Fuzzy Signatures: Relaxing Requirements and a New Construction

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

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

Abstract. Takahashi et al. (ACNS 2015) introduced the notion of *fuzzy signature*, which is a signature scheme that allows a signature to be generated using "fuzzy data" (i.e. a noisy string such as a biometric feature) as a signing key, without using any additional user-specific data (such as a helper string in the context of fuzzy extractors). They gave a generic construction of a fuzzy signature scheme from the combination of an ordinary signature scheme with some homomorphic properties regarding keys and signatures, and a new primitive that they call linear sketch, and showed a concrete instantiation based on the Waters signature scheme (EUROCRYPT 2005). A major weakness of their scheme is that fuzzy data is assumed to be distributed uniformly, and another is that it has somewhat large public parameter (proportional to the security parameter), and requires bilinear groups, and either (or both) of these properties could be barriers for implementation and/or practical use.

In this paper, we revisit the results of Takahashi et al.: We show that in their generic construction, the requirements on each of the building blocks can be relaxed in several aspects. More specifically, our relaxation for the underlying linear sketch scheme allows us to use a new linear sketch scheme (that we propose) for a fuzzy key setting different from that of Takahashi et al., for which we only require that the average min-entropy of fuzzy data is high (under the situation some part of its information is leaked). Furthermore, our relaxation on the underlying signature scheme enables us to now use the Schnorr signature scheme as a building block. Our concrete instantiation of a fuzzy signature scheme is, although relying on a random oracle, arguably much more practical than the scheme by Takahashi et al. The latter relaxation routes through a variant of related key security for signature schemes.

Keywords: Fuzzy signature · Schnorr signature · Biometrics

1 Introduction

1.1 Background and Motivation

As the information society grows rapidly, the public key infrastructure (PKI) plays a more significant role as an infrastructure for managing digital certificates.

M. Manulis et al. (Eds.): ACNS 2016, LNCS 9696, pp. 97–116, 2016.
DOI: 10.1007/978-3-319-39555-5_6

It is also expected to be widely used for personal use such as national IDs and e-government services. One of the biggest risks in the PKI, which needs to be considered in the personal use, lies in a user's private key [8]; since the user's identity is verified based only on his/her private key, the user needs to protect the private key in a highly secure manner. For example, the user is required to store his/her private key into a smart card (or USB token), and remember a password to activate the key. Such an approach, however, can reduce the usability, especially for elderly people in an aging society.

One of the promising approaches to fundamentally solve this problem is to use *biometric data* (e.g. fingerprint, face, and iris) as a private key. Since a user's biometrics is a part of human body, it can offer a more secure and usable way to link the individual with his/her private key (i.e. it is not forgotten unlike passwords and is much harder to steal than cards). Also, a sensor that captures multiple biometrics simultaneously (e.g. face and iris [4]; fingerprint and finger-vein [13]) has been widely developed to obtain a large amount of entropy at one time, and a recent study [11] has shown that very high accuracy (e.g. the false acceptance rate (FAR) is 2^{-133} (resp. 2^{-87}) when the false rejection rate (FRR) is 0.055 (resp. 0.0053)) can be achieved by combining four finger-vein features.

However, since biometric data is noisy data that fluctuates each time it is captured, it cannot be directly used as a private key. In this paper, we call such a noisy string *fuzzy data*.

Fuzzy Signature. Takahashi et al. [15] introduced a concept of digital signature called *fuzzy signature* to address this issue. Fuzzy signature consists of three algorithms $(\mathsf{KG}_{\mathsf{FS}}, \mathsf{Sign}_{\mathsf{FS}}, \mathsf{Ver}_{\mathsf{FS}})$.[1] First, the key generation algorithm $\mathsf{KG}_{\mathsf{FS}}$ takes a noisy string x as input, and outputs a verification key vk. Then, the signing algorithm $\mathsf{Sign}_{\mathsf{FS}}$ takes another noisy string x' and a message m as input, and outputs a signature σ. Finally, the verification algorithm $\mathsf{Ver}_{\mathsf{FS}}$ verifies whether σ is a valid signature (on a message m) or not, using the verification key vk. If x is close to x', σ is verified as valid. As discussed in [15], the key difference between fuzzy signatures and digital signatures using a *fuzzy extractor* [7], is that the former does not need user-specific auxiliary data (called a "helper string" in the context of fuzzy extractors) to generate a signature σ. Thus, a fuzzy signature scheme can be used to realize a biometric version of the PKI that does not require a user to carry a dedicated device containing the helper string, which is preferable in terms of the usability.

We note that a fuzzy signature scheme has a public parameter (generated by the setup algorithm) that is needed for signing messages. However, it is not user-specific, and thus the user need not carry it by himself/herself. In contrast, in fuzzy-extractor-based signature schemes, the auxiliary data (which can be made public, but) is user-specific, and has to be present at the time of signing together with the user (signer) himself/herself. Hence, it has to be carried out by the user, or stored somewhere in a remote server and a signing device has

[1] To be precise, a fuzzy signature scheme also has the "setup" algorithm for generating a public parameter that is shared by all users, as explained shortly.

to be on-line when generating a signature. More in-depth discussions between fuzzy signatures and fuzzy-extractor-based signatures can be found in [15].

Overview of the Results of Takahashi et al. [15] *and Our Motivation.* In this paper, we build on the results of Takahashi et al., and give new results on fuzzy signatures. To better explain and understand our motivation and results, let us briefly recall the technical results in [15]: In addition to formally define fuzzy signatures, Takahashi et al. formalized what they call a *fuzzy key setting*, which formalizes some necessary information about the setting over which fuzzy data is considered, e.g. the metric space to which fuzzy data belongs, the threshold with which two sampled data are considered close/far, the distribution from which each fuzzy data is assumed to be drawn, how the fluctuation of fuzzy data is modeled, etc. A fuzzy signature scheme is associated with such a fuzzy key setting.

Takahashi et al. also introduced a tool that they call *linear sketch*, which is a kind of a pair of linear encoding and error correction methods, that is associated with a fuzzy key setting. They then gave a generic construction of a fuzzy signature scheme for a fuzzy key setting from a combination of a linear sketch scheme (associated with the same fuzzy key setting) and an ordinary signature scheme that has some homomorphic properties regarding keys and signatures. They then specified a concrete fuzzy key setting in which a fuzzy data is distributed uniformly over some metric space, and showed a linear sketch scheme for it, and also showed an ordinary signature scheme based on the Waters signature scheme [16] that can be used with the linear sketch in their generic construction. By using these concrete linear sketch scheme and the signature scheme, Takahashi et al. [15] showed a concrete instantiation of a fuzzy signature scheme for the above fuzzy key setting.

Since Takahashi et al.'s fuzzy signature scheme is based on the Waters signature scheme [16], it has somewhat large public parameter (where the number of group elements in the parameter is proportional to the security parameter), and requires bilinear groups. Therefore, either (or both) of these properties, namely large parameter size and the use of bilinear groups, could be barriers for implementation (especially in computationally limited devices) and/or practical use. More importantly, they assume that fuzzy data is distributed uniformly (over some metric space). This is quite a strong assumption, and thus potentially limits the usefulness of their result. For example, biometric information, which is one of the main targets of fuzzy signatures, is typically not at all uniformly distributed. The same is true for other fuzzy data such as data produced from physically unclonable functions (PUFs).

This motivates us to study whether we can overcome these weaknesses of the fuzzy signature scheme in [15], and come up with a more efficient and easier-to-implement fuzzy signature scheme, while only requiring a more plausible requirement for fuzzy data, e.g. requiring only high min-entropy for the distribution of fuzzy data, which is a necessary requirement (because otherwise a signature can be forged by guessing the fuzzy data used as a signing key).

