On the Security of Linear Sketch Schemes against Recovering Attacks

Takahiro Matsuda[1], Kenta Takahashi[2] and Goichiro Hanaoka[1]
[1]*National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan*
[2]*Hitachi Ltd., Yokohama, Japan*

Keywords: Fuzzy Signature, Linear Sketch, Recovering Attacks.

Abstract: Recently, the notion of *fuzzy signature* was introduced by Takahashi et al. (ACNS 2015, ACNS 2016, ePrint 2017). It is a signature scheme in which signatures can be generated using "fuzzy data" (i.e. noisy data such as biometric features) as a signing key, without using any additional user-specific data (such as a helper string in the context of fuzzy extractors). One of the main building blocks in the existing fuzzy signature schemes, is a primitive called *linear sketch*, which can be interpreted as a certain form of (one-way) encoding with which fuzzy data is encoded, and is used in combination with an ordinary signature scheme with certain functional and security properties, to construct a fuzzy signature scheme. Although the security of the underlying linear sketch scheme is very important for the security of the constructed fuzzy signature schemes, a linear sketch scheme is a relatively new primitive, and what security properties its definition and the existing constructions satisfy, has not been understood well. In order to deepen our understanding of this primitive, in this paper we clarify the security properties achieved by the existing linear sketch schemes. More specifically, we formalize security of a linear sketch scheme against "recovering" attacks, and then clarify that the existing linear sketch schemes achieve sufficiently strong security against them.

1 INTRODUCTION

1.1 Background and Motivation

Motivated mainly by application scenarios of biometric authentication, Takahashi et al. (Takahashi et al., 2015; Matsuda et al., 2016; Takahashi et al., 2017) recently introduced a special kind of digital signature called *fuzzy signature*. In a fuzzy signature scheme, signatures can be generated using "fuzzy data" (i.e. noisy data such as biometric features) as a signing key. There have been some approaches that achieve a similar feature, and perhaps the most well-known (and nowadays standard) approach would be to combine an ordinary signature scheme with a fuzzy extractor (Dodis et al., 2008). However, as pointed out by Takahashi et al., the fuzzy-extractor-based approach requires (public but) user-specific auxiliary data to be present at the time of signing. Hence, for example, a user has to carry it himself/herself, or the device executing the signing algorithm has to be on-line to retrieve the auxiliary data from some (public) repository. What distinguishes a fuzzy signature scheme from the fuzzy-extractor-based approach, is that the signing algorithm does *not* require user-specific data (other than his/her own biometric feature) at the time of signing, and hence for example a user can generate a signature even if he/she is completely empty-handed. This is quite attractive, and fuzzy signatures are expected to be applied to various applications.

We briefly review the results by Takahashi et al. (Takahashi et al., 2015; Matsuda et al., 2016; Takahashi et al., 2017).[1] In (Takahashi et al., 2017), in addition to formally define fuzzy signatures, they introduced 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 to be measured from the same object, 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., 2017) also introduced a tool that they call *linear sketch*, which is also a crypto-

[1]Since (Takahashi et al., 2017) is the merged full version of (Takahashi et al., 2015) and (Matsuda et al., 2016), from here on, by "Takahashi et al." we mean (Takahashi et al., 2017) unless indicated.

Matsuda, T., Takahashi, K. and Hanaoka, G.
On the Security of Linear Sketch Schemes against Recovering Attacks.
DOI: 10.5220/0006847100760087
In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications (ICETE 2018) - Volume 2: SECRYPT*, pages 76-87
ISBN: 978-989-758-319-3

graphic primitive associated with a fuzzy key setting, and is a kind of a pair of linear encoding and error correction methods that we will detail later. (Takahashi et al., 2017) then gave a generic construction of a fuzzy signature scheme for a fuzzy key setting from the combination of a linear sketch scheme (that is associated with the same fuzzy key setting) and an ordinary signature scheme that has some homomorphic properties regarding signing/verification keys. They then gave two instantiations of concrete fuzzy signature schemes via their generic construction, where for each construction, they specified a fuzzy key setting, constructed a linear sketch scheme, and then combined it with (a modified variant of) an existing ordinary signature scheme.

Linear Sketch. The main focus in this paper is on a linear sketch scheme. As mentioned earlier, this primitive can be understood as a pair of linear encoding and error correction methods. It is associated with a fuzzy key setting and an abelian group $\mathcal{K}$, and consists of two algorithms[2]: "Sketch" and "DiffRec" (where the second algorithm 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 as a "key" (or a "mask"). 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 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). In (Takahashi et al., 2017), it is required that a linear sketch scheme satisfies additional "linearity" and "weak simulatability" properties that are used in the security proof for the generic construction.

Our Motivation. Although the security of the underlying linear sketch scheme is very important for the security of the fuzzy signature schemes constructed from the generic construction of Takahashi et al., a linear sketch scheme is a relatively new primitive, and what security properties its definition and the existing constructions satisfy, has not been well understood. As mentioned above, in the formalization in (Takahashi et al., 2017), a linear sketch scheme is associated with a fuzzy key setting, which in turn specifies the underlying metric space and distribution of fuzzy data. So far, we only have two concrete constructions of linear sketch schemes: the first scheme (denoted by "$\mathcal{S}_{\mathtt{CRT}}$") is based on the Chinese remainder theorem, and the second one (denoted by "$\mathcal{S}_{\mathtt{Hash}}$") is based on a universal hash function family, and the fuzzy data space for these constructions is the space $[0,1)^n$ with the L_∞-distance. Since the constructions $\mathcal{S}_{\mathtt{CRT}}$ and $\mathcal{S}_{\mathtt{Hash}}$ seem tailored to this specific metric space, they have to inherently deal with non-integer numbers and furthermore they cannot be used with fuzzy key settings with other natural metrics for biometric authentication such as the edit distance and Hamming distance over bit strings.

In fact, the earlier papers (Takahashi et al., 2015; Matsuda et al., 2016) left the treatment of real numbers somewhat ambiguous, and Yasuda et al. (Yasuda et al., 2017) showed that the linear sketch schemes $\mathcal{S}_{\mathtt{CRT}}$ and $\mathcal{S}_{\mathtt{Hash}}$ could be vulnerable to so-called "recovering attacks" (which recover fuzzy data x and an element s from a sketch $c = \mathsf{Sketch}(s,x)$), if the real numbers in these schemes are treated in an inappropriate way. Concurrently to (Yasuda et al., 2017), the treatment of real numbers was unambiguously specified in (Takahashi et al., 2017), and with their treatment the attacks by (Yasuda et al., 2017) were shown to no longer work. However, this situation suggests that care must be taken in the definition of linear sketch schemes.

The main motivation of this paper is to contribute to deepening our understanding of this primitive, so that we can come up with better constructions and applications, which potentially could lead to future new constructions of fuzzy signatures (with fuzzy key settings that are different from the existing schemes).

1.2 Our Contributions

In order to deepen our understanding of a linear sketch scheme, in this paper we clarify a new aspect of the security properties achieved by the existing linear sketch schemes.

More specifically, in Section 4, we introduce security of a linear sketch scheme against "recovering" attacks, which directly captures the resistance against the attacks of (Yasuda et al., 2017). Namely, it requires that recovering fuzzy data x from a sketch c (and a public parameter pp) is hard. Our formalization uses the notion of average min-entropy (Dodis et al., 2008), which naturally corresponds to the hardness of guessing a secret given some leakage. Then, as our main technical results, we show that the two linear sketch schemes in (Takahashi et al., 2017), $\mathcal{S}_{\mathtt{CRT}}$ and $\mathcal{S}_{\mathtt{Hash}}$, satisfy sufficient level of security against recovering attacks (when the treatment of real numbers in (Takahashi et al., 2017) is taken into account), which are respectively shown in Sections 5 and 6.

[2] A linear sketch scheme actually also has the setup algorithm Setup that outputs a public parameter used by the other algorithms, but we omit them in the explanation in the introduction. The formal definition appears in Section 3.

These results directly imply that the attacks of (Yasuda et al., 2017) do not apply to the linear sketch schemes in (Takahashi et al., 2017).

We remark that if a linear sketch scheme is vulnerable to recovering attacks and is used in the generic construction of (Takahashi et al., 2017), then the resulting fuzzy signature scheme is also not secure. (In this sense, security against recovering attacks can be interpreted as a guideline for future designs of linear sketch schemes.) Since (Takahashi et al., 2017) gives security proofs for their fuzzy signature schemes, it implies that the linear sketch schemes in (Takahashi et al., 2017) are known to be secure against recovering attacks. Our results show this fact more directly.

