS} consists of a verification key vk (corresponding to the signing key sk generated at the key generation) of the underlying scheme Σ, and the sketch c generated from sk and x. Here, by using some kind of error correction method with which we can remove "noise" from c and $\widetilde{c}$, and comparing them, we can calculate the "difference" Δsk between sk and $\widetilde{sk}$, similarly to what we can do for one-time pad ciphertexts.[3] Thus, if the underlying scheme Σ has the property that "given two verification keys $(vk, \widetilde{vk})$ and a (candidate) difference Δsk, one can verify that the difference between the secret keys sk and $\widetilde{sk}$ (corresponding to vk and $\widetilde{vk}$, respectively) is indeed Δsk", we can verify the signature $\sigma = (\widetilde{vk}, \widetilde{\sigma}, \widetilde{c})$ of Σ_{FS} under the verification key $VK = (vk, c)$ by first checking the validity of $\widetilde{\sigma}$ under $\widetilde{vk}$ (Step 1), then recovering Δsk from c and $\widetilde{c}$ (Step 2), and finally checking whether the difference between vk and $\widetilde{vk}$ indeed corresponds to Δsk (Step 3). The explanation so far is exactly what we do in our generic construction in Sect. 4.

To concretely realize the above strategy, we propose a variant of the Waters signature scheme [22] (which we call *modified Waters signature* (MWS)) that satisfies all our requirements. We also formalize the methods for "one-time padding secret keys (sk and $\widetilde{sk}$) by noisy strings" and "reconstructing the difference between two secret keys", as a tool that we call *linear sketch*, and show how to realize a linear sketch scheme that can be used together with the MWS scheme to realize our fuzzy signature scheme Σ_{FS}.

2 Preliminaries

In this section, we review the basic notation and the definitions of primitives.

Basic Notation. $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{R}$ denote the sets of all natural numbers, all integers, and all real numbers, respectively. If $n \in \mathbb{N}$, then we define $[n] := \{1, \ldots, n\}$. If $a, b \in \mathbb{N}$, then "$\mathtt{GCD}(a, b)$" denotes the greatest common divisor of a and b, and if $a \in \mathbb{R}$, then "$\lfloor a \rfloor$" denotes the maximum integer which does not exceed a.

[3] Recall that the original one-time pad encryption $c = m \oplus K$ (where c, m, and K are a ciphertext, a message, and a key, respectively) has "linearity" in the sense that given two ciphertexts $c_1 = m \oplus K_1$ and $c_2 = m \oplus K_2$ of the same message m under different keys K_1 and K_2, we can calculate the difference $\Delta K = K_1 \oplus K_2$ by computing $c_1 \oplus c_2$.

If S is a finite set, then "$|S|$" denotes its size, and "$x \leftarrow_{\mathtt{R}} S$" denotes that x is chosen uniformly at random from S. If Φ is a distribution (over some set), then $x \leftarrow_{\mathtt{R}} \Phi$ denotes that x is chosen according to the distribution Φ. "$x \leftarrow y$" denotes that y is (deterministically) assigned to x. If x and y are bit-strings, then $|x|$ denotes the bit-length of x, and "$(x||y)$" denotes the concatenation of x and y. "(P)PTA" denotes a *(probabilistic) polynomial time algorithm.*

If $\mathcal{A}$ is a probabilistic algorithm, then "$y \leftarrow_{\mathtt{R}} \mathcal{A}(x)$" denote that $\mathcal{A}$ computes y by taking x as input and using an internal randomness that is chosen uniformly at random, and if we need to specify the randomness, we denote by "$y \leftarrow \mathcal{A}(x; r)$" (in which case the computation of $\mathcal{A}$ is deterministic that takes x and r as input). If furthermore $\mathcal{O}$ is a (possibly probabilistic) algorithm or a function, then "$\mathcal{A}^{\mathcal{O}}$" denotes that $\mathcal{A}$ has oracle access to $\mathcal{O}$. Throughout the paper, "k" denotes a security parameter. A function $f(\cdot) : \mathbb{N} \to [0, 1]$ is said to be *negligible* if for all positive polynomials $p(\cdot)$ and all sufficiently large k, we have $f(k) < 1/p(k)$.

2.1 Bilinear Groups and Computational Problems

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 $e(g'^a, g'^b) = e(g', g')^{ab}$, and (*Non-degeneracy*:) for all generators g' of $\mathbb{G}$, $e(g', g')$ 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 $\mathcal{BG}$.

Definition 1. *We say that the computational Diffie-Hellman (CDH) assumption holds with respect to* BGGen *if for all PPTAs* $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{CDH}}_{\mathsf{BGGen},\mathcal{A}}(k) := \Pr[\mathcal{BG} \leftarrow \mathsf{BGGen}(1^k); a, b \leftarrow_{\mathtt{R}} \mathbb{Z}_p : \mathcal{A}(\mathcal{BG}, g^a, g^b) = g^{ab}]$ *is negligible.*

2.2 Signature

Syntax and Correctness. We model a signature scheme Σ as a quadruple of the PPTAs $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$ that are defined as follows: The setup algorithm Setup takes 1^k as input, and outputs a public parameter pp.; The key generation algorithm KG takes pp as input, and output a verification/signing key pair (vk, sk).; The signing algorithm Sign takes pp, sk, and a message m as input, and outputs a signature σ.; The verification algorithm Ver takes pp, vk, m, and σ as input, and outputs either $\top$ or $\bot$. Here, "$\top$" (resp. "$\bot$") indicates that σ is a valid (resp. invalid) signature of the message m under the key vk.

We require for all $k \in \mathbb{N}$, all pp output by $\mathsf{Setup}(1^k)$, all (vk, sk) output by $\mathsf{KG}(pp)$, and all messages m, we have $\mathsf{Ver}(pp, vk, m, \mathsf{Sign}(pp, sk, m)) = \top$.

`EUF-CMA` *Security.* Here, we recall the definition of *existential unforgeability against chosen message attacks* (`EUF-CMA` security).

For a signature scheme $\Sigma = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$ and an adversary $\mathcal{A}$, consider the following EUF-CMA experiment $\mathsf{Expt}^{\text{EUF-CMA}}_{\Sigma,\mathcal{A}}(k)$:

$$\begin{aligned}\mathsf{Expt}^{\text{EUF-CMA}}_{\Sigma,\mathcal{A}}(k) : [\ pp \leftarrow_{\mathrm{R}} \mathsf{Setup}(1^k);\ (vk, sk) \leftarrow_{\mathrm{R}} \mathsf{KG}(pp);\\ \mathcal{Q} \leftarrow \emptyset;\ (m', \sigma') \leftarrow_{\mathrm{R}} \mathcal{A}^{\mathcal{O}_{\mathsf{Sign}}(\cdot)}(pp, vk);\\ \text{If } m' \notin \mathcal{Q} \wedge \mathsf{Ver}(pp, vk, m', \sigma') = \top \text{ then return 1 else return 0 }],\end{aligned}$$

where $\mathcal{O}_{\mathsf{Sign}}$ is the signing oracle which takes a message m as input, updates $\mathcal{Q}$ by $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$, and returns a signature $\sigma \leftarrow_{\mathrm{R}} \mathsf{Sign}(pp, sk, m)$.

Definition 2. *We say that a signature scheme Σ is* EUF-CMA *secure if for all PPTA adversaries $\mathcal{A}$, $\mathsf{Adv}^{\text{EUF-CMA}}_{\Sigma,\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\text{EUF-CMA}}_{\Sigma,\mathcal{A}}(k) = 1]$ is negligible.*

Homomorphic Properties of Keys and Signatures. For our fuzzy signature scheme, we will utilize a signature scheme that has certain homomorphic properties regarding keys and signatures, and thus we formalize the properties here.

Definition 3. *Let $\Sigma = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme. We say that Σ is* homomorphic *if it satisfies the following properties:*

- *For all parameters pp output by Setup, the signing key space constitutes a cyclic abelian group $(\mathcal{K}_{pp}, +)$, and the key generation algorithm KG can be described by using the deterministic PTA KG' as follows:*

$$\mathsf{KG}(pp) : [sk \leftarrow_{\mathrm{R}} \mathcal{K}_{pp};\ vk \leftarrow \mathsf{KG}'(pp, sk);\ \textit{Return } (vk, sk).]. \tag{1}$$

- *There exists a deterministic PTA M_{vk} that takes a public parameter pp (output by Setup), a verification key vk (output by $\mathsf{KG}(pp)$), and a "shift" $\Delta sk \in \mathcal{K}_{pp}$ as input, and outputs the "shifted" verification key vk'.*
  We require that for all pp output by Setup and all $sk, \Delta sk \in \mathcal{K}_{pp}$, it holds that

$$\mathsf{KG}'(pp, sk + \Delta sk) = \mathsf{M}_{\mathsf{vk}}(pp, \mathsf{KG}'(pp, sk), \Delta sk). \tag{2}$$

- *There exists a deterministic PTA $\mathsf{M}_{\mathsf{sig}}$ that takes a public parameter pp (output by Setup), a verification key vk (output by $\mathsf{KG}(pp)$), a message m, a signature σ, and a "shift" $\Delta sk \in \mathcal{K}_{pp}$ as input, and outputs a "shifted" signature σ'.*
  We require that for all pp output by Setup, all messages m, all $sk, \Delta sk \in \mathcal{K}_{pp}$, the following two distributions are identical:

$$\begin{aligned}&\{\sigma' \leftarrow_{\mathrm{R}} \mathsf{Sign}(pp, sk + \Delta sk, m) : \sigma'\}, \quad \textit{and}\\ &\{\sigma \leftarrow_{\mathrm{R}} \mathsf{Sign}(pp, sk, m);\ \sigma' \leftarrow \mathsf{M}_{\mathsf{sig}}(pp, \mathsf{KG}'(pp, sk), m, \sigma, \Delta sk) : \sigma'\}.\end{aligned} \tag{3}$$