1.2 Our Contributions

In this paper, we show that in the generic construction of a fuzzy signature scheme shown by Takahashi et al. [15], the requirements on each of the building blocks used in their generic construction, can be relaxed in several aspects: Our relaxation for the underlying linear sketch scheme allows us to use a new linear sketch scheme (that we propose) for a different fuzzy key setting from that of Takahashi et al. As a result, *we only need to require that the (average) min-entropy of the distribution of fuzzy data is high (under the situation where some part of its information is leaked)*. This is our main contribution in this paper. Furthermore, our relaxation on the underlying signature scheme enables us to widen the class of signature schemes that can be used as a building block in the construction by Takahashi et al. In particular, from our relaxations, we can now use the Schnorr signature scheme [14] together with our proposed linear sketch scheme, to obtain a new concrete fuzzy signature scheme. Although our new concrete fuzzy signature scheme is secure only in the random oracle model, unlike the concrete fuzzy signature scheme by Takahashi et al. based on the Waters signature scheme [16], our concrete fuzzy signature scheme based on the Schnorr scheme does not need bilinear groups, is much more efficient, simpler, easier-to-implement, and hence more suitable for practical use, than the scheme in [15]. Below, we explain more technical details of our results.

Relaxing the Requirements on the Linear Sketch Scheme. As mentioned earlier, this primitive can be understood as a pair of linear encoding and error correction method. It is associated with a fuzzy key setting and an abelian group $(\mathcal{K}, +)$, and consists of two algorithms: "Sketch" and "$\mathsf{DiffRec}$" (where the latter stands for "difference reconstruction"). The first algorithm can be used to generate a "sketch" c of an element $s \in \mathcal{K}$ using a fuzzy data x. The second algorithm takes as input two sketches c and c', where c (resp. c') is supposedly a sketch of an element $s \in \mathcal{K}$ (resp. $s' \in \mathcal{K}$) generated by using a fuzzy data x (resp. x'), and outputs the difference $\Delta s = s - s'$ if the two fuzzy data x and x' are "close" (according to the threshold t specified in the fuzzy key setting). It was also required in [15] that a linear sketch scheme satisfies additional "linearity" and "simulatability" properties that are used in the security proof for the generic construction of a fuzzy signature scheme in [15].

In Sect. 5, we introduce four relaxations to the original definition in [15]. (1) We allow a setup algorithm that outputs a public parameter shared by all algorithms in the scheme. (2) We allow the algorithms to be probabilistic. (3) We relax the property called linearity, which is a kind of functional requirement and was originally defined like correctness (without errors), into some distributional notion. (4) We relax the property called simulatability, which is a kind of confidentiality notion, into some average-case indistinguishability-type notion.

Although each relaxation is simple and may not sound so important, the combination of them guides us to constructing a new linear sketch scheme based on a well-known universal hash function family satisfying linearity. The confidentiality (average-case indistinguishability) of our proposed linear sketch scheme

follows from the leftover hash lemma [7,9]. The biggest merit of this linear sketch scheme, compared to the one in [15], is that we can remove the assumption that fuzzy data is distributed uniformly. Interestingly, if any of our four relaxations is not introduced and the previous definition by [15] is used, our construction does not satisfy some of the requirements, and thus our relaxations are actually essential. For more details, see Sect. 5.

Relaxing the Requirements on the Underlying Signature Scheme. In Sect. 6, we show that in the generic construction of a fuzzy signature scheme shown by Takahashi et al. [15], if we can assume that the underlying signature scheme satisfies a notion of security against some kind of related key attack with respect to addition, denoted by Φ^{add}-RKA* security, and formally defined in Sect. 3.2, then one of the homomorphic properties regarding keys (and signatures) required in the construction of [15], can be removed. Interestingly, we show that if a signature scheme satisfies the standard EUF-CMA security and the homomorphic properties required in the construction of [15], then the scheme is automatically Φ^{add}-RKA* secure, while the converse is not necessarily true. Therefore, although our security proof for the generic construction requires a seemingly stronger "RKA" security for the underlying signature scheme, it is in fact a strict relaxation of the security proof by [15], and thus potentially widen the class of signature schemes that can be used as a building block for the generic construction of [15]. As a merit of our "relaxation", we show that the original Schnorr signature scheme [14] can be shown to satisfy the Φ^{add}-RKA* security in the random oracle model. For more details, see Sect. 3.2.

New Security Proof for Takahashi et al.'s Generic Construction from Relaxed Assumptions. In Sect. 6, we show a new security proof for the generic construction of a fuzzy signature scheme by Takahashi et al. [15], from exactly the same primitives, but with our relaxed (and hence weaker) assumptions. More specifically, we prove that if the underlying signature scheme satisfies our RKA security notion and has a relaxed homomorphic property, and the linear sketch scheme satisfies all the relaxed requirements we introduce, then the constructed fuzzy signature scheme is secure. The approach for the proof (e.g. the ordering of games in the sequence of games argument) is very similar to, and proceeds very closely to, the original security proof by [15]. Therefore, our contribution in this security proof is to clarify that a security proof from weaker assumptions is in fact possible, and clarify those assumptions for the underlying signature scheme and the linear sketch scheme. For more details, see Sect. 6.

1.3 Paper Organization

The rest of the paper is organized as follows: In Sect. 2, we review the basic notation, definitions, and facts. In Sect. 3.2, we review definitions for ordinary signatures, and introduce a new RKA security definition. We also show that the Schnorr signature scheme satisfies our RKA security notion. In Sect. 4, we review the definitions for fuzzy signatures, together with the definition of a fuzzy key

setting. In Sect. 5, we introduce our relaxed definitions for a linear sketch. We then specify a concrete fuzzy key setting which requires that the distribution of fuzzy data is only of high (average) min-entropy (in the presence of some kind of leakage), and propose a new construction of a linear sketch scheme. In Sect. 6, we give a new security proof for the generic construction of a fuzzy signature scheme by Takahashi et al. [15], based on our relaxed requirements for the building blocks. In Sect. 7, we give the full description of our Schnorr-based fuzzy signature scheme. In Sect. 8, we discuss the plausibility of our fuzzy key setting, and some open problems.

Due to the space limitation, the proofs of the theorems and lemmas in this paper are omitted and will be given in the full version, and we only give some high-level explanations for them in this proceedings version.

2 Preliminaries

In this section, we recall the basic notation, definitions and facts.

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 $[n] := \{1, \ldots, n\}$. Throughout the paper, we use the bold font to denote a vector (such as $\boldsymbol{x}$ and $\boldsymbol{\alpha}$). If $a \in \mathbb{R}$, then "$\lfloor a \rceil$" denotes the integer that is the nearest to a (the rounding operation). We extend the definition of "$\lfloor \cdot \rceil$" to allow it to take a real vector $\boldsymbol{a} = (a_1, a_2, \ldots)$ as input, by $\lfloor \boldsymbol{a} \rceil := (\lfloor a_1 \rceil, \lfloor a_2 \rceil, \ldots)$.

"$x \leftarrow y$" denotes that y is (deterministically) assigned to x. 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. For a function $f : D \to R$ and an element $y \in R$, we denote by "$f^{-1}(y)$" the set of preimages of y under f, namely, $f^{-1}(y) := \{x \in D | f(x) = y\}$. 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" stands for 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. 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 Basic Definitions Related to Probability and Entropy

Definition 1. *([7]) Let $(\mathcal{X}, \mathcal{Y})$ be a joint distribution defined over the set $X \times Y$. The* average min-entropy of $\mathcal{X}$ *given* $\mathcal{Y}$*, denoted by* $\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y})$*, is defined by* $\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y}) := -\log_2(\mathbf{E}_{y \leftarrow_{\mathtt{R}} \mathcal{Y}}[\max_{x' \in X} \Pr[\mathcal{X} = x' | \mathcal{Y} = y]])$.