To show the security of the linear sketch schemes against recovering attacks, we introduce the notion of *decomposability* and *partial hiding* property of a linear sketch scheme. Informally speaking, the decomposability property requires that the computation of $c = \mathsf{Sketch}(s,x)$ can be decomposed into two parts f_1 and f_2, where f_2 is independent of s, so that the computation $c = \mathsf{Sketch}(s,x)$ is equivalent to $c = (c_1,c_2) = (f_1(s,x), f_2(x))$. The *partial hiding* property requires that $c_1 = f_1(s,x)$ does not contain any information of x if s is chosen uniformly at random. We will show that if a linear sketch scheme satisfies these properties, then its security against recovering attacks can be reduced to the hardness of recovering fuzzy data x given only the leakage from the second component $f_2(x)$, which means that the security against recovering attacks can be analyzed by analyzing only the distribution of x and f_2. These properties make it easier to show the security of linear sketch schemes against recovering attacks, and we believe that their formalizations are useful for future designs of linear sketch schemes. We will show that the linear sketch schemes $\mathcal{S}_{\mathtt{CRT}}$ and $\mathcal{S}_{\mathtt{Hash}}$ satisfy these properties in Sections 5.4 and 6.3, respectively. For the formal definitions and the more details on how they are useful, see Section 4.

1.3 Paper Organization

The rest of this paper is organized as follows: In Section 2, we review some basic notation, basic definitions and facts, and the treatment of real numbers. In Section 3, we review the necessary definitions for linear sketch. In Section 4, we introduce security against recovering attacks for a linear sketch scheme. There, we also introduce decomposability and the partial hiding property of a linear sketch scheme, and show a useful lemma that is used in the subsequent sections. In Section 5, we show how strong the first linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$ is against recovering attacks. In Section 6, we do the same for the second linear sketch scheme $\mathcal{S}_{\mathtt{Hash}}$.

2 PRELIMINARIES

In this section, we review the basic notation, basic definitions and facts, and the treatment of real numbers in this paper.

2.1 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 $[n] := \{1,\dots,n\}$. If $a,b \in \mathbb{N}$, then "$\mathtt{GCD}(a,b)$" denotes the greatest common divisor of a and b. If $a \in \mathbb{R}$, then "$\lfloor a \rfloor$" denotes the maximum integer which does not exceed a (i.e. the rounding-down operation), and "$\lfloor a \rceil$" denotes the integer that is the nearest to a (i.e. the rounding operation). Throughout the paper, we use the bold font to denote a vector (such as $\mathbf{x}$ and $\mathbf{a}$). We extend the definition of "$\lfloor \cdot \rceil$" to allow it to take a real vector $\mathbf{a} = (a_1, a_2, \dots)$ as input, by $\lfloor \mathbf{a} \rceil := (\lfloor a_1 \rceil, \lfloor a_2 \rceil, \dots)$.

"$x \leftarrow y$" 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. 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. 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 $f(k) < 1/p(k)$.

2.2 Treatment of Real Numbers

Here, we recall the treatment of real numbers in (Takahashi et al., 2017). The following explanations are mostly taken verbatim from the "*On the Treatment of Real Numbers*" paragraph in (Takahashi et al., 2017, Section 6).

We assume that the significand of all real numbers is expressed in an a-priori fixed length (in bits) λ, where λ is some natural number that is a polynomial of a security parameter k. That is, a real number is expressed in the form $\frac{m}{2^\gamma}$, where m is a λ-bit integer that represents the significand and $-\gamma \in \mathbb{Z}$ is

the exponent.[3] Furthermore, if real numbers are involved in some arithmetic operations such as addition and multiplication, then the rounding-down operation is naturally applied to the significand of the resulting number, so that the result is always expressed in the above form (i.e. its significand is expressed with λ bits). We stress that this setting is natural, taking computer implementations into account.

For example, if we multiply a real number $x = \frac{m}{2^\gamma}$ (where m is a λ-bit integer and $0 \leq \gamma \leq \lambda$) with an n-bit integer a (where $n \leq \gamma$), then the resulting number $x \cdot a$ of the multiplication of x and a is treated as

$$\left\lfloor \frac{m \cdot a}{2^n} \right\rfloor \cdot 2^{-(\gamma - n)}. \quad (1)$$

That is, its significand is a λ-bit integer $\lfloor \frac{m \cdot a}{2^n} \rfloor$ and its exponent is $-(\gamma - n)$. This might not look straightforward at first glance, but note that the significand $\lfloor \frac{m \cdot a}{2^n} \rfloor$ is the result of the multiplication $m \cdot a$ rounded down to have a λ-bit precision (the denominator 2^n is due to the fact that a is an n-bit integer). The exponent is correspondingly "shifted" to take into account that a is an n-bit integer. See Figure 1 for an illustration for the calculation of $x \cdot a$.

2.3 Basic Definitions on Entropy

Here, we recall several basic definitions and a fact related to entropy used in this paper.

Definition 1. *Let $\mathcal{X}$ be a distribution defined over a set X. The* min-entropy *of $\mathcal{X}$, denoted by $\mathbf{H}_\infty(\mathcal{X})$, is defined by* $\mathbf{H}_\infty(\mathcal{X}) := -\log_2 \max_{x' \in X} \Pr[\mathcal{X} = x']$.

Definition 2 ((Dodis et al., 2008))**.** *Let $(\mathcal{X}, \mathcal{Y})$ be a joint distribution defined over the direct product of sets $X \times Y$. The* average min-entropy of $\mathcal{X}$ given $\mathcal{Y}$, *denoted by $\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y})$, is defined by*

$$\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y}) := -\log_2 \left(\mathop{\mathbf{E}}_{y \leftarrow_{\mathrm{R}} \mathcal{Y}} \left[\max_{x' \in X} \Pr[\mathcal{X} = x' | \mathcal{Y} = y] \right] \right).$$

Note that the average min-entropy $\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y})$ naturally captures the hardness of recovering a "secret" x from a "leakage" y when (x, y) is sampled according to $(\mathcal{X}, \mathcal{Y})$. More specifically, the definition of average min-entropy $\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y})$ implies that if $(x, y) \leftarrow_{\mathrm{R}} (\mathcal{X}, \mathcal{Y})$, then given y, even a computationally unbounded algorithm can succeed in finding x with probability at most $2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y})}$.

We will use the following simple fact formally shown in (Dodis et al., 2008).

Lemma 1. *Let $(\mathcal{X}, \mathcal{Y})$ be a joint distribution defined over the direct product of sets $X \times Y$. If $|Y| \leq 2^t$ for some $t \in \mathbb{N}$, then $\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Y}) \geq \mathbf{H}_\infty(\mathcal{X}) - t$.*

[3] For ease of treatment of decimal numbers, we use the convention that a positive γ implies a negative exponent.

2.4 Universal Hash Function Family

Here, we first recall the definition of a universal hash function family, then the existence of a universal hash function family with linearity.

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* $\Pr_{z \leftarrow_{\mathrm{R}} Z}[h_z(x) = h_z(x')] \leq 1/|R|$.

In the linear sketch scheme $\mathcal{S}_{\mathtt{Hash}}$ in (Takahashi et al., 2017), a universal hash function family with a linearity property is used. (See e.g. (Cheraghchi, 2011) for a concrete construction.)

Lemma 2. *Let $\mathbb{F}_p$ be a finite field with prime order p, and let $n \in \mathbb{N}$. There exists a family of universal hash function $\mathcal{H}_{lin} = \{h_z : (\mathbb{F}_p)^n \to \mathbb{F}_p\}_{z \in \mathbb{F}_{p^n}}$ such that for all $z \in \mathbb{F}_{p^n}$, the following linearity property is satisfied: For all $\mathbf{x}, \mathbf{x}' \in (\mathbb{F}_p)^n$ and $\alpha, \beta \in \mathbb{F}_p$, it holds that*

$$\alpha \cdot h_z(\mathbf{x}) + \beta \cdot h_z(\mathbf{x}') = h_z(\alpha \cdot \mathbf{x} + \beta \cdot \mathbf{x}').$$

3 DEFINITIONS FOR LINEAR SKETCH

Here, we review the definitions for a fuzzy key setting in Section 3.1 and a linear sketch scheme in Section 3.2.

In order to grasp how linear sketch schemes are used in the constructions of a fuzzy signature scheme, in Appendix, we review the definition of a fuzzy signature scheme as well as the two fuzzy signature schemes proposed in (Takahashi et al., 2017).

3.1 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 formalization of (Takahashi et al., 2017) 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 $\{e \leftarrow_{\mathrm{R}} \Phi;\ x' \leftarrow x + e : x'\}$.