  Furthermore, we require that for all pp output by Setup, all $sk, \Delta sk \in \mathcal{K}_{pp}$, and all (m, σ) satisfying $vk = \mathsf{KG}'(pp, sk)$ and $\mathsf{Ver}(pp, vk, m, \sigma) = \top$, it holds that

$$\mathsf{Ver}(pp, \mathsf{M}_{\mathsf{vk}}(pp, vk, \Delta sk), m, \mathsf{M}_{\mathsf{sig}}(pp, vk, m, \sigma, \Delta sk)) = \top. \tag{4}$$

On "Weak" Distributions of Signing Keys. Let $\Sigma = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme with the homomorphic property (as per Definition 3) with secret key space $\mathcal{K}_{pp}$ for a public parameter pp, and thus there exists the algorithm KG' such that KG can be written as in Eq. (1). Let $u : \mathbb{N} \to \mathbb{N}$ be any function. For an $\mathtt{EUF\text{-}CMA}$ adversary $\mathcal{A}$ attacking Σ, let $\widetilde{\mathsf{Adv}}^{\mathtt{EUF\text{-}CMA}}_{\Sigma,\mathcal{A}}(k)$ be the advantange of $\mathcal{A}$ in the experiment that is the same as $\mathsf{Expt}^{\mathtt{EUF\text{-}CMA}}_{\Sigma,\mathcal{A}}(k)$, except that a secret key sk is chosen by $sk \leftarrow_{\mathtt{R}} \widetilde{\mathcal{K}}_{pp}$ (instead of $sk \leftarrow_{\mathtt{R}} \mathcal{K}_{pp}$) where $\widetilde{\mathcal{K}}_{pp}$ denotes an arbitrary (non-empty) subset of $\mathcal{K}_{pp}$ satisfying $|\mathcal{K}_{pp}|/|\widetilde{\mathcal{K}}_{pp}| \leq u(k)$.

We will use the following fact, which is obtained as a corollary of the lemma shown by Dodis and Yu [5, Lemma 1].

Lemma 1. ***(Corollary of*** [5, Lemma 1]**)** *Under the above setting, for any PPTA adversary $\mathcal{A}$, it holds that $\widetilde{\mathsf{Adv}}^{\mathtt{EUF\text{-}CMA}}_{\Sigma,\mathcal{A}}(k) \leq u(k) \cdot \mathsf{Adv}^{\mathtt{EUF\text{-}CMA}}_{\Sigma,\mathcal{A}}(k)$.*

Waters Signature Scheme. Our fuzzy signature scheme is based on the Waters signature scheme [22], and thus we recall it here. (We consider the version where the setup and the key generation (for each user) is separated.)

Let $\ell = \ell(k)$ be a positive polynomial, and let BGGen be a bilinear group generator (as defined in Sect. 2.1). Then, the Waters signature scheme Σ_{Wat} for ℓ-bit messages are constructed as in Fig. 3. Σ_{Wat} is known to be $\mathtt{EUF\text{-}CMA}$ secure if the CDH assumption holds with respect to BGGen [22].

$\mathsf{Setup}_{\mathsf{Wat}}(1^k)$:	$\mathsf{Sign}_{\mathsf{Wat}}(pp, sk, m)$:
$\mathcal{BG} := (p, \mathbb{G}, \mathbb{G}_T, g, e) \leftarrow \mathsf{BGGen}(1^k)$	Parse m as $(m_1 \| \ldots \| m_\ell) \in \{0,1\}^\ell$.
$h, u', u_1, \ldots, u_\ell \leftarrow_{\mathtt{R}} \mathbb{G}$	$r \leftarrow_{\mathtt{R}} \mathbb{Z}_p$
$pp \leftarrow (\mathcal{BG}, h, u', (u_i)_{i\in[\ell]})$	$\sigma_1 \leftarrow h^{sk} \cdot (u' \prod_{i\in[\ell]} u_i^{m_i})^r$; $\sigma_2 \leftarrow g^r$
Return pp.	Return $\sigma \leftarrow (\sigma_1, \sigma_2)$.
$\mathsf{KG}_{\mathsf{Wat}}(pp)$:	$\mathsf{Ver}_{\mathsf{Wat}}(pp, vk, m, \sigma)$:
$sk \leftarrow_{\mathtt{R}} \mathbb{Z}_p$	$(\sigma_1, \sigma_2) \leftarrow \sigma$
$vk \leftarrow g^{sk}$	Parse m as $(m_1 \| \ldots \| m_\ell) \in \{0,1\}^\ell$.
Return (vk, sk).	If $e(\sigma_2, u' \cdot \prod_{i\in[\ell]} u_i^{m_i}) \cdot e(vk, h) = e(\sigma_1, g)$ then return $\top$ else return $\bot$.

Fig. 3. The Waters signature scheme Σ_{Wat} [22].

3 Definitions for Fuzzy Signature

In this section, we introduce the definitions of Fuzzy Signature (FS).

As mentioned in Sect. 1, to define FS, we need to define some "setting" that models a space to which a fuzzy data (used as a signing key of FS) belongs, a distribution from which fuzzy data is sampled, etc. We therefore first formalize it as a *fuzzy key setting* in Sect. 3.1, and then define FS that is associated with a fuzzy key setting in Sect. 3.2.

3.1 Formalization of Fuzzy Key Setting

Consider a typical biometric authentication scheme, in which a "fuzzy" biometric feature $x \in X$ (where X is some metric space) is measured and extracted from a user at the registration phase.; At the authentication phase, a biometric feature x' is measured and extracted from a (possibly different) user, and this user is considered the user who generated the biometric data x and thus authentic if x and x' are sufficiently "close" according to some metric.

We abstract out this typical setting for "identifying fuzzy objects" as a "fuzzy key setting", and formalize it here. Roughly, a fuzzy key setting specifies (1) the metric space to which fuzzy data (such as biometric data) belongs (X in the above example), (2) the distribution of fuzzy data sampled at the "registration phase" (x in the above example), and (3) the error distribution that models "fuzziness" of the fuzzy data (the relationship between x and x' in the above example).

We adopt what we call the "universal error model", which assumes that for all objects U that produce fuzzy data that we are interested in, if U produces a data x at the first measurement (say, at the registration phase), if the same object is measured next time, then the measured data x' follows the distribution $\{e \leftarrow_{\mathtt{R}} \Phi; x' \leftarrow x + e : x'\}$. That is, the error distribution Φ is independent of individual U. (We also assume that the metric space constitutes an abelian group so that addition is well-defined.)

Formally, a fuzzy key setting $\mathcal{F}$ consists of $((\mathsf{d}, X), t, \mathcal{X}, \Phi, \epsilon)$, each of which is defined as follows:

(d, X): This is a metric space, where X is a space to which a possible fuzzy data x belongs, and $\mathsf{d} : X^2 \to \mathbb{R}$ is the corresponding distance function. We furthermore assume that X constitutes an abelian group.

t: ($\in \mathbb{R}$) This is the threshold value, determined by a security parameter k. Based on t, the false acceptance rate (FAR) and the false rejection rate (FRR) are determined. We require that the $\mathtt{FAR} := \Pr[x, x' \leftarrow_{\mathtt{R}} \mathcal{X} : \mathsf{d}(x, x') < t]$ is negligible in k.

$\mathcal{X}$: This is a distribution of fuzzy data over X.

Φ: This is an error distribution (see the above explanation).

ϵ: ($\in [0, 1]$) This is an error parameter that represents FRR. We require that for all $x \in X$, $\mathtt{FRR} := \Pr[e \leftarrow_{\mathtt{R}} \Phi : \mathsf{d}(x, x + e) \geq t] \leq \epsilon$.

3.2 Fuzzy Signature

A fuzzy signature scheme Σ_{FS} for a fuzzy key setting $\mathcal{F} = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \epsilon)$ consists of the four algorithms $(\mathsf{Setup}_{\mathsf{FS}}, \mathsf{KG}_{\mathsf{FS}}, \mathsf{Sign}_{\mathsf{FS}}, \mathsf{Ver}_{\mathsf{FS}})$:

$\mathsf{Setup}_{\mathsf{FS}}$: This is the setup algorithm that takes the description of the fuzzy key setting $\mathcal{F}$ and 1^k as input (where k determines the threshold value t of $\mathcal{F}$), and outputs a public parameter pp.

$\mathsf{KG}_{\mathsf{FS}}$: This is the key generation algorithm that takes pp and a fuzzy data $x \in X$ as input, and outputs a verification key vk.

$\mathsf{Sign}_{\mathsf{FS}}$: This is the signing algorithm that takes pp, a fuzzy data $x' \in X$, and a message m as input, and outputs a signature σ.

$\mathsf{Ver}_{\mathsf{FS}}$: This is the (deterministic) verification algorithm that takes pp, vk, m, and σ as input, and outputs either $\top$ ("accept") or $\bot$ ("reject").

Correctness. We require a natural correctness requirement: For all $k \in \mathbb{N}$, all pp output by $\mathsf{Setup}_{\mathsf{FS}}(\mathcal{F}, 1^k)$, all $x, x' \in X$ such that $\mathsf{d}(x, x') < t$, and all messages m, it holds that $\mathsf{Ver}_{\mathsf{FS}}(pp, \mathsf{KG}_{\mathsf{FS}}(pp, x), m, \mathsf{Sign}_{\mathsf{FS}}(pp, x', m)) = \top$.