Definition 2. *Let $\mathcal{X}$ and $\mathcal{X}'$ be distributions defined over the same set X. The* statistical distance *between $\mathcal{X}$ and $\mathcal{X}'$, denoted by $\mathbf{SD}(\mathcal{X}, \mathcal{X}')$, is defined by $\mathbf{SD}(\mathcal{X}, \mathcal{X}') := \frac{1}{2}\sum_{z \in X} |\Pr[\mathcal{X} = z] - \Pr[\mathcal{X}' = z]|$. We say that $\mathcal{X}$ and $\mathcal{X}'$ are statistically indistinguishable, if $\mathbf{SD}(\mathcal{X}, \mathcal{X}')$ is negligible.*

2.2 Universal Hash Function Family and the Leftover Hash Lemma

Here, we first recall the definition of a universal hash function family, then its concrete construction, and finally the leftover hash lemma [9].

Definition 3. *Let $\mathcal{H} = \{h_z : D \to R\}_{z \in Z}$ be a family of hash functions, where Z denotes the seed space of $\mathcal{H}$. We say that $\mathcal{H}$ is a* universal hash function family *if for all $x, x' \in D$ such that $x \neq x'$, we have $\Pr_{z \leftarrow_{\mathrm{R}} Z}[h_z(x) = h_z(x')] \leq 1/|R|$.*

A Concrete Construction with Linearity. In this paper, we will use the following concrete construction of a universal hash function family $\mathcal{H}_{lin}$ whose domain is $\mathbb{F}_{p^n}$ and whose range is $\mathbb{F}_p$, where $\mathbb{F}_p$ is a finite field with prime order p and $n \in \mathbb{N}$. Note that $\mathbb{F}_{p^n}$, when viewed as a vector space, is isomorphic to the vector space $(\mathbb{F}_p)^n$. Let $\psi : (\mathbb{F}_p)^n \to \mathbb{F}_{p^n}$ be an isomorphism of the vector spaces, and ψ^{-1} be its inverse, which are both computable in time polynomial of $n \cdot \log_2(p)$.

Let the seed space be $Z = \mathbb{F}_{p^n}$, the domain be $D = (\mathbb{F}_p)^n$, and the range be $R = \mathbb{F}_p$. For each $z \in Z$, define the function $h_z : D \to R$ as follows: On input $\boldsymbol{x} \in (\mathbb{F}_p)^n$, $h_z(\boldsymbol{x})$ computes $y \leftarrow \psi(\boldsymbol{x}) \cdot z$, where the operation "$\cdot$" is the multiplication in the extension field $\mathbb{F}_{p^n}$. Let $(y_1, \ldots, y_n) = \psi^{-1}(y)$. The output of $h_z(\boldsymbol{x})$ is $y_1 \in \mathbb{F}_p$. The family $\mathcal{H}_{lin}$ consists of the hash functions $\{h_z\}_{z \in Z}$.

It is well-known (see, e.g. [3]) that $\mathcal{H}_{lin}$ is a universal hash function family. Furthermore, for every $z \in Z$, h_z satisfies linearity, in the following sense:

$$\forall \boldsymbol{x}, \boldsymbol{x}' \in (\mathbb{F}_p)^n \text{ and } \alpha, \beta \in \mathbb{F}_p : \quad \alpha \cdot h_z(\boldsymbol{x}) + \beta \cdot h_z(\boldsymbol{x}') = h_z(\alpha \cdot \boldsymbol{x} + \beta \cdot \boldsymbol{x}').$$

Leftover Hash Lemma. Roughly speaking, the leftover hash lemma [9] states that a universal hash function family is a good (strong) randomness extractor. Here, we recall a version of the leftover hash lemma shown by Dodis et al. [7] that allows leakage from the inputs to a universal hash function.

Lemma 1. *([7]) Let $\mathcal{H} = \{h_z : D \to R\}_{z \in Z}$ be a universal hash function family. Let U_Z and U_R be the uniform distributions over Z and R, respectively. Furthermore, let $(\mathcal{X}, \mathcal{Y})$ be a joint distribution, where the support of $\mathcal{X}$ is contained in D. Then, when z is chosen uniformly as $z \leftarrow_{\mathrm{R}} Z$, it holds that*

$$\mathbf{SD}\Big((z, h_z(\mathcal{X}), \mathcal{Y}), (U_Z, U_R, \mathcal{Y})\Big) \leq \frac{1}{2}\sqrt{2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y})} \cdot |R|}.$$

3 Definitions for (Ordinary) Signatures

In this section, we first review the definitions for (ordinary) signature schemes (Sect. 3.1). We then give the definition of our variant of related key attacks security (which we call $\mathtt{RKA}^*$ security) and some facts on it (Sect. 3.2).

3.1 Structural Properties

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: Setup is the setup algorithm that takes 1^k as input, and outputs a public parameter pp; KG is the key generation algorithm that takes pp as input, and outputs a verification/signing key pair (vk, sk); Sign is the signing algorithm that takes pp, sk, and a message m as input, and outputs a signature σ; Ver is the verification algorithm that takes pp, vk, m, and σ as input, and outputs either $\top$ ("accept") or $\bot$ ("reject").

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 $\mathsf{Ver}(pp, vk, m, \mathsf{Sign}(pp, sk, m)) = \top$.

Simple Key Generation Process. Here we formalize what we call the *simple key generation process* property, which says that the key generation algorithm KG first picks a secret key sk uniformly at random from the secret key space, and then computes the corresponding verification key vk deterministically from sk.[2]

Definition 4. *Let $\Sigma = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme. We say that Σ has a* simple key generation process *if each pp output by Setup specifies a secret key space $\mathcal{K}_{pp}$, and there exists a deterministic PTA KG' such that the key generation algorithm $\mathsf{KG}(pp)$ can be written as follows:*

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

Homomorphic Properties. Here, we review the homomorphic properties regarding keys (and signatures) of a signature scheme used by Takahashi et al. [15]. We also define a weaker version (which we simply call the *weak homomorphic* property) that only requires the first two requirements out of the three, which is sufficient for our security proof in Sect. 6 to go through.

Definition 5. *Let $\Sigma = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme with a simple key generation process (i.e. there is a deterministic PTA KG' in Definition 4). We say that Σ is* homomorphic *if it satisfies the following three properties:*

1. *For all parameters pp output by Setup, the signing key space $\mathcal{K}_{pp}$ constitutes an abelian group $(\mathcal{K}_{pp}, +)$.*
2. *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 for all pp output by Setup and all $sk, \Delta sk \in \mathcal{K}_{pp}$, it holds that

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

[2] Takahashi et al. [15] defined this property as part of the homomorphic property (Definition 5). We separate it for our convenience.

3. *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 for all* pp *output by* Setup*, all messages* m*, and all* $sk, \Delta sk \in \mathcal{K}_{pp}$*, the following two distributions are identical:*

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

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

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

If Σ *satisfies only the first two properties, then we say that* Σ *is* weakly homomorphic.

Schnorr Signature Scheme. Our concrete instantiation of a fuzzy signature scheme is based on the Schnorr signature scheme [14] and thus we review it here. Let GGen be a "group generator" that takes 1^k as input and outputs the description $\mathcal{G} = (\mathbb{G}, p, g)$ of a cyclic group $\mathbb{G} = \langle g \rangle$ with prime order $p = \Theta(2^k)$. Using the group generator GGen as a building block, the Schnorr signature scheme $\Sigma_{\mathsf{Sch}} = (\mathsf{Setup}_{\mathsf{Sch}}, \mathsf{KG}_{\mathsf{Sch}}, \mathsf{Sign}_{\mathsf{Sch}}, \mathsf{Ver}_{\mathsf{Sch}})$ is constructed as in Fig. 1.