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

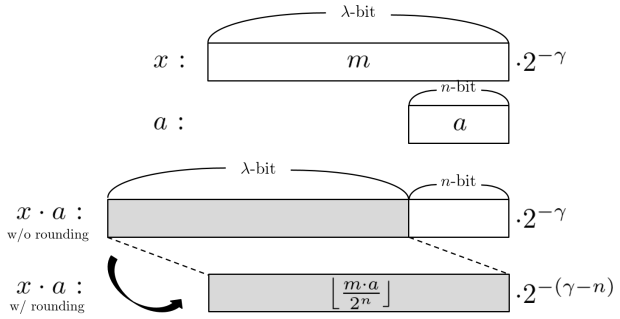

Figure 1: An illustration of multiplication of a real number $x = \frac{m}{2^\gamma}$ and an n-bit integer a. (This picture is taken from (Takahashi et al., 2017, Fig. 5).)

(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 $\mathtt{FAR} := \Pr[x, x' \leftarrow_{\mathtt{R}} \mathcal{X} : \mathsf{d}(x, x') < t]$ 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_e**:** ($\in [0,1]$) This is an error parameter that represents FRR. We require that for all $x \in X$, $\mathtt{FRR} := \Pr[e \leftarrow_{\mathtt{R}} \Phi : \mathsf{d}(x, x+e) \geq t] \leq \varepsilon_e$.

We remark that the false acceptance/rejection rate FAR and FRR, the error distribution Φ, and the error parameter ε_e are not directly used in our paper because they do not directly affect the security of linear sketch schemes.

3.2 Linear Sketch

Here, we review the definition of a linear sketch scheme in (Takahashi et al., 2017).

Let $\mathcal{F} = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \varepsilon_e)$ be a fuzzy key setting, and let $\mathcal{K}$ be a finite abelian group. A linear sketch scheme $\mathcal{S}$ for $(\mathcal{F}, \mathcal{K})$, consists of the following three algorithms $(\mathsf{Setup}, \mathsf{Sketch}, \mathsf{DiffRec})$:

Setup is the "setup" algorithm that takes 1^k, and outputs a public parameter pp.

Sketch is the "sketching" algorithm that takes pp, an element $s \in \mathcal{K}$, and fuzzy data $x \in X$ as input, and outputs a "sketch" c.

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" $\Delta s \in \mathcal{K}$.

For correctness, it is required that for all $k \in \mathbb{N}$, all $x, x' \in X$ such that $\mathsf{d}(x, x') < t$, all pp output by $\mathsf{Setup}(1^k)$, and all $s, \Delta s \in \mathcal{K}$, we have $\mathsf{DiffRec}(pp, \mathsf{Sketch}(pp, s, x), \mathsf{Sketch}(pp, s + \Delta s, x')) = \Delta s$.

Besides, the properties called *linearity* and *weak simulatability* are formalized in (Takahashi et al., 2017). Informally, the linearity ensures that given pp, a sketch value c output from $\mathsf{Sketch}(pp, s, x)$, and "shift" values $\Delta s \in \mathcal{K}$ and $\Delta x \in X$, it is possible to compute a sketch value c' that is distributed as if it is computed from $\mathsf{Sketch}(pp, s + \Delta s, x + \Delta x)$. The weak simulatability captures a weak form of confidentiality that c does not leak the information of the content s if s is chosen uniformly at random from $\mathcal{K}$ and x is sampled from $\mathcal{X}$. Since we do not directly use these properties, we omit the formal definitions. See (Takahashi et al., 2017) for the formal definitions.

4 DEFINING SECURITY AGAINST RECOVERING ATTACKS

In this section, we introduce security of a linear sketch scheme against recovering attacks. We then introduce structural properties of a linear sketch scheme that we call the *decomposability* and *partial hiding* property. Finally, we show a lemma that is useful for analyzing the security of linear sketch schemes that satisfy these properties. This lemma plays the main role in the subsequent sections.

In the following definitions, let $\mathcal{F} = ((d, X), t, \mathcal{X}, \Phi, \varepsilon_e)$ be a fuzzy key setting, let $\mathcal{K}$ be an abelian group, and let $\mathcal{S} = (\mathsf{Setup}, \mathsf{Sketch}, \mathsf{DiffRec})$ be a linear sketch for $(\mathcal{F}, \mathcal{K})$.

Definition 4. *Let $\varepsilon_r = \varepsilon_r(k) \in [0,1]$. We say that a linear sketch scheme $\mathcal{S}$ for $(\mathcal{F}, \mathcal{K})$ is* ε_r-secure against recovering attacks *if $\mathsf{Adv}^{\mathtt{recover}}_{\mathcal{S}}(k) := 2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{P},\mathcal{C})} \leq \varepsilon_r$ holds, where $(\mathcal{P}, \mathcal{C})$ is the joint distribution defined*

as follows:

$$(\mathcal{P},\mathcal{C}) := \{\ pp \leftarrow_{\mathtt{R}} \mathsf{Setup}(1^k);\ x \leftarrow_{\mathtt{R}} \mathcal{X};\ s \leftarrow_{\mathtt{R}} \mathcal{K};\ c \leftarrow_{\mathtt{R}} \mathsf{Sketch}(pp,s,x) : (pp,c)\ \}. \quad (2)$$

(Note that $(\mathcal{X},\mathcal{P},\mathcal{C})$ forms a joint distribution.)

As mentioned earlier, the average min-entropy $\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{P},\mathcal{C})$ (in the form of $2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{P},\mathcal{C})}$) gives an upper bound of the best (computationally unbounded) adversary's advantage in guessing x given (pp,c), when (x,pp,c) is sampled according to the joint distribution $(\mathcal{X},\mathcal{P},\mathcal{C})$. Hence, if ε_r is (negligibly) small, ε_r-security against recovering attacks guarantees that recovering x from (pp,c) is hard.

Next, we introduce a structural property for a linear sketch scheme that we call *decomposability*, which means that an execution of $\mathsf{Sketch}(pp,s,x)$ can be decomposed into computing two sub-functions f_1 and f_2, where f_1 is a (possibly probabilistic) function whose output depends on (pp,s,x), while f_2 is a deterministic function whose output depends only on (pp,x) (and independent of s), so that a sketch c consists of the outputs from f_1 and f_2, i.e., $c = (f_1(pp,s,x), f_2(pp,x))$. We also require the property that we call *partial hiding*, which requires that the first function f_1 hides the information of x from its output c_1 if the input s is chosen randomly. (We do not require the same for f_2, and hence the name "partial".) The usefulness of the decomposability and partial-hiding property will be evident later.

Definition 5. *We say that a linear sketch scheme $\mathcal{S}$ for $(\mathcal{F},\mathcal{K})$ is* decomposable*, if there exist the following two efficiently computable functions f_1, f_2:*

- *f_1 is a (possibly probabilistic) function that takes pp (output by Setup), $s \in \mathcal{K}$, and $x \in X$ as input, and outputs some value c_1.*
- *f_2 is a deterministic function that takes pp (output by Setup) and $x \in X$ as input, and outputs some value c_2.*

It is required that for all pp output by Setup, $x \in \mathcal{X}$, and $s \in \mathcal{K}$, the following two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ are identically distributed:

$$\mathcal{D}_1 := \{c = (c_1,c_2) \leftarrow_{\mathtt{R}} \mathsf{Sketch}(pp,s,x) : c\ \},$$
$$\mathcal{D}_2 := \{c_1 \leftarrow_{\mathtt{R}} f_1(pp,s,x);\ c_2 \leftarrow f_2(pp,x) : (c_1,c_2)\}$$

f_1 and f_2 are called the decomposed functions *of* Sketch.

Furthermore, we say that a decomposable linear sketch scheme $\mathcal{S}$ (with the decomposed functions f_1 and f_2) is partially hiding *if for all pp output by* Setup*, the distribution $\{x \leftarrow_{\mathtt{R}} \mathcal{X};\ s \leftarrow_{\mathtt{R}} \mathcal{K};\ c_1 \leftarrow_{\mathtt{R}} f_1(pp,x,s) : c_1\}$ is independent of the original fuzzy data distribution $\mathcal{X}$.*

We now show that any decomposable linear sketch scheme $\mathcal{S} = (\mathsf{Setup},\mathsf{Sketch},\mathsf{DiffRec})$ with partial hiding property, satisfies a convenient property that its security against recovering attacks can be analyzed only from $\mathcal{X}$ and f_2 (irrelevantly to f_1, s, or c).