EUF-CMA *Security.* For a fuzzy signature scheme, we consider EUF-CMA security in a similar manner to that for an ordinary signature scheme, reflecting the universal error model of a fuzzy key setting.

For a fuzzy signature scheme Σ_{FS} for a fuzzy key setting $\mathcal{F} = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \epsilon)$ and an adversary $\mathcal{A}$, consider the following experiment $\mathsf{Expt}^{\text{EUF-CMA}}_{\Sigma_{\mathsf{FS}}, \mathcal{F}, \mathcal{A}}(k)$:

$$\mathsf{Expt}^{\text{EUF-CMA}}_{\Sigma_{\mathsf{FS}}, \mathcal{F}, \mathcal{A}}(k) : [\ pp \leftarrow_{\mathrm{R}} \mathsf{Setup}_{\mathsf{FS}}(\mathcal{F}, 1^k);\ x \leftarrow_{\mathrm{R}} \mathcal{X};\ vk \leftarrow \mathsf{KG}_{\mathsf{FS}}(pp, x);$$
$$\mathcal{Q} \leftarrow \emptyset;\ (m', \sigma') \leftarrow_{\mathrm{R}} \mathcal{A}^{\mathcal{O}_{\mathsf{Sign}_{\mathsf{FS}}}(\cdot)}(pp, vk) :$$
$$\text{If } m' \notin \mathcal{Q} \wedge \mathsf{Ver}_{\mathsf{FS}}(pp, vk, m', \sigma') = \top \text{ then return 1 else return 0 }],$$

where $\mathcal{O}_{\mathsf{Sign}_{\mathsf{FS}}}$ is the signing oracle that takes a message m as input, and operates as follows: It updates $\mathcal{Q}$ by $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$, samples $e \leftarrow_{\mathrm{R}} \Phi$, computes a signature $\sigma \leftarrow_{\mathrm{R}} \mathsf{Sign}_{\mathsf{FS}}(pp, x + e, m)$, and returns σ.

Definition 4. *We say that a fuzzy signature scheme Σ_{FS} is* EUF-CMA *secure if for all PPTA adversaries $\mathcal{A}$, $\mathsf{Adv}^{\text{EUF-CMA}}_{\Sigma_{\mathsf{FS}}, \mathcal{F}, \mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\text{EUF-CMA}}_{\Sigma_{\mathsf{FS}}, \mathcal{F}, \mathcal{A}}(k) = 1]$ is negligible.*

4 Generic Construction

In this section, we show a generic construction for a fuzzy signature scheme. This construction is based on a new tool that we call *linear sketch* and a signature scheme with the homomorphic property (as per Definition 3). We introduce a *linear sketch* scheme in Sect. 4.1, and then in Sect. 4.2, we show the generic construction.

4.1 Linear Sketch

Definition 5. *Let $\mathcal{F} = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \epsilon)$ be a fuzzy key setting. We say that a pair of deterministic PTAs $\mathcal{S} = (\mathsf{Sketch}, \mathsf{DiffRec})$ is a* linear sketch *scheme for $\mathcal{F}$, if it satisfies the following three properties:*

Syntax and Correctness: Sketch *is the "sketching" algorithm that takes the description Λ of an abelian group $(\mathcal{K}, +)$, an element $s \in \mathcal{K}$, and a fuzzy data $x \in X$ as input, and outputs a "sketch" c.;* $\mathsf{DiffRec}$ *is the "difference reconstruction" algorithm that takes Λ and two values c, c' (supposedly output by* Sketch*) as input, and outputs the "difference" $\Delta s \in \mathcal{K}$.*
It is required that for all $x, x' \in X$ such that $\mathsf{d}(x, x') < t$, and for all $s, \Delta s \in \mathcal{K}$, it holds that

$$\mathsf{DiffRec}(\Lambda, \mathsf{Sketch}(\Lambda, s, x), \mathsf{Sketch}(\Lambda, s + \Delta s, x')) = \Delta s. \tag{5}$$

Linearity: *There exists a deterministic PTA* $\mathsf{M_c}$ *satisfying the following: For all* $x, e \in X$ *such that* $\mathsf{d}(x, x+e) < t$*, and for all* $s, \Delta s \in \mathcal{K}$*, it holds that*

$$\mathsf{Sketch}(\Lambda, s+\Delta s, x+e) = \mathsf{M_c}(\Lambda, \mathsf{Sketch}(\Lambda, s, x), \Delta s, e). \tag{6}$$

Simulatability: *There exists a PPTA* Sim *such that for all* $s \in \mathcal{K}$*, the following two distributions are statistically indistinguishable (in the security parameter* k *that is associated with* t *in* $\mathcal{F}$*):*

$$\{x \leftarrow_{\mathrm{R}} \mathcal{X};\ c \leftarrow \mathsf{Sketch}(\Lambda, s, x) : c\} \quad \textit{and} \quad \{c \leftarrow_{\mathrm{R}} \mathsf{Sim}(\Lambda) : c\}. \tag{7}$$

4.2 Generic Construction

Let $\mathcal{F} = ((d, X), t, \mathcal{X}, \Phi, \epsilon)$ be a fuzzy key setting, and let $\Sigma = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme. We assume that Σ has the homomorphic property (Definition 3), namely, its secret key space (given pp) is a cyclic abelian group $(\mathcal{K}_{pp}, +)$, and has the additional algorithms KG', $\mathsf{M_{vk}}$, and $\mathsf{M_{sig}}$. Let $\mathcal{S} = (\mathsf{Sketch}, \mathsf{DiffRec})$ be a linear sketch scheme for $\mathcal{F}$. Using Σ and $\mathcal{S}$, we construct a fuzzy signature scheme $\Sigma_{\mathsf{FS}} = (\mathsf{Setup_{FS}}, \mathsf{KG_{FS}}, \mathsf{Sign_{FS}}, \mathsf{Ver_{FS}})$ for the fuzzy key setting $\mathcal{F}$ as in Fig. 4.

$\mathsf{Setup_{FS}}(\mathcal{F}, 1^k)$:	$\mathsf{Sign_{FS}}(pp, x', m)$:	$\mathsf{Ver_{FS}}(pp, VK, m, \sigma)$:
$pp_s \leftarrow_{\mathrm{R}} \mathsf{Setup}(1^k)$	$(pp_s, \Lambda) \leftarrow pp$	$(pp_s, \Lambda) \leftarrow pp$
Let $\Lambda := (\mathcal{K}_{pp_s}, +)$.	$\widetilde{sk} \leftarrow_{\mathrm{R}} \mathcal{K}_{pp_s}$	$(vk, c) \leftarrow VK$
Return $pp \leftarrow (pp_s, \Lambda)$.	$\widetilde{vk} \leftarrow \mathsf{KG}'(pp_s, \widetilde{sk})$	$(\widetilde{vk}, \widetilde{\sigma}, \widetilde{c}) \leftarrow \sigma$
$\mathsf{KG_{FS}}(pp, x)$:	$\widetilde{\sigma} \leftarrow_{\mathrm{R}} \mathsf{Sign}(pp_s, \widetilde{sk}, m)$	If $\mathsf{Ver}(pp_s, \widetilde{vk}, m, \widetilde{\sigma}) = \perp$ then return $\perp$.
$(pp_s, \Lambda) \leftarrow pp$	$\widetilde{c} \leftarrow \mathsf{Sketch}(\Lambda, \widetilde{sk}, x')$	$\Delta sk \leftarrow \mathsf{DiffRec}(\Lambda, c, \widetilde{c})$
$sk \leftarrow_{\mathrm{R}} \mathcal{K}_{pp_s}$	Return $\sigma \leftarrow (\widetilde{vk}, \widetilde{\sigma}, \widetilde{c})$.	If $\mathsf{M_{vk}}(pp_s, vk, \Delta sk) = \widetilde{vk}$ then return $\top$ else return $\perp$.
$vk \leftarrow \mathsf{KG}'(pp_s, sk)$		
$c \leftarrow \mathsf{Sketch}(\Lambda, sk, x)$		
Return $VK \leftarrow (vk, c)$.		

Fig. 4. A generic construction of a fuzzy signature scheme Σ_{FS} for a fuzzy key setting $\mathcal{F}$ based on a signature scheme Σ with the homomorphic property and a linear sketch scheme $\mathcal{S}$ for $\mathcal{F}$.

The security of the fuzzy signature scheme Σ_{FS} is guaranteed as follows.

Theorem 1. *If* Σ *is* EUF-CMA *secure and* $\mathcal{S}$ *is a linear sketch scheme, then the fuzzy signature scheme* Σ_{FS} *is* EUF-CMA *secure.*

Proof Sketch of Theorem 1. The formal proof of Theorem 1 is given in the full version due to the lack of space, and here we give an overview of the proof.

Let $\mathcal{A}$ be any PPTA adversary that attacks the EUF-CMA security of the fuzzy signature scheme Σ_{FS}. Note that in the original EUF-CMA experiment $\mathsf{Expt}^{\text{EUF-CMA}}_{\Sigma_{\mathsf{FS}}, \mathcal{F}, \mathcal{A}}(k)$, the verification key VK is generated as follows:

$$[x \leftarrow_{\mathrm{R}} \mathcal{X};\ sk \leftarrow_{\mathrm{R}} \mathcal{K}_{pp_s};\ vk \leftarrow \mathsf{KG}'(pp_s, sk);\ \underline{c \leftarrow \mathsf{Sketch}(\Lambda, sk, x)};\ VK \leftarrow (vk, c)].$$

Then, consider a "simulated process" for generating VK, which is the same as above except that the step with the underline is replaced with "$c \leftarrow_{\mathrm{R}} \mathsf{Sim}(\Lambda)$". Then, by the simulatability of the linear sketch scheme $\mathcal{S}$, the distribution of VK generated in the original process and that of the simulated process are statistically indistinguishable.