It was shown by Pointcheval and Stern [12] that the Schnorr scheme is $\mathtt{EUF\text{-}CMA}$ secure in the random oracle model where H is modeled as a random oracle, under the DL assumption (which requires that given $\mathcal{G} = (\mathbb{G}, p, g)$ and g^x for a randomly chosen $x \in \mathbb{Z}_p$, it is hard to compute x). Furthermore, it should be clear from the relation between a signing key sk and the corresponding verification key $vk = g^{sk}$ that the Schnorr scheme admits a simple key generation process $\mathsf{KG}'(sk) = vk$ with the signing key space $\mathbb{Z}_p$, and furthermore given y and a "shift" Δsk, we can obtain a "shifted" verification key $vk' = \mathsf{KG}'(sk + \Delta sk) := (vk) \cdot g^{\Delta x}$, which results in $g^{sk} \cdot g^{\Delta sk} = g^{sk + \Delta sk} = \mathsf{KG}'(sk + \Delta sk)$. Hence, the following lemma holds:

Lemma 2. *The Schnorr signature scheme* Σ_{Sch} *(in Fig. 1) satisfies the weak homomorphic property in the sense of Definition 5.*

$\mathsf{Setup}_{\mathsf{Sch}}(1^k)$:	$\mathsf{Sign}_{\mathsf{Sch}}(pp, sk, m)$:	$\mathsf{Ver}_{\mathsf{Sch}}(pp, vk, m, \sigma)$:
$\mathcal{G} := (\mathbb{G}, p, g) \leftarrow \mathsf{GGen}(1^k)$	$x \leftarrow sk$	$y \leftarrow vk$
Let $H : \{0,1\}^* \to \mathbb{Z}_p$	$r \leftarrow_{\mathrm{R}} \mathbb{Z}_p$	$(h, s) \leftarrow \sigma$
be a hash function.	$R \leftarrow g^r$	$R \leftarrow g^s \cdot y^{-h}$
Return $pp \leftarrow (\mathcal{G}, H)$.	$h \leftarrow H(R \Vert m)$	If $H(R \Vert m) = h$ then
$\mathsf{KG}_{\mathsf{Sch}}(pp)$:	$s \leftarrow r + x \cdot h \bmod p$	return $\top$ else return $\bot$.
$x \leftarrow_{\mathrm{R}} \mathbb{Z}_p$; $y \leftarrow g^x$	Return $\sigma \leftarrow (h, s)$.	
Return $(vk, sk) \leftarrow (y, x)$.		

Fig. 1. The Schnorr signature scheme Σ_{Sch}.

3.2 A Variant of Related Key Attacks Security

Φ-RKA* *Security.* Here, we introduce an extension of EUF-CMA security for signature schemes, which we call Φ-RKA* security[3], that considers security against an adversary who may mount a kind of related-key attacks (RKA). Like the popular definition of RKA security for signature schemes by Bellare et al. [1], Φ-RKA* is defined with respect to a class of functions that captures an adversary's ability to modify signing keys. Our definition, however, has subtle differences from the definition of [1]. The main difference is that in our definition, an adversary is allowed to modify the verification key under which its forgery is verified, while we do not allow an adversary to use a message as its forgery if it is already signed by the signing oracle.

Formally, let $\Sigma = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme with a simple key generation process, namely, there exists a deterministic PTA KG' such that KG can be written as Eq. (1). Let Φ be a class of functions both of whose domain and the range are the signing key space of Σ. For Σ, Φ, and an adversary $\mathcal{A}$, consider the following Φ-RKA* experiment $\mathsf{Expt}^{\Phi\text{-RKA}^*}_{\Sigma,\mathcal{A}}(k)$:

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

where $\mathcal{O}_{\mathsf{Sign}}$ is the signing oracle which takes (the description of) a function $\phi \in \Phi$ and a message m as input, updates the "used message list" $\mathcal{Q}$ by $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{m\}$, and returns a signature $\sigma \leftarrow_{\mathtt{R}} \mathsf{Sign}(pp, \phi(sk), m)$.

Definition 6. *We say that a signature scheme Σ is Φ-RKA* secure if for all PPTA adversaries $\mathcal{A}$, $\mathsf{Adv}^{\Phi\text{-RKA}^*}_{\Sigma,\mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\Phi\text{-RKA}^*}_{\Sigma,\mathcal{A}}(k) = 1]$ is negligible.*

Note that if we consider Φ to be consisting only of the identity function, then we recover the standard EUF-CMA security for a signature scheme.

The Class of Functions. In this paper, we will treat Φ-RKA* security with respect to addition, which is captured by the following functions (where $\mathcal{K}$ denotes the signing key space of a signature scheme that we assume constitutes an abelian group):

Addition: $\Phi^{\mathrm{add}} := \{\phi^{\mathrm{add}}_a | a \in \mathcal{K}\}$, where $\phi^{\mathrm{add}}_a(x) = x + a$.

Sufficient Conditions for Φ^{add}-RKA* Security. It turns out that any EUF-CMA secure signature scheme (with a simple key generation process) that satisfies the three requirements of the homomorphic property (Definition 5) is automatically Φ^{add}-RKA* secure, and hence these are sufficient conditions for Φ^{add}-RKA* security. (Due to the space limitation, we provide its proof in the full version.)

[3] The asterisk (*) indicates that the notion is different from that of Bellare et al. [1].

Theorem 1. *Any* EUF-CMA *secure signature scheme (with a simple key generation process) satisfying the homomorphic property (Definition 5) is* Φ^{add}-RKA* *secure.*

Φ^{add}-RKA* *Security of the Schnorr Signature Scheme.* As we mentioned in the previous subsection, the Schnorr signature scheme Σ_{Sch} admits a simple key generation process, and is weakly homomorphic, where its signing key space is the abelian group $(\mathbb{Z}_p, +)$. The following theorem formally states that the Schnorr signature scheme satisfies Φ^{add}-RKA* security. The proof can be shown very similarly to the proof of the EUF-CMA security of the Schnorr scheme using the general forking lemma of Bellare and Neven [2], and its Φ^{add}-weak-RKA security shown in [10]. We provide its proof in the full version.

Theorem 2. *If the DL assumption holds with respect to* GGen*, then the Schnorr signature scheme* Σ_{Sch} *(in Fig. 1) is* Φ^{add}-RKA* *secure in the random oracle model where* H *is modeled as a random oracle.*

4 Definitions for Fuzzy Signatures

In this section, we recall the definitions of a fuzzy key setting (Sect. 4.1) and a fuzzy signature scheme (Sect. 4.2), both of which are from [15].

4.1 Formalization of Fuzzy Key Setting

A fuzzy key setting specifies a metric space to which fuzzy data (such as biometric data) belongs, the threshold with which two sampled fuzzy data are considered close/far, the distribution from which each fuzzy data is assumed to be sampled, and the error distribution that models "fluctuation" of fuzzy data. The false acceptance rate (FAR) and the false rejection rate (FRR), are also defined. The formalization of [15] adopts the so-called *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 (e.g. at the registration), and the same object is measured next time, then the measured data x' follows the distribution $\{e \leftarrow_{\mathrm{R}} \Phi;\ x' \leftarrow x + e : x'\}$. (That is, Φ is the same, regardless of individual U.)

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 $\mathtt{FAR} := \Pr[x, x' \leftarrow_{\mathrm{R}} \mathcal{X} : \mathsf{d}(x, x') < t]$ 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 $x \in X$, $\mathtt{FRR} := \Pr[e \leftarrow_{\mathrm{R}} \Phi : \mathsf{d}(x, x + e) \geq t] \leq \epsilon$.

4.2 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 that 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 $\mathsf{Ver}_{\mathsf{FS}}(pp, \mathsf{KG}_{\mathsf{FS}}(pp, x), m, \mathsf{Sign}_{\mathsf{FS}}(pp, x', m)) = \top$.