Lemma 3. *Let $\mathcal{S} = (\mathsf{Setup},\mathsf{Sketch},\mathsf{DiffRec})$ be a linear sketch scheme $\mathcal{S}$ for $(\mathcal{F},\mathcal{K})$. Assume that $\mathcal{S}$ satisfies decomposability (with decomposed functions f_1 and f_2) and the partial hiding property. Then, it holds that $\mathsf{Adv}^{\mathtt{recover}}_{\mathcal{S}}(k) = 2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{P},f_2(\mathcal{P},\mathcal{X}))}$.*

Proof of Lemma 3. Consider the following distribution that outputs (pp,c_1,c_2):

$$\{\ pp \leftarrow_{\mathtt{R}} \mathsf{Setup}(1^k);\ x \leftarrow_{\mathtt{R}} \mathcal{X};\ s \leftarrow_{\mathtt{R}} \mathcal{K};\ c_1 \leftarrow_{\mathtt{R}} f_1(pp,s,x);\ c_2 \leftarrow f_2(pp,x) : (pp,c_1,c_2)\ \}.$$

Let $\mathcal{P}$, $\mathcal{C}_1$, and $\mathcal{C}_2$ be the distributions that correspond to computing and outputting pp, c_1 and c_2 in the above distribution, respectively. Note that $(\mathcal{P},\mathcal{C}_1,\mathcal{C}_2)$ forms a joint distribution, and furthermore we have $\mathcal{C}_2 = f_2(\mathcal{P},\mathcal{X})$ by definition. Due to the decomposability of $\mathcal{S}$, the joint distribution $(\mathcal{P},\mathcal{C})$ (defined in Eq. (2)) and the joint distribution $(\mathcal{P},\mathcal{C}_1,\mathcal{C}_2)$ are equivalent, and thus it holds that

$$\begin{aligned}\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{P},\mathcal{C}) &= \widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{P},\mathcal{C}_1,\mathcal{C}_2)\\ &= \widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{P},\mathcal{C}_1,f_2(\mathcal{P},\mathcal{X})).\end{aligned}$$

Furthermore, by the partial hiding property and the fact that $\mathcal{X}$ is independent of the generation of a public parameter, $\mathcal{X}$ and $\mathcal{C}_1$ are guaranteed to be independent, which means that we have $\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{P},\mathcal{C}_1,f_2(\mathcal{P},\mathcal{X})) = \widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{P},f_2(\mathcal{P},\mathcal{X}))$.

Combining the above arguments, we obtain $\mathsf{Adv}^{\mathtt{recover}}_{\mathcal{S}}(k) = 2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{P},\mathcal{C})} = 2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{P},f_2(\mathcal{P},\mathcal{X}))}$. This completes the proof of Lemma 3. □

5 RECOVERING ATTACK SECURITY OF LINEAR SKETCH SCHEME $\mathcal{S}_{\mathtt{CRT}}$

In this and next sections, we show our main results: security of the linear sketch schemes in (Takahashi et al., 2017) against recovering attacks. In this section, we do this for the first linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$.

This section is organized as follows: In Section 5.1, we recall a concrete fuzzy key setting $\mathcal{F}_1$ with which the linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$ is associated. Then, in Section 5.2, we review some mathematical background for describing $\mathcal{S}_{\mathtt{CRT}}$. In Section 5.3, we give the description of $\mathcal{S}_{\mathtt{CRT}}$. Finally, in Section 5.4, we show how secure the linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$ is against recovering attacks.

A large part of Sections 5.1, 5.2, and 5.3 is taken verbatim from Sections 6.1, 6.2, and 6.3 of (Takahashi et al., 2017), respectively.

5.1 Specific Fuzzy Key Setting $\mathcal{F}_1$

Here, we recall a concrete fuzzy key setting $\mathcal{F}_1 = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \varepsilon_e)$ for which the linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$ 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_∞-distance 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

$$k = \lfloor -n \log_2(2t) \rfloor. \tag{3}$$

Furthermore, we require that $n = O(\log_2 k)$, so that 2^n can be considered to be upper-bounded by some polynomial of k. This property was used in (Takahashi et al., 2017) to show the weak simulatability of the linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$.

Distribution $\mathcal{X}$: The uniform distribution over a "discretized" version of $X = [0,1)^n$. Specifically, let $\lambda \in \mathbb{N}$ be the natural number that denotes the representation length of a real number (see Section 2.2). We require that each coordinate x_i of a data $\mathbf{x} = (x_1, \ldots, x_n) \in X$ be distributed as $\{j_i \leftarrow_{\mathtt{R}} \mathbb{Z}_{2^\lambda} : \frac{j_i}{2^\lambda}\}$. Furthermore, we require λ to be sufficiently large (at least $\lceil k/n \rceil$).

Error distribution Φ and Error parameter ε_e: Φ is any efficiently samplable distribution over X such that $\mathtt{FRR} \leq \varepsilon_e$ for all $x \in X$.

5.2 Mathematical Preliminaries

We recall some mathematical background that is necessary for describing the first linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$ in (Takahashi et al., 2017).

Let $n \in \mathbb{N}$, and $w_1, \ldots, w_n \in \mathbb{N}$ 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

$$\forall i \in [n] : w_i \leq \frac{1}{2t}, \text{ and } \forall i \neq j \in [n] : \mathtt{GCD}(w_i, w_j) = 1, \tag{4}$$

and $W = \prod_{i \in [n]} w_i = \Theta(2^k)$, where k is defined as in Eq. (3). Note that Eqs. (3) and (4) imply that we have $w_i \leq 2^{\lceil k/n \rceil}$ for all $i \in [n]$. 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 $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{N}^n$, we define

$$\mathbf{v} \bmod \mathbf{w} := (v_1 \bmod w_1, \ldots, v_n \bmod w_n). \tag{5}$$

For vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{N}^n$, we define the equivalence relation "$\sim$" by

$$\mathbf{v}_1 \sim \mathbf{v}_2 \quad \overset{\text{def}}{\iff} \quad \mathbf{v}_1 \bmod \mathbf{w} = \mathbf{v}_2 \bmod \mathbf{w},$$

and let $\mathbb{Z}^n_{\mathbf{w}} := \mathbb{Z}^n / \sim$ be the quotient set of $\mathbb{Z}^n$ by $\sim$. Note that $(\mathbb{Z}^n_{\mathbf{w}}, +)$ constitutes an abelian group, where the addition is modulo $\mathbf{w}$ as defined in Eq. (5).

According to the Chinese remainder theorem (CRT), each element in $\mathbb{Z}_W$ can be uniquely represented as an element in $\mathbb{Z}^n_{\mathbf{w}}$, and vice versa. Let $\mathsf{CRT}_{\mathbf{w}} : \mathbb{Z}^n_{\mathbf{w}} \to \mathbb{Z}_W$ be the mapping that transforms an element in $\mathbf{s} \in \mathbb{Z}^n_{\mathbf{w}}$ into $s \in \mathbb{Z}_W$ via the CRT, and we denote by $\mathsf{CRT}^{-1}_{\mathbf{w}}$ the inverse function of $\mathsf{CRT}_{\mathbf{w}}$. Note that for all $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{Z}^n_{\mathbf{w}}$, it holds that

$$\mathsf{CRT}_{\mathbf{w}}(\mathbf{v}_1 + \mathbf{v}_2) = \mathsf{CRT}_{\mathbf{w}}(\mathbf{v}_1) + \mathsf{CRT}_{\mathbf{w}}(\mathbf{v}_2) \bmod W.$$

Similarly to $\mathbb{Z}^n_{\mathbf{w}}$, we define $\mathbb{R}^n_{\mathbf{w}} := \mathbb{R}^n / \sim$ to 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 $0 \leq r < w_i$.

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

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

where $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$. Note that it holds that

$$\mathsf{E}_{\mathbf{w}}(\mathbf{x} + \mathbf{e}) = \mathsf{E}_{\mathbf{w}}(\mathbf{x}) + \mathsf{E}_{\mathbf{w}}(\mathbf{e}) \pmod{\mathbf{w}}. \tag{6}$$

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

$$\mathsf{C}_{\mathbf{w}}\Big((y_1, \ldots, y_n) \Big) := \Big(\lfloor y_1 + 0.5 \rfloor, \ldots, \lfloor y_n + 0.5 \rfloor \Big). \tag{7}$$

We note that the rounding-down 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. (4), the following properties are satisfied: For any $\mathbf{x}, \mathbf{x}' \in X$, if $\mathsf{d}(\mathbf{x}, \mathbf{x}') = |\mathbf{x} - \mathbf{x}'\|_\infty < t$, then we have

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

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

$$\mathsf{C}_{\mathbf{w}}\Big(\mathsf{E}_{\mathbf{w}}(\mathbf{x}) - \mathsf{E}_{\mathbf{w}}(\mathbf{x}') \Big) = \mathbf{0}. \tag{8}$$

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

$$\mathsf{C}_{\mathbf{w}}(\mathbf{x} + \mathbf{s}) = \mathsf{C}_{\mathbf{w}}(\mathbf{x}) + \mathbf{s} \pmod{\mathbf{w}}. \tag{9}$$

5.3 Scheme Description

Let $\mathcal{F}_1 = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \varepsilon_e)$ be the fuzzy key setting defined in Section 5.1, and let $\mathbf{w} = (w_1, \ldots, w_n) = \mathsf{WGen}(t, n)$, where n is the dimension of X, and let $W = \prod_{i \in [n]} w_i$. Let $\mathsf{CRT}_{\mathbf{w}}$, $\mathsf{CRT}_{\mathbf{w}}^{-1}$, $\mathsf{E}_{\mathbf{w}}$, and $\mathsf{C}_{\mathbf{w}}$ be the functions defined in the previous subsection. Using these objects, the linear sketch scheme $\mathcal{S}_{\mathtt{CRT}} = (\mathsf{Setup}, \mathsf{Sketch}, \mathsf{DiffRec})$ for $\mathcal{F}_1$ and the additive group $(\mathbb{Z}_W, +)$, is constructed as in Figure 2.