Furthermore, note also that the signing oracle $\mathcal{O}_{\mathsf{Sign}_{\mathsf{FS}}}(m)$ in the original EUF-CMA experiment $\mathsf{Expt}^{\text{EUF-CMA}}_{\Sigma_{\mathsf{FS}},\mathcal{F},\mathcal{A}}$ generates a signature σ as follows:

$$[e \leftarrow_{\mathrm{R}} \Phi;\ \widetilde{sk} \leftarrow_{\mathrm{R}} \mathcal{K}_{pp_s};\ \widetilde{vk} \leftarrow \mathsf{KG}'(pp_s, \widetilde{sk});\ \widetilde{\sigma} \leftarrow_{\mathrm{R}} \mathsf{Sign}(pp_s, \widetilde{sk}, m);\ \widetilde{c} \leftarrow \mathsf{Sketch}(\Lambda, \widetilde{sk}, x+e);\ \sigma \leftarrow (\widetilde{vk}, \widetilde{\sigma}, \widetilde{c})].$$

By the homomorphic property of the underlying signature scheme Σ, and the linearity property of the linear sketch scheme $\mathcal{S}$, the following process generates a signature σ whose distribution is exactly the same as σ generated as above.

$$[e \leftarrow_{\mathrm{R}} \Phi; \Delta sk \leftarrow_{\mathrm{R}} \mathcal{K}_{pp_s};\ \widetilde{sk} \leftarrow sk + \Delta sk;\ \widetilde{vk} \leftarrow \mathsf{M}_{\mathsf{vk}}(pp_s, vk, \Delta sk);\ \widehat{\sigma} \leftarrow_{\mathrm{R}} \mathsf{Sign}(pp_s, sk, m);\ \widetilde{\sigma} \leftarrow \mathsf{M}_{\mathsf{sig}}(pp_s, vk, m, \widehat{\sigma}, \Delta sk);\ \widetilde{c} \leftarrow \mathsf{M}_{\mathsf{c}}(\Lambda, c, \Delta sk, e);\ \sigma \leftarrow (\widetilde{vk}, \widetilde{\sigma}, \widetilde{c})]. \quad (8)$$

Now, notice that an EUF-CMA adversary $\mathcal{B}$ for the underlying signature scheme Σ, who is given (pp_s, vk) and has access to the signing oracle $\mathcal{O}_{\mathsf{Sign}}(\cdot) := \mathsf{Sign}(pp_s, sk, \cdot)$, can perform the simulated process for generating VK (as explained above) and also simulate the process in Eq. (8) for $\mathcal{A}$. Furthermore, in the full proof, we show that if $\mathcal{A}$ outputs a successful forgery pair $(m', \sigma' = (\widetilde{vk}', \widetilde{\sigma}', \widetilde{c}'))$ such that $\mathsf{Ver}_{\mathsf{FS}}(pp, VK, m', \sigma') = \top$, then we can "extract" a successful forgery pair $(m', \widehat{\sigma}')$ such that $\mathsf{Ver}(pp_s, vk, m', \widehat{\sigma}') = \top$ by using the algorithms $\mathsf{DiffRec}$ and $\mathsf{M}_{\mathsf{sig}}$. (Roughly speaking, we can calculate the difference $\Delta sk'$ that corresponds to the difference between vk and $\widetilde{vk}'$ from c and $\widetilde{c}'$ via $\mathsf{DiffRec}$, and use $\Delta sk'$ to calculate $\widehat{\sigma}'$ via $\mathsf{M}_{\mathsf{sig}}$.) This enables us to turn $\mathcal{A}$ into an adversary (reduction algorithm) $\mathcal{B}$ attacking the EUF-CMA security of Σ. □

5 Instantiation

In this section, we first specify a concrete fuzzy key setting $\mathcal{F}$ for which our proposed fuzzy signature scheme is constructed in Sect. 5.1. Next, in Sect. 5.2, we provide some mathematical preliminaries used for our concrete linear sketch scheme and signature scheme. Armed with them, in Sects. 5.3 and 5.4, we show the concrete linear sketch scheme $\mathcal{S}$ for $\mathcal{F}$ and the signature scheme Σ_{MWS}, respectively, that can be used in our generic construction given in Sect. 4, which results in our proposed fuzzy signature scheme.

Our proposed fuzzy signature scheme for the fuzzy setting $\mathcal{F}$ (introduced in Sect. 5.1) is obtained straightforwardly from our generic construction in which $\mathcal{S}$ and Σ_{MWS} shown in this section are used. Though somewhat redundant, for the reader's convenience, we give a full description of the scheme in Appendix B.

On the Treatment of Real Numbers. Below, we use real numbers to represent and process fuzzy data. We assume that a suitable representation with sufficient accuracy is chosen to encode the real numbers whenever they need to be treated by the algorithms considered below. (If an algorithm takes a real number as input, its running time is according to the encoded version of input.)

5.1 Fuzzy Key Setting

Here, we specify a concrete fuzzy key setting $\mathcal{F} = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \epsilon)$ for which our FS scheme is constructed.

Metric space (d, X)**:** We define the space X by $X := [0,1)^n \subset \mathbb{R}^n$, where n is a parameter specified by the context (e.g. an object from which we measure fuzzy data). We use the L_∞-norm as the distance function $\mathsf{d} : X \times X \to \mathbb{R}$. Namely, for $\mathbf{x} = (x_1, \ldots, x_n) \in X$ and $\mathbf{x}' = (x'_1, \ldots, x'_n) \in X$, we define $\mathsf{d}(\mathbf{x}, \mathbf{x}') := \|\mathbf{x} - \mathbf{x}'\|_\infty := \max_{i \in [n]} |x_i - x'_i|$. Note that X forms an abelian group with respect to coordinate-wise addition (modulo 1).

Threshold t**:** For a security parameter k, we define the threshold $t \in \mathbb{R}$ so that

$$k = \lfloor -n \log_2(2t) \rfloor. \tag{9}$$

Looking ahead, this guarantees that the algorithm "WGen" that we introduce in the next subsection, is a PTA in k. We do not show that $\mathtt{FAR}$ is negligible here, because it is indirectly implied by the $\mathtt{EUF\text{-}CMA}$ security of our proposed fuzzy signature scheme.

Distribution $\mathcal{X}$**:** The uniform distribution over $[0,1)^n$. (Regarding how to relax this requirement, see the discussion in Sect. 6.)

Error distribution Φ **and Error parameter** ϵ**:** Φ is any efficiently samplable (according to k) distribution over X such that $\mathtt{FRR} \leq \epsilon$ for all $x \in X$.

5.2 Mathematical Preliminaries

Group Isomorphism Based on Chinese Remainder Theorem. Let $n \in \mathbb{N}$. Let $w_1, \ldots, w_n \in \mathbb{N}$ be positive integers with the same bit length (i.e. $\lceil \log_2 w_1 \rceil = \cdots = \lceil \log_2 w_n \rceil$), such that

$$\forall i \in [n] : w_i \leq 1/(2t), \qquad \text{and} \qquad \forall i \neq j \in [n] : \mathtt{GCD}(w_i, w_j) = 1, \tag{10}$$

and $W = \prod_{i \in [n]} w_i = \Theta(2^k)$, where k is defined as in Eq. (9).

We assume that there exists a deterministic algorithm WGen that on input (t, n) outputs $\mathbf{w} = (w_1, \ldots, w_n)$ satisfying the above.

For vectors $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{Z}^n$ and $\mathbf{w} = (w_1, \ldots w_n) \in \mathbb{Z}^n$, we define

$$\mathbf{v} \bmod \mathbf{w} := (v_1 \bmod w_1, \ldots, v_n \bmod w_n). \tag{11}$$

For vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{Z}^n$, we define the equivalence relation "$\sim$" by $\mathbf{v}_1 \sim \mathbf{v}_2 \overset{\text{def}}{\Leftrightarrow} \mathbf{v}_1 \bmod \mathbf{w} = \mathbf{v}_2 \bmod \mathbf{w}$, and let $\mathbb{Z}^n_{\mathbf{w}} := \mathbb{Z}^n / \sim$ be the quotient set of $\mathbb{Z}^n$ by $\sim$.

(Note that $(\mathbb{Z}_{\mathbf{w}}^n, +)$ constitutes an abelian group, where the addition is modulo $\mathbf{w}$ as defined in Eq. (11).)

Consider the following system of equations: given $\mathbf{v}, \mathbf{w} \in \mathbb{Z}^n$, find V such that $V \bmod w_i = v_i$ $(i \in [n])$. According to the Chinese remainder theorem (CRT), the solutoin V is determined uniquely modulo W. Thus, for a fixed $\mathbf{w} \in \mathbb{Z}^n$, we can define a mapping $\mathsf{CRT}_{\mathbf{w}} : \mathbb{Z}_{\mathbf{w}}^n \to \mathbb{Z}_W$ such that $\mathsf{CRT}_{\mathbf{w}}(\mathbf{v}) = V \in \mathbb{Z}_W$. We denote by $\mathsf{CRT}_{\mathbf{w}}^{-1}$ the "inverse" procedure of $\mathsf{CRT}_{\mathbf{w}}$.