$\mathtt{EUF\text{-}CMA}$ *Security.* $\mathtt{EUF\text{-}CMA}$ security of a fuzzy signature scheme is defined in a similar manner to that of 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 $\mathsf{Expt}^{\mathtt{EUF\text{-}CMA}}_{\Sigma_{\mathsf{FS}}, \mathcal{F}, \mathcal{A}}(k)$:

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

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 $e \leftarrow_{\mathtt{R}} \Phi$, computes a signature $\sigma \leftarrow_{\mathtt{R}} \mathsf{Sign}_{\mathsf{FS}}(pp, x + e, m)$, and returns σ.

Definition 7. *We say that a fuzzy signature scheme* Σ_{FS} *is* $\mathtt{EUF\text{-}CMA}$ *secure if for all PPTA adversaries* $\mathcal{A}$, $\mathsf{Adv}^{\mathtt{EUF\text{-}CMA}}_{\Sigma_{\mathsf{FS}}, \mathcal{F}, \mathcal{A}}(k) := \Pr[\mathsf{Expt}^{\mathtt{EUF\text{-}CMA}}_{\Sigma_{\mathsf{FS}}, \mathcal{F}, \mathcal{A}}(k) = 1]$ *is negligible.*

5 Linear Sketch

In this section, we introduce our new definitions for the primitive called *linear sketch* that was first formalized by Takahashi et al. [15], which plays an important role in the generic construction in [15]. We then propose a new construction of a linear sketch for a concrete fuzzy key setting in which the distribution of fuzzy data has high average min-entropy (in the presence of leakage).

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 with respect to the encoded version of the input.)

5.1 Our Relaxed Definition

Informally speaking, a linear sketch is associated with a fuzzy key setting and an abelian group $(\mathcal{K}, +)$, and consists of two algorithms: "Sketch" and "$\mathsf{DiffRec}$" whose functionalities are explained shortly. It was also required in [15] that a linear sketch scheme satisfies additional "linearity" and "simulatability" properties that are used in the security proof for the generic construction of a fuzzy signature scheme in [15].

We introduce four relaxations to the original definition in [15]: **(1)** We introduce a setup algorithm that produces a public parameter, which is used by the two main algorithms Sketch and $\mathsf{DiffRec}$, and also by the auxiliary algorithm $\mathsf{M_c}$ that is used for defining "linearity"; **(2)** We allow the sketching algorithm Sketch, and the auxiliary algorithm $\mathsf{M_c}$, to be probabilistic (as opposed to defining them as deterministic algorithms in [15]); **(3)** We relax the linearity property to some weaker "distributional" variant, while in [15] it was defined like correctness that needs to be satisfied without any failure; **(4)** We relax the simulatability property, which captures confidentiality of sketches produced by Sketch, to a weaker variant that we call "average-case indistinguishability".

Formally, our definition of a linear sketch scheme is as follows:

Definition 8. *Let $\mathcal{F} = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \epsilon)$ be a fuzzy key setting. We say that a tuple of PPTAs $\mathcal{S} = (\mathsf{Setup}, \mathsf{Sketch}, \mathsf{DiffRec})$ is a* linear sketch *scheme for $\mathcal{F}$, if it satisfies the following three properties:*

Syntax and Correctness: *Each algorithm of $\mathcal{S}$ has the following interface:*

- Setup *is the "setup" algorithm that takes the description Λ of an abelian group $(\mathcal{K}, +)$ as input, and outputs a public parameter pp (which we assume contains the information of Λ).*
- Sketch *is the "sketching" algorithm that takes pp, an element $s \in \mathcal{K}$, and a fuzzy data $x \in X$ as input, and outputs a "sketch" c.*
- $\mathsf{DiffRec}$ *is the (deterministic) "difference reconstruction" algorithm that takes pp and two values c, c' (supposedly output by Sketch) as input, and outputs the "difference" $\Delta s \in \mathcal{K}$.*

We require that for all $x, x' \in X$ such that $\mathsf{d}(x, x') < t$, all (descriptions of) abelian groups $\Lambda = (\mathcal{K}, +)$, all pp output by $\mathsf{Setup}(\Lambda)$, and all $s, \Delta s \in \mathcal{K}$, it holds that $\mathsf{DiffRec}(pp, \mathsf{Sketch}(pp, s, x), \mathsf{Sketch}(pp, s + \Delta s, x')) = \Delta s$.

Linearity: *There exists a PPTA $\mathsf{M_c}$ satisfying the following: For all abelian groups $\Lambda = (\mathcal{K}, +)$, all pp output by $\mathsf{Setup}(\Lambda)$, all $x, e \in X$ such that $\mathsf{d}(x, x + e) < t$, and for all $s, \Delta s \in \mathcal{K}$, the following two distributions are statistically indistinguishable (in the security parameter k that is associated with t in $\mathcal{F}$):*

$$\{c \leftarrow_{\mathrm{R}} \mathsf{Sketch}(pp, s, x);\ c' \leftarrow_{\mathrm{R}} \mathsf{Sketch}(pp, s + \Delta s, x + e) : (c, c')\}, \quad \text{and}$$
$$\{c \leftarrow_{\mathrm{R}} \mathsf{Sketch}(pp, s, x);\ c' \leftarrow_{\mathrm{R}} \mathsf{M_c}(pp, c, \Delta s, e) : (c, c')\} \tag{5}$$

Average-Case Indistinguishability: *For all abelian groups $\Lambda = (\mathcal{K}, +)$, the following two distributions are statistically indistinguishable (in the security parameter k that is associated with t in $\mathcal{F}$):*

$$\{pp \leftarrow_{\mathrm{R}} \mathsf{Setup}(\Lambda);\ x \leftarrow_{\mathrm{R}} \mathcal{X};\ s \leftarrow_{\mathrm{R}} \mathcal{K};\ c \leftarrow_{\mathrm{R}} \mathsf{Sketch}(pp, s, x) : (pp, s, c)\},\quad \textit{and}$$
$$\{pp \leftarrow_{\mathrm{R}} \mathsf{Setup}(\Lambda);\ x \leftarrow_{\mathrm{R}} \mathcal{X};\ s, s' \leftarrow_{\mathrm{R}} \mathcal{K};\ c \leftarrow_{\mathrm{R}} \mathsf{Sketch}(pp, s, x) : (pp, s', c)\} \tag{6}$$

Here are a couple of remarks:

- The word "average-case" of average-case indistinguishability is due to the property that its definition guarantees that the element s in a sketch c is hidden only when it is chosen randomly from $\mathcal{K}$.
- As mentioned just above Definition 8, our definition is obtained by relaxing the definition in [15] in several regards. (In the full version, we provide the original definitions for a linear sketch given in [15] for a comparison.) In particular, we can cast any linear sketch that satisfies the definition in [15] by defining the public parameter pp to be the description of an abelian group Λ itself: Then, the linearity property (resp. simulatability) in the sense of [15] implies the linearity property (resp. average-case indistinguishability) in our definition.

5.2 Our New Construction

Here, we propose a new construction of a linear sketch scheme for a concrete fuzzy key setting. We first specify the fuzzy key setting with which our scheme is associated, and then give our construction.

Specific Fuzzy Key Setting. Here, we specify a concrete fuzzy key setting $\mathcal{F} = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \epsilon)$ for which our linear sketch scheme and our Schnorr-based fuzzy signature scheme are constructed.

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

Threshold t**:** For a security parameter k, we require the threshold $t \in \mathbb{R}$, where $(1/(2t)) \in \mathbb{N}$, to satisfy

$$k \leq \lfloor -n \log_2(2t) \rfloor. \tag{7}$$

Distribution $\mathcal{X}$**:** An efficiently samplable distribution over X that satisfies the assumption on the average min-entropy that we state later.

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 $x \in X$.