Note that the setup algorithm Setup in this linear sketch scheme actually does nothing, and the main algorithms Sketch and $\mathsf{DiffRec}$ are deterministic. Furthermore, recall that the rounding-down operation is applied after multiplication of a real number and an integer is performed, so that the resulting real number has a λ-bit significand. Concretely, let $s \in \mathbb{Z}_W$ and $\mathbf{s} = (s_1, \ldots, s_n) = \mathsf{CRT}_{\mathbf{w}}^{-1}$, and suppose that $\mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{R}_{\mathbf{w}}^n$ is a sketch output from $\mathsf{Sketch}(pp, s, \mathbf{x})$, where $\mathbf{x} = (x_1, \ldots, x_n) \leftarrow_{\mathtt{R}} \mathcal{X}$ and thus each x_i is of the form $x_i = \frac{j_i}{2^\lambda}$ for some λ-bit integer j_i. Then, since each w_i is a $\lceil k/n \rceil$-bit integer, $x_i' = w_i x_i$ results in $\lfloor \frac{w_i \cdot j_i}{2^{\lceil k/n \rceil}} \rfloor \cdot 2^{-(\lambda - \lceil k/n \rceil)}$. Consequently, each $c_i \in \mathbb{R}_{w_i}$ is of the following form:

$$c_i = s_i + \left\lfloor \frac{w_i \cdot j_i}{2^{\lceil k/n \rceil}} \right\rfloor \cdot 2^{-(\lambda - \lceil k/n \rceil)} \mod w_i. \tag{10}$$

It was shown in (Takahashi et al., 2017) that $\mathcal{S}_{\mathtt{CRT}}$ satisfies correctness, and also satisfies the linearity and weak simulatability properties.

5.4 Security against Recovering Attacks

In this subsection, we show how secure the linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$ is against recovering attacks. To this end, we first show that $\mathcal{S}_{\mathtt{CRT}}$ satisfies decomposability with the partial hiding property.

Lemma 4. *The linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$ satisfies the decomposability and partial hiding property.*

Proof of Lemma 4. Let $s \in \mathbb{Z}_W$ and $\mathbf{s} = (s_1, \ldots, s_n) := \mathsf{CRT}_{\mathbf{w}}^{-1}(s) \in \mathbb{Z}_{\mathbf{w}}$. Let $\mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{R}_{\mathbf{w}}^n$ be a sketch output from $\mathsf{Sketch}(pp, s, \mathbf{x})$, where $\mathbf{x} = (x_1, \ldots, x_n) \leftarrow_{\mathtt{R}} \mathcal{X}$ and thus each x_i is of the form $x_i = \frac{j_i}{2^\lambda}$ for some λ-bit integer j_i. As mentioned in the previous subsection, due to the rounding-down operation performed at each multiplication $w_i \cdot x_i$, each coordinate c_i of the sketch $\mathbf{c}$ is computed as in Eq. (10). Here, we consider the "integer" part $c_{in}^{(i)}$ and the "decimal" part $c_{de}^{(i)}$ of c_i, which are as follows:

$$c_{in}^{(i)} = \lfloor c_i \rfloor = s_i + \left\lfloor \left\lfloor \frac{w_i \cdot j_i}{2^{\lceil k/n \rceil}} \right\rfloor \cdot 2^{-(\lambda - \lceil k/n \rceil)} \right\rfloor \mod w_i, \tag{11}$$

$$c_{de}^{(i)} = c_i \bmod 1 = \left\lfloor \frac{w_i \cdot j_i}{2^{\lceil k/n \rceil}} \right\rfloor \cdot 2^{-(\lambda - \lceil k/n \rceil)} \bmod 1. \tag{12}$$

Note that $c_{in}^{(i)} \in \mathbb{Z}_{w_i}$ and $c_{de}^{(i)} \in [0, 1)$. Note also that the decimal part $c_{de}^{(i)}$ is independent of s_i due to "mod 1". We can make use of this fact, and define the decomposed functions $f_1(pp, s, \mathbf{x})$ and $f_2(pp, \mathbf{x})$ as follows:

$f_1(pp, s, \mathbf{x})$: This function outputs $\mathbf{c}_1 = (c_{in}^{(1)}, \ldots, c_{in}^{(n)}) \in \mathbb{Z}_{\mathbf{w}}^n$, where each $c_{in}^{(i)}$ is computed as in Eq. (11).

$f_2(pp, \mathbf{x})$: This function outputs $\mathbf{c}_2 = (c_{de}^{(1)}, \ldots, c_{de}^{(n)}) \in [0, 1)^n$, where each $c_{de}^{(i)}$ is computed as in Eq. (12).

Furthermore, it is straightforward to see that f_1 satisfies the partial hiding property. Specifically, if $s \in \mathbb{Z}_W$ is chosen uniformly at random, then $\mathbf{s} = (s_1, \ldots, s_n) = \mathsf{CRT}^{-1}(s)$ is distributed uniformly over $\mathbb{Z}_{\mathbf{w}}^n$. Hence, each s_i acts as a key for one-time pad encryption, and thus each $c_{in}^{(i)} \in \mathbb{Z}_{w_i}$ does not contain any information of x_i. Therefore, when $s \in \mathbb{Z}_W$ is chosen uniformly at random, the output $\mathbf{c}_1$ of $f_1(pp, s, \mathbf{x})$ becomes independent of $\mathbf{x}$. This completes the proof of Lemma 4. □

We now show how secure the linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$ is against recovering attacks.

Theorem 1. *The linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$ satisfies 2^{-k}-security against recovering attacks.*

Proof of Theorem 1. Due to Lemmas 3 and 4, and the fact that $\mathcal{P}$ does not involve any probabilistic operation, we have

$$\mathsf{Adv}_{\mathcal{S}_{\mathtt{CRT}}}^{\mathtt{recover}}(k) = 2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X} | \mathcal{P}, f_2(\mathcal{P}, \mathcal{X}))} = 2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X} | f_2(\mathcal{P}, \mathcal{X}))}.$$

Moreover, due to our treatment of real numbers, each coordinate $c_{de}^{(i)}$ of the output of f_2 can have at most $2^{\lambda - \lceil k/n \rceil}$ values. Hence, the entire output of f_2 can take at most $(2^{\lambda - \lceil k/n \rceil})^n \leq 2^{n \cdot \lambda - k}$ values. Since $\mathbf{H}_\infty(\mathcal{X}) = n \cdot \lambda$ holds due to the assumption on the fuzzy data distribution, by applying Lemma 1, we can estimate $\widetilde{\mathbf{H}}_\infty(\mathcal{X} | f_2(\mathcal{P}, \mathcal{X}))$ as follows:

$$\begin{aligned} \widetilde{\mathbf{H}}_\infty(\mathcal{X} | f_2(\mathcal{P}, \mathcal{X})) &\geq \mathbf{H}_\infty(\mathcal{X}) - (n \cdot \lambda - k) \\ &= n \cdot \lambda - (n \cdot \lambda - k) = k. \end{aligned}$$

Combining the two inequalities, we obtain $\mathsf{Adv}_{\mathcal{S}_{\mathtt{CRT}}}^{\mathtt{recover}}(k) \leq 2^{-k}$, which means that $\mathcal{S}_{\mathtt{CRT}}$ satisfies 2^{-k}-security against recovering attacks. This completes the proof of Theorem 1. □

$\mathsf{Setup}(1^k)$:	$\mathsf{Sketch}(pp, s \in \mathbb{Z}_W, \mathbf{x} \in [0,1)^n)$:	$\mathsf{DiffRec}(pp, \mathbf{c}, \mathbf{c}')$:
Return $pp \leftarrow (w_1, \ldots, w_n, W)$.	$\mathbf{c} \leftarrow (\mathsf{CRT}_{\mathbf{w}}^{-1}(s) + \mathsf{E}_{\mathbf{w}}(\mathbf{x})) \bmod \mathbf{w}$ Return $\mathbf{c}$.	$\Delta\mathbf{s} \leftarrow \mathsf{C}_{\mathbf{w}}(\mathbf{c}' - \mathbf{c})$ $\Delta s \leftarrow \mathsf{CRT}_{\mathbf{w}}(\Delta\mathbf{s})$ Return Δs.