Note that $\mathsf{CRT}_{\mathbf{w}}$ satisfies the following homomorphism: For all $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{Z}_{\mathbf{w}}^n$, it holds that $\mathsf{CRT}_{\mathbf{w}}(\mathbf{v}_1 + \mathbf{v}_2) = \mathsf{CRT}_{\mathbf{w}}(\mathbf{v}_1) + \mathsf{CRT}_{\mathbf{w}}(\mathbf{v}_2) \bmod W$. Since $\mathsf{CRT}_{\mathbf{w}}$ is bijective between $\mathbb{Z}_{\mathbf{w}}^n$ and $\mathbb{Z}_W$, $\mathsf{CRT}_{\mathbf{w}}$ is an isomorphism.

Coding and Error Correction. Let $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{N}^n$ be the n-dimensional vector satisfying the requirements in Eq. (10). Similarly to $\mathbb{Z}_{\mathbf{w}}^n$, we define $\mathbb{R}_{\mathbf{w}}^n := \mathbb{R}^n / \sim$ be the quotient set of real vector space $\mathbb{R}^n$ by the equivalence relation $\sim$, where for a real number $y \in \mathbb{R}$, we define $r = y \bmod w_i$ by the number such that $\exists n \in \mathbb{Z} : y = nw_i + r$ and $0 \le r < w_i$.

Let $\mathsf{E}_{\mathbf{w}} : \mathbb{R}^n \to \mathbb{R}_{\mathbf{w}}^n$ be the following function:

$$\mathsf{E}_{\mathbf{w}}(\mathbf{x}) := (w_1 x_1, \ldots, w_n x_n) \in \mathbb{R}_{\mathbf{w}}^n. \tag{12}$$

Note that $\mathsf{E}_{\mathbf{w}}(\mathbf{x} + \mathbf{e}) = \mathsf{E}_{\mathbf{w}}(\mathbf{x}) + \mathsf{E}_{\mathbf{w}}(\mathbf{e}) \pmod{\mathbf{w}}$ holds. Therefore, $\mathsf{E}_{\mathbf{w}}$ can be viewed as a kind of linear coding.

Let $\mathsf{C}_{\mathbf{w}} : \mathbb{R}_{\mathbf{w}}^n \to \mathbb{Z}_{\mathbf{w}}^n$ be the following function:

$$\mathsf{C}_{\mathbf{w}}((y_1, \ldots, y_n)) := (\lfloor y_1 + 0.5 \rfloor, \ldots, \lfloor y_n + 0,5 \rfloor). \tag{13}$$

We note that the round-off operation $\lfloor y_i + 0.5 \rfloor$ in $\mathsf{C}_{\mathbf{w}}$ can be regarded as a kind of error correction. Specifically, by the conditions in Eq. (10), the following properties are satisfied: For any $\mathbf{x}, \mathbf{x}' \in X$, if $\|\mathbf{x} - \mathbf{x}'\|_\infty < t$, then we have

$$\|\mathsf{E}_{\mathbf{w}}(\mathbf{x}) - \mathsf{E}_{\mathbf{w}}(\mathbf{x}')\|_\infty < t \cdot \max_{i \in [n]} \{w_i\} \le 0.5.$$

Therefore, for such $\mathbf{x}, \mathbf{x}'$, it always holds that

$$\mathsf{C}_{\mathbf{w}}\Big(\mathsf{E}_{\mathbf{w}}(\mathbf{x}) - \mathsf{E}_{\mathbf{w}}(\mathbf{x}')\Big) = \mathbf{0}. \tag{14}$$

Additionally, for any $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{s} \in \mathbb{Z}_{\mathbf{w}}^n$, the following holds:

$$\mathsf{C}_{\mathbf{w}}(\mathbf{x} + \mathbf{s}) = \mathsf{C}_{\mathbf{w}}(\mathbf{x}) + \mathbf{s} \pmod{\mathbf{w}}. \tag{15}$$

5.3 Linear Sketch

Let $\mathcal{F} = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \epsilon)$ be the fuzzy key setting defined in Sect. 5.1, and let $\mathbf{w} = (w_1, \ldots, w_n) = \mathsf{WGen}(t, n)$, where n is the dimension of X, and let $W = \prod_{i \in [n]} w_i$. We consider the linear sketch scheme $\mathcal{S} = (\mathsf{Sketch}, \mathsf{DiffRec})$ for $\mathcal{F}$ and the additive group $(\mathbb{Z}_W, +)$ $(=: \Lambda)$, as described in Fig. 5 (left).

$\mathsf{Sketch}(\Lambda, s \in \mathbb{Z}_W, \mathbf{x} \in [0,1)^n)$:	$\mathsf{M_c}(\Lambda, \mathbf{c}, \Delta s, \mathbf{e})$:
$\mathbf{c} \leftarrow (\mathsf{CRT}_{\mathbf{w}}^{-1}(s) + \mathsf{E}_{\mathbf{w}}(\mathbf{x})) \bmod \mathbf{w}$	$\mathbf{c}' \leftarrow (\mathbf{c} + \mathsf{CRT}_{\mathbf{w}}^{-1}(\Delta s) + \mathsf{E}_{\mathbf{w}}(\mathbf{e})) \bmod \mathbf{w}$
Return $\mathbf{c}$.	Return $\mathbf{c}'$.
$\mathsf{DiffRec}(\Lambda, \mathbf{c}, \mathbf{c}')$:	$\mathsf{Sim}(\Lambda)$:
$\Delta\mathbf{s} \leftarrow \mathsf{C}_{\mathbf{w}}(\mathbf{c} - \mathbf{c}')$; $\quad \Delta s \leftarrow \mathsf{CRT}_{\mathbf{w}}(\Delta\mathbf{s})$	$\mathbf{c} \leftarrow_{\mathrm{R}} \mathbb{R}_{\mathbf{w}}^n$
Return Δs.	Return $\mathbf{c}$.

Fig. 5. The linear sketch scheme $\mathcal{S} = (\mathsf{Sketch}, \mathsf{DiffRec})$ for the fuzzy key setting $\mathcal{F}$ (left), and the auxiliary algorithms $\mathsf{M_c}$ and Sim for showing the linearity property and the simulatability property, respectively (right).

We remark that although a sketch $\mathbf{c} = \mathsf{Sketch}(\Lambda, s, \mathbf{x})$ leaks some information of $\mathbf{x}$ (in particular, it leaks $w_i x_i \bmod 1$ for every $i \in [n]$) even if $s \in \mathbb{Z}_W$ is chosen uniformly at random, it does not affect the EUF-CMA security of our fuzzy signature scheme.

Lemma 2. *The linear sketch scheme $\mathcal{S}$ in Fig. 5 (left) satisfies Definition 5.*

Proof of Lemma 2. Correctness follows from the properties of the functions $\mathsf{CRT}_{\mathbf{w}}$, $\mathsf{E}_{\mathbf{w}}$, and $\mathsf{C}_{\mathbf{w}}$. Specifically, let $\mathbf{x}, \mathbf{x}' \in X$ be such that $\mathsf{d}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_\infty < t$. Let $s, \Delta s \in \mathbb{Z}_W$, and let $\mathbf{s} = \mathsf{CRT}_{\mathbf{w}}^{-1}(s)$ and $\Delta\mathbf{s} = \mathsf{CRT}_{\mathbf{w}}^{-1}(\Delta s)$. Furthermore, let $\mathbf{c} = \mathsf{Sketch}(\Lambda, s, \mathbf{x}) = (\mathbf{s} + \mathsf{E}_{\mathbf{w}}(\mathbf{x})) \bmod \mathbf{w}$ and $\mathbf{c}' = \mathsf{Sketch}(\Lambda, s + \Delta s, \mathbf{x}') = (\mathbf{s} + \Delta\mathbf{s} + \mathsf{E}_{\mathbf{w}}(\mathbf{x}')) \bmod \mathbf{w}$. Then, we have

$$\mathsf{C}_{\mathbf{w}}(\mathbf{c} - \mathbf{c}') = \mathsf{C}_{\mathbf{w}}\Big(\mathbf{s} + \mathsf{E}_{\mathbf{w}}(\mathbf{x}) - (\mathbf{s} + \Delta\mathbf{s} + \mathsf{E}_{\mathbf{w}}(\mathbf{x}'))\Big) \overset{(*)}{=} \Delta\mathbf{s} + \mathsf{C}_{\mathbf{w}}\Big(\mathsf{E}_{\mathbf{w}}(\mathbf{x}) - \mathsf{E}_{\mathbf{w}}(\mathbf{x}')\Big) \overset{(\dagger)}{=} \Delta\mathbf{s},$$

where (*) is due to Eq. (15) (we omit to write "mod $\mathbf{w}$"), and (†) is due to Eq. (14) and $\|\mathbf{x} - \mathbf{x}'\|_\infty < t$. Thus, $\mathsf{DiffRec}(\Lambda, \mathsf{Sketch}(\Lambda, s, \mathbf{x}), \mathsf{Sketch}(\Lambda, s + \Delta s, \mathbf{x}')) = \mathsf{CRT}_{\mathbf{w}}(\mathsf{C}_{\mathbf{w}}(\mathbf{c} - \mathbf{c}')) = \mathsf{CRT}_{\mathbf{w}}(\Delta\mathbf{s}) = \Delta s$, satisfying Eq. (5).