Other than the requirement on $\mathcal{X}$, the above specification of the fuzzy key setting is essentially the same as the one used in [15].[4] Takahashi et al. required $\mathcal{X}$

[4] Actually, [15] set the security parameter k to be exactly $\lfloor -n \log_2(2t) \rfloor$. However, we need more strict threshold for t, so that we can use the leftover hash lemma in the proof of Theorem 3 (given in the full version).

to be the uniform distribution. However, this is a somewhat strong requirement, and may not be suitable for potential applications of fuzzy signature schemes. In this work, we succeed in relaxing the requirement on $\mathcal{X}$, from the uniform distribution to a distribution with sufficiently high average min-entropy.

More specifically, let $\mathcal{X}'$ be the "scaled-up" version of $\mathcal{X}$, namely, $\mathcal{X}'$ is the distribution obtained by multiplying the integer $1/(2t) \in \mathbb{N}$ to the outcome of the distribution $\mathcal{X}$. Since $\mathcal{X}$ is a distribution over $[0,1)^n$, $\mathcal{X}'$ is a distribution over $[0, 1/(2t))^n$. Now, let us divide $\mathcal{X}'$ into the "integer" part $\mathcal{X}'_{in}$ and the "decimal" part $\mathcal{X}'_{de}$. Namely, let $\boldsymbol{x}' = (x'_1, \ldots, x'_n)$ be a vector produced from $\mathcal{X}'$. Then, $\mathcal{X}'_{in}$ is the distribution of the n-dimensional vector whose i-th element is the integer part of x'_i. Similarly, $\mathcal{X}'_{de}$ is the distribution of the n-dimensional vector whose i-th element is the decimal part of x'_i. Note that the joint distribution $(\mathcal{X}'_{in}, \mathcal{X}'_{de})$ contains the same information as $\mathcal{X}'$ (and hence as $\mathcal{X}$).

The requirement we impose on the distribution $\mathcal{X}$ of fuzzy data, is that we have $\widetilde{\mathbf{H}}_\infty(\mathcal{X}'_{in}|\mathcal{X}'_{de}) \geq \log p + \omega(\log k)$, where p is the order of the field over which we consider the universal hash $\mathcal{H}_{lin}$. (We note that $\widetilde{\mathbf{H}}_\infty(\mathcal{X}'_{in}|\mathcal{X}'_{de}) = \widetilde{\mathbf{H}}_\infty(\mathcal{X}'|\mathcal{X}'_{de}) = \widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{X}'_{de})$. Furthermore, since p will also be the order of the group over which the Schnorr scheme is constructed, we typically set $p = \Theta(2^k)$, equivalently $\log p = \Theta(k)$.) We would like to emphasize that this requirement is arguably much more relaxed than requiring that $\mathcal{X}$ is the uniform distribution over X (which was done in [15]). We discuss the plausibility of this requirement later in Sect. 8.

Our Construction. Let $\mathcal{F} = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \epsilon)$ be the fuzzy key setting as defined above. Let $\mathbb{F}_p$ be a finite field with prime order p satisfying $p \geq 1/(2t)$. Here, we identify $\mathbb{F}_p$ with $\mathbb{Z}_p$, and thus we freely interpret an element in the former set as an element in the latter set, and vice versa. Let $\mathcal{H}_{lin} = \{\ h_z : (\mathbb{F}_p)^n \to \mathbb{F}_p\}_{z \in \mathbb{F}_{p^n}}$ be the universal hash function family with linearity that we reviewed in Sect. 2. For each $z \in \mathbb{F}_{p^n}$ and $s \in \mathbb{F}_p$, we define "$h_z^{-1}(s)$" as the set of preimages of s under h_z. That is, $h_z^{-1}(s) := \{\boldsymbol{\alpha} \in (\mathbb{F}_p)^n | h_z(\boldsymbol{\alpha}) = s\}$. Hence, the notation "$\boldsymbol{\alpha} \leftarrow_{\mathrm{R}} h_z^{-1}(s)$" means that we choose a vector $\boldsymbol{\alpha}$ uniformly from the set $h_z^{-1}(s)$ (which can be performed in time polynomial of $n \cdot \log_2(p)$). Furthermore, for notational convenience, let $T := 1/(2t) \in \mathbb{N}$.

Then, using these, our linear sketch scheme $\mathcal{S} = (\mathsf{Setup}, \mathsf{Sketch}, \mathsf{DiffRec})$ for $\mathcal{F}$ and the additive group $(\mathbb{Z}_p, +)$ (=: Λ) is constructed as described in Fig. 2 (left), where we also give a description of the auxiliary algorithm $\mathsf{M_c}$ for convenience. The output space of Sketch is $(\mathbb{R}_p)^n$, where $\mathbb{R}_p := \mathbb{R}/p\mathbb{R}$.

The following guarantees that our construction satisfies all the requirements.

Theorem 3. *The linear sketch scheme $\mathcal{S}$ in Fig. 2 (left) for the fuzzy key setting $\mathcal{F}$ that we specified above, satisfies all the properties of Definition 8.*

Due to the space limitation, we provide the formal proof in the full version. Roughly speaking, the correctness follows from the linearity of the universal hash family $\mathcal{H}_{lin}$ and a simple algebra; The linearity property of $\mathcal{S}$ follows from

$\mathsf{Setup}(\Lambda = (\mathbb{Z}_p, +))$:
$z \leftarrow_{\mathtt{R}} \mathbb{F}_{p^n}$; $pp \leftarrow (\Lambda, z)$
Return pp.

$\mathsf{Sketch}(pp, s, \boldsymbol{x})$: (where $s \in \mathbb{Z}_p$ and $\boldsymbol{x} \in [0,1)^n$)
$\boldsymbol{\alpha} \leftarrow_{\mathtt{R}} h_z^{-1}(s)$; $\boldsymbol{c} \leftarrow \boldsymbol{\alpha} + T \cdot \boldsymbol{x}$ (†)
Return $\boldsymbol{c} \in (\mathbb{R}_p)^n$.

$\mathsf{DiffRec}(pp, \boldsymbol{c}, \boldsymbol{c}')$:
$\Delta\boldsymbol{c} \leftarrow \boldsymbol{c}' - \boldsymbol{c}$ (†); $\Delta s \leftarrow h_z(\lfloor \Delta\boldsymbol{c} \rceil)$
Return $\Delta s \in \mathbb{F}_p$.

$\mathsf{M_c}(pp, \boldsymbol{c}, \Delta s, \boldsymbol{e})$:
$\Delta\boldsymbol{\alpha} \leftarrow_{\mathtt{R}} h_z^{-1}(\Delta s)$
$\boldsymbol{c}' \leftarrow (\boldsymbol{c} + \Delta\boldsymbol{\alpha} + T \cdot \boldsymbol{e})$ (†)
Return $\boldsymbol{c}' \in (\mathbb{R}_p)^n$.