Figure 2: The linear sketch scheme $\mathcal{S}_{\mathtt{CRT}} = (\mathsf{Setup}, \mathsf{Sketch}, \mathsf{DiffRec})$ for the fuzzy key setting $\mathcal{F}_1$. In the figure, additions/subtractions are done in $\mathbb{R}^n_{\mathbf{w}}$.

6 RECOVERING ATTACK SECURITY OF THE LINEAR SKETCH SCHEME $\mathcal{S}_{\mathtt{Hash}}$

In this section, we show the security of the second linear sketch scheme $\mathcal{S}_{\mathtt{Hash}}$ in (Takahashi et al., 2017) against recovering attacks.

This section is organized as follows: In Section 6.1, we recall a concrete fuzzy key setting $\mathcal{F}_2$ with which the linear sketch scheme $\mathcal{S}_{\mathtt{Hash}}$ is associated. Then in Section 6.2, we give the description of $\mathcal{S}_{\mathtt{Hash}}$. Finally, in Section 6.3, we show how secure the linear sketch scheme $\mathcal{S}_{\mathtt{Hash}}$ is against recovering attacks.

A large part of Sections 6.1 and 6.2 is taken verbatim from Sections 7.1 and 7.2 of (Takahashi et al., 2017), respectively.

6.1 Specific Fuzzy Key Setting $\mathcal{F}_2$

Here, we recall a concrete fuzzy key setting $\mathcal{F}_2 = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \varepsilon_e)$ for which the linear sketch scheme $\mathcal{S}_{\mathtt{Hash}}$ is 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_∞-distance. 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 require the threshold $t \in \mathbb{R}$ to satisfy

$$k \leq \lfloor -n \log_2(2t) \rfloor. \tag{13}$$

For notational convenience, let $T := 1/(2t)$.

Distribution $\mathcal{X}$: An efficiently samplable distribution over a "discretized" version of $X = [0,1)^n$. That is, letting $\lambda \in \mathbb{N}$ denote the length of the significand of a real number, if $\mathbf{x} = (x_1, \ldots, x_n)$ is sampled from $\mathcal{X}$, then each x_i is of the form $\frac{j_i}{2^\lambda}$, where j_i is some λ-bit integer. (See Section 2.2.) We require $T \leq 2^\lambda$.

Furthermore, we require that $\mathcal{X}$ satisfy the assumption on the average min-entropy that we state later.

Error distribution Φ and Error parameter ε_e: Φ is any efficiently samplable distribution over X such that $\mathtt{FRR} \leq \varepsilon_e$ for all $x \in X$.

The Requirement on the Distribution of Fuzzy Data $\mathcal{X}$. Let $\mathcal{X}'$ be the "scaled-up" version of $\mathcal{X}$, namely, $\mathcal{X}'$ is the distribution obtained by multiplying the value $T = 1/(2t)$ to the outcome of the distribution $\mathcal{X}$, where the rounding-down operation is performed for each coordinate of $\mathcal{X}'$ as explained in Section 2.2. Since $\mathcal{X}$ is a distribution over $[0,1)^n$, $\mathcal{X}'$ is a distribution over $[0,T)^n$. Now, let us divide $\mathcal{X}'$ into the "integer" part $\mathcal{X}'_{in}$ and the "decimal" part $\mathcal{X}'_{de}$. Namely, let $\mathbf{x}' = (x'_1, \ldots, x'_n)$ 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 each coordinate of the integer part $\mathcal{X}'_{in}$ is represented by $\lceil \log_2 T \rceil$ bits, and thus each coordinate of the decimal part $\mathcal{X}'_{de}$ will have $(\lambda - \lceil \log_2 T \rceil)$-bit precision, so that the significand of the entire x'_i is expressed in λ bits. Note also that the joint distribution $(\mathcal{X}'_{in}, \mathcal{X}'_{de})$ contains the same information as $\mathcal{X}'$.

The requirement on the distribution $\mathcal{X}$ is that we have

$$\widetilde{\mathbf{H}}_\infty(\mathcal{X}'_{in} | \mathcal{X}'_{de}) \geq \log_2 p + \omega(\log_2 k), \tag{14}$$

where p is the order of the field over which the universal hash family $\mathcal{H}_{lin}$ (guaranteed by Lemma 2) is constructed.

6.2 Scheme Description

Let $\mathcal{F}_2 = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \varepsilon_e)$ be the fuzzy key setting as defined above. Let $\mathbb{F}_p$ be a finite field with prime order p satisfying $p \geq T = 1/(2t)$. Here, we identify $\mathbb{F}_p$ with $\mathbb{Z}_p$, and 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 as guaranteed by Lemma 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, $h_z^{-1}(s) := \{\mathbf{a} \in (\mathbb{F}_p)^n | h_z(\mathbf{a}) = s\}$. Hence, the notation "$\mathbf{a} \leftarrow_{\mathtt{R}} h_z^{-1}(s)$" means that we choose a vector $\mathbf{a}$ uniformly from the set $h_z^{-1}(s)$.

Then, using these ingredients, the linear sketch scheme $\mathcal{S}_{\mathtt{Hash}} = (\mathsf{Setup}, \mathsf{Sketch}, \mathsf{DiffRec})$ for $\mathcal{F}_2$ and the additive group $(\mathbb{Z}_p, +)$, is constructed as described in Figure 3.

We remind the reader that we are treating real numbers as explained in Section 2.2. We remark that as in the first linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$, the rounding-down operation is applied after the multiplication $T \cdot \mathbf{x}$, so that the resulting real number in each coordinate has a λ-bit significand. Concretely, let $s \in \mathbb{Z}_p$, and let $\mathbf{a} = (a_1, \ldots, a_n) \in h_z^{-1}(s)$, and suppose that $\mathbf{c} = (c_1, \ldots, c_n) \in (\mathbb{R}_p)^n$ is output from $\mathsf{Sketch}(pp, s, \mathbf{x})$, where $\mathbf{x} = (x_1, \ldots, x_n) \leftarrow_{\mathtt{R}} \mathcal{X}$ and thus each x_i is of the form $x_i = \frac{j_i}{2^\lambda}$ for some λ-bit integer j_i. Then, for each $i \in [n]$, $x_i' = T \cdot x_i$ results in $\lfloor \frac{T \cdot j_i}{2^{\lceil \log_2 T \rceil}} \rfloor \cdot 2^{-(\lambda - \lceil \log_2 T \rceil)}$. Consequently, each $c_i \in \mathbb{R}_p$ is of the following form:

$$c_i = a_i + \left\lfloor \frac{T \cdot j_i}{2^{\lceil \log_2 T \rceil}} \right\rfloor \cdot 2^{-(\lambda - \lceil \log_2 T \rceil)} \mod p. \quad (15)$$

It was shown in (Takahashi et al., 2017) that $\mathcal{S}_{\mathtt{Hash}}$ satisfies correctness, and also satisfies the linearity and weak simulatability properties.

6.3 Security against Recovering Attacks

In this subsection, we show how secure the linear sketch scheme $\mathcal{S}_{\mathtt{Hash}}$ is against recovering attacks. To this end, we first show that $\mathcal{S}_{\mathtt{Hash}}$ satisfies decomposability with the partial hiding property.

Lemma 5. *The linear sketch scheme $\mathcal{S}_{\mathtt{Hash}}$ satisfies the decomposability and partial hiding property.*

Proof of Lemma 5. Let $s \in \mathbb{Z}_p$ and $\mathbf{a} = (a_1, \ldots, a_n) \in h_z^{-1}(s)$. Let $\mathbf{c} = (c_1, \ldots, c_n) \in (\mathbb{R}_p)^n$ be a sketch output from $\mathsf{Sketch}(pp, s, \mathbf{x})$, where $\mathbf{x} = (x_1, \ldots, x_n) \leftarrow_{\mathtt{R}} \mathcal{X}$ and thus each x_i is of the form $x_i = \frac{j_i}{2^\lambda}$ for some λ-bit integer j_i. As mentioned in the previous subsection, due to the rounding-down operation performed at each multiplication $T \cdot x_i$, each coordinate c_i of the sketch $\mathbf{c}$ is computed as in Eq. (15). Here, we consider the "integer" part $c_{in}^{(i)}$ and the "decimal" part $c_{de}^{(i)}$ of c_i, which are as follows:

$$c_{in}^{(i)} = \lfloor c_i \rfloor = a_i + \left\lfloor \left\lfloor \frac{T \cdot j_i}{2^{\lceil \log_2 T \rceil}} \right\rfloor \cdot 2^{-\ell'} \right\rfloor \mod p, \quad (16)$$

$$c_{de}^{(i)} = c_i \bmod 1 = \left\lfloor \frac{T \cdot j_i}{2^{\lceil \log_2 T \rceil}} \right\rfloor \cdot 2^{-\ell'} \bmod 1. \quad (17)$$

where $\ell' = \lambda - \lceil \log_2 T \rceil$. Note that $c_{in}^{(i)} \in \mathbb{Z}_p$ and $c_{de}^{(i)} \in [0, 1)$. Note also that the only component that depends on s is $\mathbf{a}$, but due to "mod 1," the decimal part $c_{de}^{(i)}$ becomes independent of it. Similarly to the first linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$, we can define the decomposed functions $f_1(pp, s, \mathbf{x})$ and $f_2(pp, \mathbf{x})$ as follows:

$f_1(pp, s, \mathbf{x})$: This function is a probabilistic function, which first picks $\mathbf{a} = (a_1, \ldots, a_n) \in h_z^{-1}(s)$ uniformly at random, and outputs $\mathbf{c}_1 = (c_{in}^{(1)}, \ldots, c_{in}^{(n)}) \in (\mathbb{Z}_p)^n$, where each $c_{in}^{(i)}$ is computed as Eq. (16).