Regarding linearity, we consider the algorithm $\mathsf{M_c}$ as described in Fig. 5 (upper-right). To see that $\mathsf{M_c}$ satisfies linearity, let $\mathbf{x}, \mathbf{e} \in \mathbb{R}_{\mathbf{w}}^n$ and $s, \Delta s \in \mathbb{Z}_W$, and let $\mathbf{s} = \mathsf{CRT}_{\mathbf{w}}^{-1}(s)$ and $\Delta\mathbf{s} = \mathsf{CRT}_{\mathbf{w}}^{-1}(\Delta s)$. Then, note that $\mathsf{Sketch}(\Lambda, s, \mathbf{x}) = (\mathbf{s} + \mathsf{E}_{\mathbf{w}}(\mathbf{x})) \bmod \mathbf{w}$ and $\mathsf{CRT}_{\mathbf{w}}^{-1}(s + \Delta s) = (\mathbf{s} + \Delta\mathbf{s}) \bmod \mathbf{w}$. Thus, it holds that

$$\begin{aligned}\mathsf{M_c}(\Lambda, \mathsf{Sketch}(\Lambda, s, \mathbf{x}), \Delta s, \mathbf{e}) &= \Big(\mathbf{s} + \mathsf{E}_{\mathbf{w}}(\mathbf{x}) + \Delta\mathbf{s} + \mathsf{E}_{\mathbf{w}}(\mathbf{e})\Big) \bmod \mathbf{w} \\ &= \Big(\mathbf{s} + \Delta\mathbf{s} + \mathsf{E}_{\mathbf{w}}(\mathbf{x} + \mathbf{e})\Big) \bmod \mathbf{w} = \mathsf{Sketch}(\Lambda, s + \Delta s, \mathbf{x} + \mathbf{e}),\end{aligned}$$

satisfying Eq. (6).

Regarding simulatability, note that by our requirement that $\mathcal{X}$ is the uniform distribution over $[0,1)^n$, if $\mathbf{x} \leftarrow_{\mathrm{R}} \mathcal{X}$, then the output of $\mathsf{Sketch}(\Lambda, s, \mathbf{x})$ is uniformly distributed over $\mathbb{R}_{\mathbf{w}}^n$, no matter what $s \in \mathbb{Z}_W$ is. Therefore, the probabilistic algorithm $\mathsf{Sim}(\Lambda)$ described in Fig. 5 (bottom-right) that outputs a uniformly distributed value $\mathbf{c}$ over $\mathbb{R}_{\mathbf{w}}^n$ satisfies the simulatability. This completes the proof of Lemma 2. □

5.4 Modified Waters Signature Scheme

Here, we show a variant of the Waters signature [22], which we call the *modified Waters signature* (MWS) scheme Σ_{MWS}.

Specific Bilinear Group Generator $\mathsf{BGGen}_{\mathsf{MWS}}$. In the MWS scheme, we use a (slightly) non-standard way for specifying bilinear groups, namely, the order p of (symmetric) bilinear groups is generated based on an integer $W = \prod_{i\in[n]} w_i$, where $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{Z}^n$ satisfies the conditions in Eq. (10), so that p is the smallest prime satisfying $W|p-1$. More concretely, we consider the following algorithm PGen for choosing p from W: On input $W \in \mathbb{N}$, for $i = 1, 2, \ldots$ check if $p = iW + 1$ is a prime and return p if this is the case. Otherwise, increment $i \leftarrow i + 1$ and go to the next iteration.

According to the prime number theorem, the density of primes among the natural numbers that are less than N is roughly $1/\ln N$, and thus, for i's that are exponentially smaller than W, the probability that $iW + 1$ is a prime can be roughly estimated as $1/\ln W$. Therefore, by using the above algorithm PGen, one can find a prime p satisfying $W|p-1$ by performing the primality testing for $O(\ln W) = O(k)$ times on average (recall that $W = \Theta(2^k)$). Furthermore, if $\mathsf{PGen}(W)$ outputs p, then it is guaranteed that $p/W = O(k)$. (This fact is used for security).

Let $\mathsf{BGGen}_{\mathsf{MWS}}$ denote an algorithm that, given 1^k, runs $\mathbf{w} \leftarrow \mathsf{WGen}(t, n)$ where t and n are the parameters from the fuzzy data setting $\mathcal{F}$ corresponding the security parameter k, computes $W \leftarrow \prod_{i\in[n]} w_i$, $p \leftarrow \mathsf{PGen}(W)$, and outputs a description of bilinear groups $\mathcal{BG} = (p, \mathbb{G}, \mathbb{G}_T, g, e)$, where $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups with order p and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map.

Construction. Using $\mathsf{BGGen}_{\mathsf{MWS}}$ and the algorithms in the original Waters signature scheme Σ_{Wat} (see Fig. 3), the MWS scheme $\Sigma_{\mathsf{MWS}} = (\mathsf{Setup}_{\mathsf{MWS}}, \mathsf{KG}_{\mathsf{MWS}}, \mathsf{Sign}_{\mathsf{MWS}}, \mathsf{Ver}_{\mathsf{MWS}})$ is constructed as in Fig. 6 (left). Note that the component pp_{Wat} in a public parameter pp (generated by $\mathsf{Setup}_{\mathsf{MWS}}$) is distributed identically to that generated in the original Waters scheme Σ_{Wat} in which the bilinear group generator $\mathsf{BGGen}_{\mathsf{MWS}}$ is used. Therefore, Σ_{MWS} can be viewed as the original Waters scheme Σ_{Wat}, except that (1) we specify how to generate the parameter of bilinear groups by $\mathsf{BGGen}_{\mathsf{MWS}}$, and (2) we use a secret key sk' (for the Waters scheme) of the form $sk' = z^{sk} \bmod p$, thereby we change the signing key space from $\mathbb{Z}_p$ to $\mathbb{Z}_W$.

In the following, we show that Σ_{MWS} satisfies `EUF-CMA` security (based on the CDH assumption with respect to $\mathsf{BGGen}_{\mathsf{MWS}}$) and the homomorphic property (Definition 3), and thus can be used as the underlying signature scheme for our generic construction of a fuzzy signature scheme. (One might suspect the plausibility of the CDH assumption with respect to $\mathsf{BGGen}_{\mathsf{MWS}}$ due to our specific choice of p. We discuss it in Appendix C.)

Lemma 3. *If the CDH assumption holds with respect to* $\mathsf{BGGen}_{\mathsf{MWS}}$*, then the MWS scheme* Σ_{MWS} *is* `EUF-CMA` *secure.*

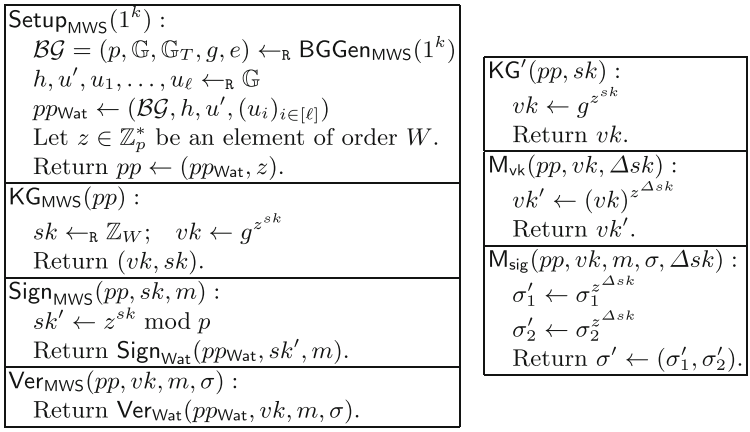

Fig. 6. The modified Waters signature (MWS) scheme Σ_{MWS} (left), and the auxiliary algorithms $(\mathsf{KG}', \mathsf{M}_{\mathsf{vk}}, \mathsf{M}_{\mathsf{sig}})$ for showing the homomorphic property (right).

Proof of Lemma 3. Let $pp = (pp_{\mathsf{Wat}}, z)$ be a public parameter output by $\mathsf{Setup}_{\mathsf{MWS}}$, let $D_{pp}^{(1)} = \{sk \leftarrow_{\mathrm{R}} \mathbb{Z}_W; sk' \leftarrow z^{sk} \bmod p : sk'\}$ and $D_{pp}^{(2)} = \{sk' \leftarrow_{\mathrm{R}} \mathbb{Z}_p : sk'\}$. Note that the support of $D_{pp}^{(1)}$ is a strict subset of that of $D_{pp}^{(2)}$.

Now, let $\mathcal{A}$ be any PPTA that attacks the EUF-CMA security of the MWS scheme. Let Expt_1 be the original EUF-CMA experiment, i.e. $\mathsf{Expt}_{\Sigma_{\mathsf{MWS}},\mathcal{A}}^{\text{EUF-CMA}}(k)$, and let Expt_2 be the experiment that is defined in the same manner as Expt_1, except that sk' is sampled according to the distribution $D_{pp}^{(2)}$. For both $i \in \{1, 2\}$, let Adv_i be the advantage of $\mathcal{A}$ (i.e. the probability of $\mathcal{A}$ outputting a successful forgery) in Expt_i. Then, by Lemma 1, we have $\mathsf{Adv}_1 \leq (p/W) \cdot \mathsf{Adv}_2 = O(k) \cdot \mathsf{Adv}_2$. Furthermore, it is straightforward to see that succeeding in forging in Expt_2 is as difficult as succeeding in breaking the EUF-CMA security of the original Waters scheme Σ_{Wat} (in which the bilinear group generator $\mathsf{BGGen}_{\mathsf{MWS}}$ is used), and thus Adv_2 is negligible if Σ_{Wat} is EUF-CMA secure.

Finally, due to Waters [22], if the CDH assumption holds with respect to $\mathsf{BGGen}_{\mathsf{MWS}}$, then the Waters scheme Σ_{Wat} (in which $\mathsf{BGGen}_{\mathsf{MWS}}$ is used,) is EUF-CMA secure. Combining all the explanations proves the lemma. □

Lemma 4. *The MWS scheme Σ_{MWS} is homomorphic (as per Definition 3).*

Proof of Lemma 4. Consider the algorithms $(\mathsf{KG}', \mathsf{M}_{\mathsf{vk}}, \mathsf{M}_{\mathsf{sig}})$ that are described in Fig. 6 (right). It is easy to see that using KG', $\mathsf{KG}_{\mathsf{MWS}}$ can be rewritten with the process in Eq. (1), where the secret key space is $\mathbb{Z}_W$.