Fig. 2. Our proposed linear sketch scheme $\mathcal{S} = (\mathsf{Setup}, \mathsf{Sketch}, \mathsf{DiffRec})$ for the fuzzy key setting $\mathcal{F}$ (left), and the auxiliary algorithm $\mathsf{M_c}$ for showing the linearity property (right). (†) The operation "+" (resp. "−") in $(\mathbb{R}_p)^n$ are the coordinate-wise addition (resp. subtraction) in $\mathbb{R}_p$.

the linearity of $\mathcal{H}_{lin}$ and the simple observation that $\{\boldsymbol{\alpha} \leftarrow_{\mathtt{R}} h_z^{-1}(s); \Delta\boldsymbol{\alpha} \leftarrow_{\mathtt{R}} h_z^{-1}(\Delta s) : \boldsymbol{\alpha} + \Delta\boldsymbol{\alpha}\}$ yields the uniform distribution over the set $h_z^{-1}(s + \Delta s)$ for any $z \in \mathbb{F}_{p^n}$ and $s, \Delta s \in \mathbb{F}_p$; The high-level ideas for the proof for the average-case indistinguishability are as follows: Note that the distribution $D = \{z \leftarrow_{\mathtt{R}} \mathbb{F}_{p^n}; \boldsymbol{x} \leftarrow_{\mathtt{R}} \mathcal{X}; s \leftarrow_{\mathtt{R}} \mathbb{F}_p; \boldsymbol{\alpha} \leftarrow_{\mathtt{R}} h_z^{-1}(s); \boldsymbol{c} \leftarrow \boldsymbol{\alpha} + T \cdot \boldsymbol{x} : (z, s, \boldsymbol{c})\}$, which corresponds to the first distribution in Eq. (6), is equivalent to $D' = \{z \leftarrow_{\mathtt{R}} \mathbb{F}_{p^n}; \boldsymbol{x} \leftarrow_{\mathtt{R}} \mathcal{X}; \boldsymbol{\alpha} \leftarrow_{\mathtt{R}} (\mathbb{F}_p)^n; \boldsymbol{c} \leftarrow \boldsymbol{\alpha} + T \cdot \boldsymbol{x} : (z, s = h_z(\boldsymbol{\alpha}), \boldsymbol{c})\}$. Now, define the joint distribution $(A, C) := \{x \leftarrow_{\mathtt{R}} \mathcal{X}; \boldsymbol{\alpha} \leftarrow_{\mathtt{R}} (\mathbb{F}_p)^n; \boldsymbol{c} \leftarrow \boldsymbol{\alpha} + T \cdot \boldsymbol{x} : (\boldsymbol{\alpha}, \boldsymbol{c})\}$. In the full proof, we show that $\widetilde{\mathbf{H}}_\infty(A|C) = \widetilde{\mathbf{H}}_\infty(\mathcal{X}'_{in}|\mathcal{X}'_{de})$. This, together with our requirement on $\mathcal{X}$, allows us to invoke the leftover hash lemma to conclude that D' is statistically close to some distribution D''. We will then show that this D'' is equivalent to the distribution corresponding to the second one in Eq. (6).

6 Generic Construction and Our New Security Proof

In this section, we revisit the generic construction for a fuzzy signature scheme by Takahashi et al. [15], which uses a linear sketch and a signature scheme as building blocks, and show its new security proof.

The Generic Construction by Takahashi et al. [15]. Let $\mathcal{F} = ((d, X), t, \mathcal{X}, \Phi, \epsilon)$ be a fuzzy key setting, and let $\mathcal{S} = (\mathsf{Setup}_l, \mathsf{Sketch}, \mathsf{DiffRec})$ be a linear sketch for $\mathcal{F}$. Let $\Sigma = (\mathsf{Setup}_s, \mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$ be a signature scheme with a simple key generation process (i.e. there exists a deterministic PTA KG'). We assume that Σ is weakly homomorphic (as per Definition 5), namely, its signing key space (given pp) is an abelian group $(\mathcal{K}_{pp}, +)$, and has the additional algorithm $\mathsf{M_{vk}}$. Using $\mathcal{S}$ and Σ, the generic construction of a 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 $\mathcal{F}$, originally proposed by Takahashi et al. [15], is constructed as in Fig. 3.

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

Fig. 3. The generic construction of a fuzzy signature scheme Σ_{FS} for a fuzzy key setting $\mathcal{F}$ by Takahashi et al. [15], which combines a linear sketch scheme $\mathcal{S}$ for $\mathcal{F}$ and a weakly homomorphic signature scheme Σ.

Our New Security Proof. Takahashi et al. [15] required that the underlying signature scheme Σ to be homomorphic (not just weak one) and EUF-CMA secure. Here, we show that if we can assume the Φ^{add}-RKA* security for Σ, then we only need to require it to satisfy the "weak homomorphic property" (which does not require the algorithm $\mathsf{M}_{\mathsf{sig}}$) in Definition 5. Our result is in fact a relaxation of the requirements in Takahashi et al.'s construction, because as we showed in Theorem 1, an EUF-CMA secure signature scheme that satisfies the homomorphic property is automatically Φ^{add}-RKA* secure, while a Φ^{add}-RKA* secure signature scheme is not necessarily homomorphic.

Theorem 4. *If Σ is weakly homomorphic and is Φ^{add}-RKA* secure, and $\mathcal{S}$ is a linear sketch scheme for $\mathcal{F}$ (in the sense of Definition 8), then the fuzzy signature scheme Σ_{FS} for $\mathcal{F}$ constructed as in Fig. 3 is* EUF-CMA *secure.*

Due to the space limitation, we give the formal proof in the full version. As mentioned earlier, our security proof follows very similarly to that of [15]. Our proof is via the sequence of games argument. We gradually change the original EUF-CMA security experiment for an adversary $\mathcal{A}$ against our construction of a fuzzy signature scheme by using the weakly homomorphic property of Σ, and the linearity property and average-case indistinguishability of $\mathcal{S}$, so that if $\mathcal{A}$ is still successful in the final game, we can use $\mathcal{A}$ to break the Φ^{add}-RKA* security of the underlying signature scheme Σ. The main difference from the security proof in [15] is that the Φ^{add}-RKA* security of Σ allows us to combine two of the games in the sequence of the games considered in the security proof in [15] in one step. For the details, see the proof in the full version.

7 Instantiation

Here, we show a concrete instantiation of a fuzzy signature scheme by using the Schnorr signature scheme (Fig. 1) and the linear sketch scheme proposed in Sect. 5.2 as the building blocks in the generic construction in Sect. 6.

$\mathsf{Setup}_{\mathsf{FS}}(\mathcal{F}, 1^k)$:	$\mathsf{Sign}_{\mathsf{FS}}(pp, \boldsymbol{x}', m)$:	$\mathsf{Ver}_{\mathsf{FS}}(pp, VK, m, \sigma)$:
$\mathcal{G} := (\mathbb{G}, p, g) \leftarrow \mathsf{GGen}(1^k)$	$\widetilde{sk} \leftarrow_{\mathtt{R}} \mathbb{Z}_p$	$(vk, \boldsymbol{c}) \leftarrow VK$
Let $H : \{0,1\}^* \to \mathbb{Z}_p$	$\widetilde{vk} \leftarrow g^{\widetilde{sk}}$	$(\widetilde{vk}, \widetilde{h}, \widetilde{s}, \widetilde{\boldsymbol{c}}) \leftarrow \sigma$
be a hash function.	$r \leftarrow_{\mathtt{R}} \mathbb{Z}_p$	$R \leftarrow g^{\widetilde{s}} \cdot (vk)^{-\widetilde{h}}$
$z \leftarrow_{\mathtt{R}} \mathbb{F}_{p^n}$	$R \leftarrow g^r$	If $H(R\|m) \neq \widetilde{h}$ then return $\perp$.
Return $pp \leftarrow (\mathcal{G}, z, H)$.	$\widetilde{h} \leftarrow H(R\|m)$	$\Delta\boldsymbol{c} \leftarrow \widetilde{\boldsymbol{c}} - \boldsymbol{c}$ (†)
$\mathsf{KG}_{\mathsf{FS}}(pp, \boldsymbol{x})$:	$\widetilde{s} \leftarrow r + x \cdot \widetilde{h} \bmod p$	$\Delta sk \leftarrow h_s(\lfloor \Delta\boldsymbol{c} \rceil)$
$sk \leftarrow_{\mathtt{R}} \mathbb{Z}_p$	$\boldsymbol{\alpha}' \leftarrow_{\mathtt{R}} h_z^{-1}(\widetilde{sk})$	If $vk \cdot g^{\Delta sk} = \widetilde{vk}$ then
$vk \leftarrow g^{sk}$	$\widetilde{\boldsymbol{c}} \leftarrow \boldsymbol{\alpha}' + T \cdot \boldsymbol{x}'$ (†)	return $\top$ else return $\perp$.
$\boldsymbol{\alpha} \leftarrow_{\mathtt{R}} h_z^{-1}(sk)$	$\sigma \leftarrow (\widetilde{vk}, \widetilde{h}, \widetilde{s}, \widetilde{\boldsymbol{c}})$.	
$\boldsymbol{c} \leftarrow \boldsymbol{\alpha} + T \cdot \boldsymbol{x}$ (†)	Return σ.	
Return $VK \leftarrow (vk, \boldsymbol{c})$.		