$f_2(pp, \mathbf{x})$: This function outputs $\mathbf{c}_2 = (c_{de}^{(1)}, \ldots, c_{de}^{(n)}) \in [0, 1)^n$, where each $c_{de}^{(i)}$ is computed as Eq. (17).

Furthermore, it is not hard to see that f_1 satisfies the partial hiding property. Specifically, recall that given $s \in \mathbb{Z}_p$, Sketch chooses $\mathbf{a} = (a_1, \ldots, a_n) \in (\mathbb{Z}_p)^n$ uniformly at random from the set $h^{-1}(s)$. Due to the linearity of the underlying universal hash family $\mathcal{H}_{lin}$, if $s \in \mathbb{Z}_p$ is also chosen uniformly at random, $\mathbf{a}$ is distributed uniformly in $(\mathbb{Z}_p)^n$. Hence, each a_i acts as a key for one-time pad encryption, and thus each $c_{in}^{(i)} \in \mathbb{Z}_p$ does not contain any information of x_i. Therefore, when $s \in \mathbb{Z}_p$ is chosen uniformly at random, the output $\mathbf{c}_1$ of $f_1(pp, s, \mathbf{x})$ becomes independent of $\mathbf{x}$. This completes the proof of Lemma 5. □

We now show how secure the linear sketch scheme $\mathcal{S}_{\mathtt{Hash}}$ is against recovering attacks.

Theorem 2. *The linear sketch scheme $\mathcal{S}_{\mathtt{Hash}}$ satisfies $(p^{-1} \cdot k^{-\omega(1)})$-security against recovering attacks.*

Proof of Theorem 2. Due to Lemmas 3 and 5, and the fact that f_2 does not use a public parameter $pp = z$ and thus independent of $\mathcal{P}$, we have

$$\mathsf{Adv}^{\mathtt{recover}}_{\mathcal{S}_{\mathtt{CRT}}}(k) = 2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X} | \mathcal{P}, f_2(\mathcal{P}, \mathcal{X}))} = 2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X} | f_2(\mathcal{P}, \mathcal{X}))}.$$

Moreover, it is straightforward to see that the distribution $f_2(\mathcal{P}, \mathcal{X})$ is exactly the distribution $\mathcal{X}'_{de}$ introduced in Section 6.1, namely, the distribution of the vector of the "decimal"-part of the scaled-up version $\mathcal{X}'$ of $\mathcal{X}$. Note also that given $\mathcal{X}'_{de}$, guessing the original distribution $\mathcal{X}$ is a harder task than guessing the "integer"-part $\mathcal{X}'_{in}$. Hence, we have $\widetilde{\mathbf{H}}_\infty(\mathcal{X} | f_2(\mathcal{P}, \mathcal{X})) \geq \widetilde{\mathbf{H}}_\infty(\mathcal{X}'_{in} | \mathcal{X}'_{de}) \geq \log_2 p + \omega(\log_2 k)$. Due to Eq. (14), we can estimate $\mathsf{Adv}^{\mathtt{recover}}_{\mathcal{S}_{\mathtt{Hash}}}(k)$ as follows:

$$\begin{aligned} \mathsf{Adv}^{\mathtt{recover}}_{\mathcal{S}_{\mathtt{Hash}}}(k) &= 2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X} | f_2(\mathcal{P}, \mathcal{X}))} \leq 2^{-\widetilde{\mathbf{H}}_\infty(\mathcal{X}'_{in} | \mathcal{X}'_{de})} \\ &\leq 2^{-(\log_2 p + \omega(\log_2 k))} = p^{-1} \cdot k^{-\omega(1)}. \end{aligned}$$

Hence, $\mathcal{S}_{\mathtt{Hash}}$ satisfies $(p^{-1} \cdot k^{-\omega(1)})$-security against recovering attacks. This completes the proof of Theorem 2. □

$\mathsf{Setup}(1^k)$:	$\mathsf{Sketch}(pp, s \in \mathbb{Z}_p, \mathbf{x} \in [0,1)^n)$:	$\mathsf{DiffRec}(pp, \mathbf{c}, \mathbf{c}')$:
$z \leftarrow_{\mathrm{R}} \mathbb{F}_{p^n}$	$\mathbf{a} \leftarrow_{\mathrm{R}} h_z^{-1}(s)$	$\Delta\mathbf{c} \leftarrow \mathbf{c}' - \mathbf{c}$
Return $pp \leftarrow z$.	$\mathbf{c} \leftarrow \mathbf{a} + T \cdot \mathbf{x}$	$\Delta s \leftarrow h_z(\lfloor \Delta\mathbf{c} \rceil)$
	Return $\mathbf{c}$.	Return Δs.

Figure 3: The linear sketch scheme $\mathcal{S}_{\mathtt{Hash}} = (\mathsf{Setup}, \mathsf{Sketch}, \mathsf{DiffRec})$ for the fuzzy key setting $\mathcal{F}_2$. In the figure, the additions/subtractions are done in $(\mathbb{R}_p)^n$.

REFERENCES

Cheraghchi, M. (2011). Capacity achieving codes from randomness condensers. http://arxiv.org/pdf/0901.1866v2. pdf. Preliminary version appeared in ISIT 2009.

Dodis, Y., Ostrovsky, R., Reyzin, L., and Smith, A. (2008). Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139.

Goldwasser, S., Micali, S., and Rivest, R. (1988). A digital signature schemes secure against adaptive chosen-message attacks. *SIAM J. Computing*, 17(2):281–308.

Matsuda, T., Takahashi, K., Murakami, T., and Hanaoka, G. (2016). Fuzzy signatures: Relaxing requirements and a new construction. In *Proc. of ACNS 2016*, volume 9696 of *LNCS*, pages 97–116. Springer.

Schnorr, C. (1990). Efficient signature generation for smart cards. In *Proc. of CRYPTO 1989*, volume 435 of *LNCS*, pages 239–252. Springer.

Takahashi, K., Matsuda, T., Murakami, T., Hanaoka, G., and Nishigaki, M. (2015). A signature scheme with a fuzzy private key. In *Proc. of ACNS 2015*, volume 9092 of *LNCS*, pages 105–126. Springer.

Takahashi, K., Matsuda, T., Murakami, T., Hanaoka, G., and Nishigaki, M. (2017). Signature schemes with a fuzzy private key. IACR Cryptology ePrint Archive: Report 2017/1188. https://eprint.iacr.org/2017/1188. This is the merged full version of (Takahashi et al., 2015) and (Matsuda et al., 2016).

Waters, B. (2005). Efficient identity-based encryption without random oracles. In *Proc. of EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer.

Yasuda, M., Shimoyama, T., Takenaka, M., Abe, N., Yamada, S., and Yamaguchi, J. (2017). Recovering attacks against linear sketch in fuzzy signature schemes of ACNS 2015 and 2016. In *Proc. of ISPEC 2017*, volume 10701 of *LNCS*, pages 409–421. Springer.

APPENDIX

Here, we provide a minimal definition of a fuzzy signature scheme. Then, in order to grasp how the linear sketch schemes $\mathcal{S}_{\mathtt{CRT}}$ and $\mathcal{S}_{\mathtt{Hash}}$ are used to construct fuzzy signature schemes, we briefly review the first and second fuzzy signature schemes in (Takahashi et al., 2017).

Syntax. A fuzzy signature scheme Σ_{FS} for a fuzzy key setting $\mathcal{F} = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \varepsilon)$ 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 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").

Let $\delta = \delta(k) \in [0,1]$. A fuzzy signature scheme Σ_{FS} for a fuzzy key setting $\mathcal{F} = (\mathsf{d}, X), t, \mathcal{X}, \Phi, \varepsilon_e)$ satisfies δ-correctness if it holds that

$$\Pr\Big[\, pp \leftarrow_{\mathrm{R}} \mathsf{Setup}_{\mathsf{FS}}(1^k);\ x \leftarrow_{\mathrm{R}} \mathcal{X};\ vk \leftarrow_{\mathrm{R}} \mathsf{KG}_{\mathsf{FS}}(pp, x);\ e \leftarrow_{\mathrm{R}} \Phi;\ \sigma \leftarrow_{\mathrm{R}} \mathsf{Sign}_{\mathsf{FS}}(pp, x+e, m) : \mathsf{Ver}_{\mathsf{FS}}(pp, vk, m, \sigma) = \top \,\Big] \geq 1 - \delta$$

for all $k \in \mathbb{N}$ and all messages m.