Moreover, it should also be easy to see that M_{vk} satisfies the requirement in Eq. (2). Indeed, let $pp = (pp_{\mathsf{Wat}}, z)$ be a public parameter, and let $sk, \Delta sk \in \mathbb{Z}_W$. Then, it holds that $\mathsf{M}_{\mathsf{vk}}(pp, \mathsf{KG}'(pp, sk), \Delta sk) = (g^{z^{sk}})^{z^{\Delta sk}} = g^{z^{sk+\Delta sk}} = \mathsf{KG}'(pp, sk + \Delta sk)$, satisfying Eq. (2).

Finally, we observe that $\mathsf{M}_{\mathsf{sig}}$ satisfies the requirements in Eqs. (3) and (4). Let $pp = (pp_{\mathsf{Wat}}, z)$ and $sk, \Delta sk \in \mathbb{Z}_W$ as above, and $m = (m_1 \| \ldots \| m_\ell) \in \{0,1\}^\ell$

be a message to be signed. Let (σ_1, σ_2) be a signature on the message m that is generated by $\mathsf{Sign}_{\mathsf{MWS}}(pp, sk, m; r)$, where $r \in \mathbb{Z}_p$ is a randomness. By definition, σ_1 and σ_2 are of the form $\sigma_1 = h^{z^{sk}} \cdot (u' \prod_{i \in [\ell]} u_i^{m_i})^r$ and $\sigma_2 = g^r$, respectively. Thus, if $\sigma' = (\sigma'_1, \sigma'_2)$ is output by $\mathsf{M}_{\mathsf{sig}}(pp, vk, m, \sigma, \Delta sk)$, then it holds that $\sigma'_1 = \sigma_1^{z^{\Delta sk}} = h^{z^{sk+\Delta sk}} \cdot (u' \prod_{i \in [\ell]} u_i^{m_i})^{r \cdot z^{\Delta sk}}$, and $\sigma'_2 = \sigma_2^{z^{\Delta sk}} = g^{r \cdot z^{\Delta sk}}$. This implies $\sigma' = (\sigma'_1, \sigma'_2) = \mathsf{Sign}_{\mathsf{MWS}}(pp, sk + \Delta sk, m; r \cdot z^{\Delta sk})$. Note that for any $\Delta sk \in \mathbb{Z}_W$, if $r \leftarrow_{\mathtt{R}} \mathbb{Z}_p$, then $((r \cdot z^{\Delta sk}) \bmod p)$ is uniformly distributed in $\mathbb{Z}_p$. This implies that the distributions considered in Eq. (3) are identical. Furthermore, by the property of the MWS scheme (which is inherited from the original Waters scheme), any signature $\sigma' = (\sigma'_1, \sigma'_2)$ satisfying $\mathsf{Ver}_{\mathsf{MWS}}(pp, vk, m, \sigma') = \top$ must satisfy the property that there exists $r' \in \mathbb{Z}_p$ such that $\mathsf{Sign}_{\mathsf{MWS}}(pp, sk, m; r') = \sigma'$. Putting everything together implies that for any $sk, \Delta sk \in \mathbb{Z}_W$, any message $m \in \{0,1\}^\ell$, any signature σ such that $\mathsf{Ver}_{\mathsf{MWS}}(pp, vk, m, \sigma) = \top$, if $vk = \mathsf{KG}'(pp, sk)$, $vk' = \mathsf{M}_{\mathsf{vk}}(pp, vk, \Delta sk)$, and $\sigma' = \mathsf{M}_{\mathsf{sig}}(pp, vk, m, \sigma, \Delta sk)$, then it holds that $\mathsf{Ver}_{\mathsf{MWS}}(pp, vk', m, \sigma') = \top$. Therefore, the requirement regarding Eq. (4) is satisfied as well. This completes the proof of Lemma 4. □

6 Towards Public Biometric Infrastructure

As one of the promising applications of our fuzzy signature scheme Σ_{FS}, we discuss how it can be used to realize a biometric-based PKI that we call the *public biometric infrastructure (PBI)*.

The PBI is a biometric-based PKI that allows to use biometric data itself as a private key. Since it does not require a helper string to extract a private key, it does not require users to carry a dedicated device that stores it. Like the PKI, it provides the following functionalities: (1) registration, (2) digital signature, (3) authentication, and (4) cryptographic communication. At the time of registration, a user presents his/her biometric data x, from which the public key pk is generated. A certificate authority (CA) issues a public key certificate to ensure the link between pk and the user's identify (in the same way as the PKI). It must be sufficiently hard to restore x or estimate any "acceptable" biometric feature (i.e. biometric feature $\tilde{x}$ that is sufficiently close to x) from pk. This requirement is often referred to as *irreversibility* [8,19]. Note that the irreversibility is clearly included in the unforgeability, since the adversary who obtains x or $\tilde{x}$ can forge a signature σ for any message m. Since our fuzzy signature scheme Σ_{FS} is `EUF-CMA` secure, it also satisfies the irreversibility.

It is well-known that a digital signature scheme can be used to realize authentication and cryptographic communication, as standardized in [9]. Firstly, a challenge-response authentication protocol can be constructed based on a digital signature scheme (refer to [18] for details). Secondly, an authenticated key exchange (AKE) protocol can also be constructed based on a digital signature scheme and Diffie-Hellman Key Exchange protocol. In the same way, we can construct an authentication protocol and a cryptographic communication protocol in the PBI using our fuzzy signature scheme Σ_{FS}.

Remaining Challenges and Future Work. In Sect. 5, we showed an EUF-CMA secure FS scheme Σ_{FS}. However, we proved this under the assumption that a noisy string is uniform and has enough entropy. Thus, when using a biometric feature as a noisy string in Σ_{FS}, its EUF-CMA security is, for now, guaranteed only in the case where a biometric feature is uniform and has enough entropy.

A well-known approach to measure the biometric entropy is Daugman's *discrimination entropy* [2]. He considered a distribution of a Hamming distance m between two iriscodes (well-known iris features [3]) that are extracted from two different irises, and showed that it can be quite well approximated using the binomial distribution $B(n, p)$, where $n = 249$ and $p = 0.5$. He referred to the parameter n $(= 249)$ as a discrimination entropy. The probability that two different iriscodes exactly match can be approximated to be 2^{-249}. However, it does not mean that a fuzzy signature scheme using the iriscode x is as secure as an ordinary signature scheme with a 249-bit private key, since the adversary does not have to estimate the original iriscode x, but only has to estimate an iriscode $\tilde{x}$ that is sufficiently close to x.

If a single biometric feature does not have enough entropy, we can use a multibiometric fusion scheme [16] that combines multiple sources of biometric information (e.g. fingerprint, face, and iris; left iris and right iris) to increase entropy. A multibiometric sensor that simultaneously acquires multiple biometrics (e.g. iris and face [1]; fingerprint and finger-vein [15]) has also been widely developed in recent years. Thus, we consider that using multiple biometrics is one possible direction to increase entropy without affecting usability.

Also, a biometric feature is non-uniform in general. The relation between the security in the uniform key setting (ideal model) and the one in the non-uniform key setting (real model) has been studied in several works in cryptography, e.g. [5]. As future work, we plan to prove the security of our fuzzy signature scheme in the non-uniform case, by applying (or extending) the techniques from them.

Acknowledgement. The authors would like to thank the anonymous reviewers of ACNS 2015 for their invaluable comments and suggestions.

A More on the Limitations of Fuzzy-Extractor-Based Approaches

The right of Fig. 1 shows an example of a digital signature system using the fuzzy extractor. Assume that the client generates a signature on a message, and the server verifies it. At the time of registration, a signing key sk and a helper string P are generated from a noisy string (e.g. biometric feature) x, and a verification key vk corresponding to sk is generated and stored in a server-side DB. At the time of signing, the client generates a signature σ on a message m using P and another noisy string x', and sends σ to the server. The server verifies whether σ is a valid signature on m under vk. If x' is close to x, it outputs "$\top$" (valid). Otherwise, it outputs "$\bot$" (invalid). The important point here is that the helper

string P has to be stored in some place so that the client can retrieve it at the time of signing.

There are three possible models for storing the helper string: Store-on-Token (SOT), Store-on-Client (SOC), and Store-on-Server (SOS). In the SOT, the helper string is stored in a hardware token (e.g. smart card, USB token). Since this model requires each user to possess a token, it reduces usability. In the SOC, the helper string is stored in a client device. Although this model can be applied to the applications where each user has his/her own client device, it cannot be employed if the client device is shared by general public (e.g. bank ATM, POS, and kiosk terminal). In the SOS, the helper string is stored in a server-side DB, and the client queries for the helper string to the server at the time of signing. However, it cannot be used in an offline environment (i.e. a user generates a signature, which is sent to the server later, offline).

To sum up, the SOT reduces usability, and the SOC/SOS limit the client environment. Although a digital signature scheme using biometrics is proposed in [10,11] and an extended version of the PKI based on biometrics is discussed in [17], all of them require additional data like the helper string and suffer from this kind of problem.

B Full Description of the Proposed Fuzzy Signature Scheme

Let $\ell = \ell(k)$ be a positive polynomial that denotes the length of messages. Let $\mathcal{F} = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \epsilon)$ be the fuzzy key setting defined in Sect. 5.1, where t (and n) are determined according to the security patameter k. Let $\mathsf{BGGen}_{\mathsf{MWS}}$ be the bilinear group generator defined in Sect. 5.4. Then, our proposed fuzzy signature scheme $\Sigma_{\mathsf{FS}} = (\mathsf{Setup}_{\mathsf{FS}}, \mathsf{KG}_{\mathsf{FS}}, \mathsf{Sign}_{\mathsf{FS}}, \mathsf{Ver}_{\mathsf{FS}})$ for the fuzzy key setting $\mathcal{F}$ is constructed as in Fig. 7.