Fig. 4. The proposed Schnorr-based fuzzy signature scheme. (†) The operation "+" (resp. "−") in $(\mathbb{R}_p)^n$ are the coordinate-wise addition (resp. subtraction) in $\mathbb{R}_p$.

Let $\mathcal{F} = ((d, X), t, \mathcal{X}, \Phi, \epsilon)$ be the fuzzy key setting we specified in Sect. 5, and suppose the dimension of the fuzzy data space is n. Let GGen be a group generator (which we assume to produce a description of a group whose order is p). Let $\mathcal{H}_{lin} = \{h_z : (\mathbb{F}_p)^n \to \mathbb{F}_p\}_{z \in \mathbb{F}_{p^n}}$ be the universal hash family with linearity that we introduce in Sect. 2. (As in previous sections, we identify $\mathbb{F}_p$ with $\mathbb{Z}_p$.) Let $H : \{0,1\}^* \to \mathbb{Z}_p$ be a cryptographic hash function. Using these building blocks, we construct a 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 $\mathcal{F}$ as in Fig. 4.

The following statement on security is obtained as a corollary of Theorems 2, 3 and 4, and Lemma 2.

Theorem 5. *If the DL assumption holds with respect to* GGen*, then the fuzzy signature scheme* Σ_{FS} *in Fig. 4 is* $\mathtt{EUF\text{-}CMA}$ *secure in the random oracle model where* H *is modeled as a random oracle.*

Although our scheme is secure only in the random oracle model due to the reliance on the Schnorr scheme, it has several practical advantages compared to the concrete instantiation based on the Waters signature scheme shown in [15]: Our scheme does not require bilinear maps, and the verification key size can be much shorter than that in [15]. More importantly, our scheme works for the fuzzy key setting in which fuzzy data cannot be assumed to be distributed uniformly over the data space (which was required in [15]), but that only its average min-entropy (given some parts of the fuzzy data) is sufficiently high.

8 Discussion

On the Plausibility of Our Requirement on the Distribution of Fuzzy Data. As we have seen in the previous sections, in this work we have succeeded in relaxing the requirement on the distribution of fuzzy data than the one required by Takahashi et al. [15], and have given a more efficient concrete instantiation of

a fuzzy signature scheme based on the Schnorr scheme, which is secure in the random oracle model under the DL assumption.

A natural question would be whether practical fuzzy key settings can satisfy our requirement, especially the requirement on the average min-entropy in the presence of leakage (the "decimal" part of the "scaled-up version" of fuzzy data, $\widetilde{\mathbf{H}}_\infty(\mathcal{X}'_{in}|\mathcal{X}'_{de})$ in our notation). In the biometric setting, which is one of the main motivations for considering fuzzy signature schemes (and thus is one of the most important settings that should be captured by the formalization of fuzzy data settings), a well-known approach to measure the biometric entropy is *discrimination entropy* proposed by Daugman [5]. He considered a distribution of a Hamming distance m between two iriscodes (well-known iris features [6]) 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}. This is a positive news for us, and for the future of related research.

However, of course, that the probability of two different iriscodes matching is approximated as 2^{-249}, does not necessarily mean that using iriscode x as fuzzy data gives us 249-bit security. Especially, in our case, we need to take into account the leakage (information leaked from the "decimal" part $\mathcal{X}'_{de}$), when the data is cast into our setting. We have to choose the threshold t by taking into account various other things, such as FAR and FRR. (Note that an 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.) Therefore, it seems not so easy to use the results from [5,6] just as it is.

If a single biometric feature does not have enough entropy, then one of the promising solutions to the problem would be to combine multiple biometric features. For example, Murakami et al. [11] recently showed that by combining four finger-vein features, $\mathtt{FAR} = 2^{-133}$ (resp. $\mathtt{FAR} = 2^{-87}$) can be achieved in the case when $\mathtt{FRR} = 0.055$ (resp. $\mathtt{FRR} = 0.0053$). Also, a multibiometric sensor that simultaneously acquires multiple biometrics (e.g. iris and face [4]; fingerprint and finger-vein [13]) has also been widely developed. Thus, we believe that using multiple biometrics is a promising direction for increasing entropy without affecting usability (which is also an important factor in practice).

It is also important to note that (an approximation of) $\widetilde{\mathbf{H}}_\infty(\mathcal{X}'_{in}|\mathcal{X}'_{de})$ could be experimentally estimated by using real fuzzy data (in a similar manner done in [11]). This is an important feature in order for fuzzy signature schemes (and security systems based on them) to be used in practice.

Open Problems. It would be important to tackle the problem of whether we can realize the fuzzy key setting required in our work by some practical biometric settings/systems. It is also worth tackling whether further relaxing the requirement than our specific fuzzy key setting is possible, or considering settings that are different from ours. For example, can we construct a fuzzy signature scheme with other types of metric spaces (e.g. Euclid distance, hamming distance, edit

distance, etc.)? It would also be worth clarifying whether we can construct more fuzzy signature schemes based on other existing signature schemes.

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

References

1. Bellare, M., Cash, D., Miller, R.: Cryptography secure against related-key attacks and tampering. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 486–503. Springer, Heidelberg (2011)
2. Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: CCS 2006, pp. 390–399 (2006)
3. Cheraghchi, M.: Capacity achieving codes from randomness condensers (2011). http://arxiv.org/pdf/0901.1866v2.pdf. Preliminary version appeared in ISIT 2009
4. Connaughton, R., Bowyer, K.W., Flynn, P.J.: Fusion of face and iris biometrics, Chapter 12. In: Burge, M.J., Bowyer, K.W. (eds.) Handbook of Iris Recognition, pp. 219–237. Springer, London (2013)
5. Daugman, J.: The importance of being random: Statistical principles of iris recognition. Pattern Recogn. **36**(2), 279–291 (2003)
6. Daugman, J.: How iris recognition works. IEEE Trans. Circ. Syst. Video Technol. **14**, 21–30 (2004)
7. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008)
8. 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)
9. Håstad, J., Impagliazzo, R., Levin, L., Luby, M.: Construction of a pseudorandom generator from any one-way function. SIAM J. Comput. **28**(4), 1364–1396 (1999)
10. Morita, H., Schuldt, J.C.N., Matsuda, T., Hanaoka, G., Iwata, T.: On the security of the schnorr signature scheme and DSA against related-key attacks. In: Kwon, S., Yun, A. (eds.) ICISC 2015. LNCS, vol. 9558, pp. 20–35. Springer, Heidelberg (2016). doi:10.1007/978-3-319-30840-1_2
11. Murakami, T., Ohki, T., Takahashi, K.: Optimal sequential fusion for multibiometric cryptosystems. Elsevier Information Fusion (To appear)
12. Pointcheval, D., Stern, J.: Security proofs for signature schemes. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 387–398. Springer, Heidelberg (1996)
13. Raghavendra, R., Raja, K.B., Surbiryala, J., Busch, C.: A low-cost multimodal biometric sensor to capture finger vein and fingerprint. In: IJCB 2014, pp. 1–7 (2014)
14. Schnorr, C.-P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 239–252. Springer, Heidelberg (1990)
15. 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., Polychronakis, M. (eds.) ACNS 2015. LNCS, vol. 9092, pp. 105–126. Springer, Heidelberg (2015). doi:10.1007/978-3-319-28166-7_6
16. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)