It should be straightforward to see that Σ_{FS} is a straightforward combination of the linear sketch scheme $\mathcal{S}$ for $\mathcal{F}$ shown in Sect. 5.3 and the MWS scheme Σ_{MWS} shown in Sect. 5.4.

C On the Plausibility of the CDH Assumption with Respect to $\mathsf{BGGen}_{\mathsf{MWS}}$

For the security of the MWS scheme Σ_{MWS} constructed in Sect. 5.4, we need to assume that the CDH assumption holds with respect to $\mathsf{BGGen}_{\mathsf{MWS}}$. One might suspect the plausibility of this assumption because of our specific choice of the order p. However, to the best of our knowledge, there is no effective attack on the discrete logarithm assumption in the groups $\mathbb{G}$ and $\mathbb{G}_T$, let alone the CDH assumption.

Actually, the discrete logarithm problem for the multiplicative group $(\mathbb{Z}_p^*, \cdot)$ is easy because $W | p - 1$ and $W = \prod_{i \in [n]} w_i$, and thus we can apply the Pohlig-Hellman algorithm [13] to reduce an instance of the discrete logarithm problem

$\mathsf{Setup}_{\mathsf{FS}}(\mathcal{F}, 1^k)$: Let $\mathcal{BG} := (p, \mathbb{G}, \mathbb{G}_T, g, e) \leftarrow \mathsf{BGGen}_{\mathsf{MWS}}(1^k)$ $h, u', u_1, \ldots, u_\ell \leftarrow_{\mathtt{R}} \mathbb{G}$ Let z be an element of $\mathbb{Z}_p^*$ of order W. Let $\Lambda := (\mathbb{Z}_W, +)$. Return $pp \leftarrow (\mathcal{BG}, h, u', (u_i)_{i\in[\ell]}, z, \Lambda)$.	$\mathsf{KG}_{\mathsf{FS}}(pp, \mathbf{x})$: $sk \leftarrow_{\mathtt{R}} \mathbb{Z}_W$ $vk \leftarrow g^{z^{sk}}$ $\mathbf{c} \leftarrow (\mathsf{CRT}_{\mathbf{w}}^{-1}(sk) + \mathsf{E}_{\mathbf{w}}(\mathbf{x})) \bmod \mathbf{w}$ Return $VK \leftarrow (vk, \mathbf{c})$.
$\mathsf{Sign}_{\mathsf{FS}}(pp, \mathbf{x}', m)$: Parse m as $(m_1 \| \ldots \| m_\ell) \in \{0,1\}^\ell$. $\widetilde{sk} \leftarrow_{\mathtt{R}} \mathbb{Z}_W$ $\widetilde{vk} \leftarrow g^{z^{\widetilde{sk}}}$ $r \leftarrow_{\mathtt{R}} \mathbb{Z}_p$ $\widetilde{\sigma}_1 \leftarrow h^{z^{\widetilde{sk}}} \cdot (u' \prod_{i\in[\ell]} u_i^{m_i})^r$ $\widetilde{\sigma}_2 \leftarrow g^r$ $\widetilde{\mathbf{c}} \leftarrow (\mathsf{CRT}_{\mathbf{w}}^{-1}(\widetilde{sk}) + \mathsf{E}_{\mathbf{w}}(\mathbf{x}')) \bmod \mathbf{w}$ Return $\sigma \leftarrow (\widetilde{vk}, \widetilde{\sigma}_1, \widetilde{\sigma}_2, \widetilde{\mathbf{c}})$.	$\mathsf{Ver}_{\mathsf{FS}}(pp, vk, m, \sigma)$: $(\widetilde{vk}, \widetilde{\sigma}_1, \widetilde{\sigma}_2, \widetilde{\mathbf{c}}) \leftarrow \sigma$ Parse m as $(m_1 \| \ldots \| m_\ell) \in \{0,1\}^\ell$. If $e(\widetilde{\sigma}_2, u' \cdot \prod_{i\in[\ell]} u_i^{m_i}) \cdot e(\widetilde{vk}, h)$ $\neq e(\widetilde{\sigma}_1, g)$ then return $\bot$. $\Delta\mathbf{s} \leftarrow \mathsf{C}_{\mathbf{w}}(\mathbf{c} - \widetilde{\mathbf{c}})$ $\Delta sk \leftarrow \mathsf{CRT}_{\mathbf{w}}(\Delta\mathbf{s})$ If $(vk)^{z^{\Delta sk}} = \widetilde{vk}$ then return $\top$ else return $\bot$.

Fig. 7. The full description of the proposed fuzzy signature scheme Σ_{FS}.

in $\mathbb{Z}_p^*$ to instances of the discrete logarithm problems in $\mathbb{Z}_{w_i}$. *However, it does not mean that the Pohlig-Hellman algorithm is applicable to the discrete logarithm problem in* $\mathbb{G}$ *or* $\mathbb{G}_T$*, whose order is a prime.*

Note that a verification/signing key pair (vk, sk) of the MWS scheme Σ_{MWS} is of the following form $(vk, sk) = (g^{z^{sk}}, sk)$, where $sk \leftarrow_{\mathtt{R}} \mathbb{Z}_W$, and z and W are in a public parameter pp. In fact, due to the existence of the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, a variant of Pollard' ρ-algorithm [14] is applicable, and one can recover sk from vk (and pp) with $O(\sqrt{W})$ steps. However, this is exponential time in a security parameter k. (Recall that $W = \Theta(2^k)$.) This also does not contradict the `EUF-CMA` security of the MWS scheme shown in Lemma 3.

References

1. 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. Springer, London (2013). Chap. 12
2. Daugman, J.: The importance of being random: statistical principles of iris recognition. Pattern Recogn. **36**(2), 279–291 (2003)
3. Daugman, J.: How iris recognition works. IEEE Trans. Circuits Syst. Video Technol. **14**, 21–30 (2004)
4. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Heidelberg (2004)
5. Dodis, Y., Yu, Y.: Overcoming weak expectations. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 1–22. Springer, Heidelberg (2013)
6. Ellison, C., Schneier, B.: Ten risks of PKI: what you're not being told about public key infrastructure. Comput. Secur. J. **16**(1), 1–7 (2000)

7. Fan, L., Zheng, J., Yang, J.: A biometric identity based signature in the standard model. In: Proceedings of the IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC 2009). pp. 552–556 (2009)
8. ISO/IEC JTC 1/SC 27 24745: Biometric information protection (2011)
9. ISO/IEC JTC 1/SC 27 9798–3: Mechanisms using digital signature techniques (1998)
10. Jo, J.-G., Seo, J.-W., Lee, H.-W.: Biometric digital signature key generation and cryptography communication based on fingerprint. In: Preparata, F.P., Fang, Q. (eds.) FAW 2007. LNCS, vol. 4613, pp. 38–49. Springer, Heidelberg (2007)
11. Kwon, T., Lee, H., Lee, J.: A practical method for generating digital signatures using biometrics. IEICE Trans. Commun. **E90–B**(6), 1381–1389 (2007)
12. Pappu, R., Recht, B., Taylor, J., Gershenfeld, N.: Physical one-way functions. Science **297**(5589), 2026–2030 (2002)
13. Pohlig, S.C., Hellman, M.E.: An improved algorithm for computing logarithms over $gf(p)$ and its cryptographic significance (corresp.). IEEE Trans. Inf. Theor. **24**, 106–110 (1978)
14. Pollard, J.M.: Monte carlo methods for index computation (mod p). Math.Comput. **32**, 918–924 (1978)
15. Raghavendra, R., Raja, K.B., Surbiryala, J., Busch, C.: A low-cost multimodal biometric sensor to capture finger vein and fingerprint. In: Proceedings of 2014 IEEE the International Joint Conference on Biometrics (IJCB 2014), pp. 1–7 (2014)
16. Ross, A., Nandakumar, K., Jain, A.K.: Handbook of Multibiometrics. Springer, Heidelberg (2006)
17. Scheirer, W.J., Bishop, B., Boult, T.E.: Beyond pki: the biocryptographic key infrastructure. In: Proceedings of the 2010 IEEE International Workshop on Information Forensics and Security (WIFS 2010), pp. 1–6 (2010)
18. Schneier, B.: Applied Cryptography. Wiley, New York (1995)
19. Simoens, K., Yang, B., Zhou, X., Beato, F., Busch, C., Newton, E., Preneel, B.: Criteria towards metrics for benchmarking template protection algorithms. In: Proceedings of the 5th IAPR International Conference on Biometrics (ICB 2012) (2012)
20. Wang, C., Chen, W., Liu, Y.: A fuzzy identity based signature scheme. In: Proceedings of the International Conference on E-Business and Information System Security (EBISS 2009), pp. 1–5 (2009)
21. Wang, C., Kim, J.-H.: Two constructions of fuzzy identity based signature. In: Proceedings of the 2nd International Conference on Biomedical Engineering and Informatics (BMEI 2009), pp. 1–5 (2009)
22. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
23. Wu, Q.: Fuzzy biometric identity-based signature in the standard model. J. Comput. Inf. Syst. **8**(20), 8405–8412 (2012)
24. Yang, P., Cao, Z., Dong, X.: Fuzzy identity based signature with applications to biometric authentication. Comput. Electr. Eng. **37**(4), 532–540 (2011)