In (Takahashi et al., 2017), as a security requirement for a fuzzy signature scheme, an analogue of the standard existential unforgeability against chosen message attacks (EUF-CMA security) (Goldwasser et al., 1988) was defined. We omit the formal definition since we do not directly deal with it in this paper.

First Scheme Σ_{FS1}. This scheme is constructed for the specific fuzzy key setting $\mathcal{F}_1$, from the combination of the linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$ and a variant of the Waters scheme (Waters, 2005) (called the *modified Waters signature (MWS)* scheme in (Takahashi et al., 2017)). (For the fuzzy key setting $\mathcal{F}_1$ and the linear sketch scheme $\mathcal{S}_{\mathtt{CRT}}$, see Sections 5.1 and 5.3, respectively.)

Let $\ell = \ell(k)$ be a positive polynomial that denotes the length of messages. Let $\mathcal{F}_1 = ((\mathsf{d}, X), t, \mathcal{X}, \Phi, \varepsilon_e)$ be the fuzzy key setting defined in Section 5.1, where

$\mathsf{Setup}_{\mathsf{FS1}}(1^k)$:
$g, h, u', u_1, \ldots, u_\ell \leftarrow_{\mathtt{R}} \mathbb{G}$
Let z be an element of $\mathbb{Z}_p^*$ of order W.
Return $pp \leftarrow (g, h, u', (u_i)_{i\in[\ell]}, z)$.

$\mathsf{KG}_{\mathsf{FS1}}(pp, \mathbf{x})$:
$sk \leftarrow_{\mathtt{R}} \mathbb{Z}_W$; $vk \leftarrow g^{z^{sk}}$
$\mathbf{c} \leftarrow (\mathsf{CRT}_{\mathbf{w}}^{-1}(sk) + \mathsf{E}_{\mathbf{w}}(\mathbf{x})) \bmod \mathbf{w}$
Return $VK \leftarrow (vk, \mathbf{c})$.

$\mathsf{Sign}_{\mathsf{FS1}}(pp, \mathbf{x}', m)$:
Parse m as $(m_1 \| \ldots \| m_\ell) \in \{0,1\}^\ell$.
$\widetilde{sk} \leftarrow_{\mathtt{R}} \mathbb{Z}_W$; $\widetilde{vk} \leftarrow g^{z^{\widetilde{sk}}}$; $r \leftarrow_{\mathtt{R}} \mathbb{Z}_p$; $\widetilde{\sigma}_2 \leftarrow g^r$
$\widetilde{\sigma}_1 \leftarrow h^{z^{\widetilde{sk}}} \cdot (u' \cdot \prod_{i\in[\ell]} u_i^{m_i})^r$
$\widetilde{\mathbf{c}} \leftarrow (\mathsf{CRT}_{\mathbf{w}}^{-1}(\widetilde{sk}) + \mathsf{E}_{\mathbf{w}}(\mathbf{x}')) \bmod \mathbf{w}$
Return $\sigma \leftarrow (\widetilde{vk}, \widetilde{\sigma}_1, \widetilde{\sigma}_2, \widetilde{\mathbf{c}})$.

$\mathsf{Ver}_{\mathsf{FS1}}(pp, VK, m, \sigma)$:
$(vk, \mathbf{c}) \leftarrow VK$; $(\widetilde{vk}, \widetilde{\sigma}_1, \widetilde{\sigma}_2, \widetilde{\mathbf{c}}) \leftarrow \sigma$
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 $\bot$.
$\Delta\mathbf{s} \leftarrow \mathsf{C}_{\mathbf{w}}(\widetilde{\mathbf{c}} - \mathbf{c})$; $\Delta sk \leftarrow \mathsf{CRT}_{\mathbf{w}}(\Delta\mathbf{s})$
If $(vk)^{z^{\Delta sk}} = \widetilde{vk}$ then return $\top$ else return $\bot$.

$\mathsf{Setup}_{\mathsf{FS2}}(1^k)$:
$g \leftarrow_{\mathtt{R}} \mathbb{G}$; $z \leftarrow_{\mathtt{R}} \mathbb{F}_{p^n}$
Let $H : \{0,1\}^* \to \mathbb{Z}_p$
be a cryptographic hash function.
Return $pp \leftarrow (g, z, H)$.

$\mathsf{KG}_{\mathsf{FS2}}(pp, \mathbf{x})$:
$sk \leftarrow_{\mathtt{R}} \mathbb{Z}_p$; $vk \leftarrow g^{sk}$
$\mathbf{a} \leftarrow_{\mathtt{R}} h_z^{-1}(sk)$; $\mathbf{c} \leftarrow \mathbf{a} + T \cdot \mathbf{x}$
Return $VK \leftarrow (vk, \mathbf{c})$.

$\mathsf{Sign}_{\mathsf{FS2}}(pp, \mathbf{x}', m)$:
$\widetilde{sk} \leftarrow_{\mathtt{R}} \mathbb{Z}_p$; $\widetilde{vk} \leftarrow g^{\widetilde{sk}}$; $r \leftarrow_{\mathtt{R}} \mathbb{Z}_p$; $R \leftarrow g^r$
$\widetilde{h} \leftarrow H(R \| m)$; $\widetilde{s} \leftarrow r + (\widetilde{sk}) \cdot \widetilde{h} \bmod p$
$\mathbf{a}' \leftarrow_{\mathtt{R}} h_z^{-1}(\widetilde{sk})$; $\widetilde{\mathbf{c}} \leftarrow \mathbf{a}' + T \cdot \mathbf{x}'$
Return $\sigma \leftarrow (\widetilde{vk}, \widetilde{h}, \widetilde{s}, \widetilde{\mathbf{c}})$.

$\mathsf{Ver}_{\mathsf{FS2}}(pp, VK, m, \sigma)$:
$(vk, \mathbf{c}) \leftarrow VK$; $(\widetilde{vk}, \widetilde{h}, \widetilde{s}, \widetilde{\mathbf{c}}) \leftarrow \sigma$
$R \leftarrow g^{\widetilde{s}} \cdot (\widetilde{vk})^{-\widetilde{h}}$
If $H(R \| m) \neq \widetilde{h}$ then return $\bot$.
$\Delta\mathbf{c} \leftarrow \widetilde{\mathbf{c}} - \mathbf{c}$; $\Delta sk \leftarrow h_s(\lfloor \Delta\mathbf{c} \rceil)$
If $vk \cdot g^{\Delta sk} = \widetilde{vk}$ then return $\top$ else return $\bot$.

Figure 4: (**Left**) The description of the fuzzy signature scheme $\Sigma_{\mathsf{FS1}} = (\mathsf{Setup}_{\mathsf{FS1}}, \mathsf{KG}_{\mathsf{FS1}}, \mathsf{Sign}_{\mathsf{FS1}}, \mathsf{Ver}_{\mathsf{FS1}})$, and (**Right**) the description of the fuzzy signature scheme $\Sigma_{\mathsf{FS2}} = (\mathsf{Setup}_{\mathsf{FS2}}, \mathsf{KG}_{\mathsf{FS2}}, \mathsf{Sign}_{\mathsf{FS2}}, \mathsf{Ver}_{\mathsf{FS2}})$. In both of the descriptions, the steps related to the underlying linear sketch schemes are highlighted with the box.

t (and n) are determined according to the security parameter k. let $\mathbf{w} = (w_1, \ldots, w_n) = \mathsf{WGen}(t, n)$, where n is the dimension of X, and let $W = \prod_{i\in[n]} w_i$. Let $\mathsf{CRT}_{\mathbf{w}}$, $\mathsf{CRT}_{\mathbf{w}}^{-1}$, $\mathsf{E}_{\mathbf{w}}$, and $\mathsf{C}_{\mathbf{w}}$ be the functions defined in Section 5.2. Let $(\mathbb{G}, \mathbb{G}_T, e, p)$ be a description of symmetric bilinear groups with prime order $p = \Omega(2^k)$ such that $W | p-1$, where $\mathbb{G}$ and $\mathbb{G}_T$ are groups of prime order p and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficiently computable bilinear map.[4]

Then, using these ingredients, the fuzzy signature scheme $\Sigma_{\mathsf{FS1}} = (\mathsf{Setup}_{\mathsf{FS1}}, \mathsf{KG}_{\mathsf{FS1}}, \mathsf{Sign}_{\mathsf{FS1}}, \mathsf{Ver}_{\mathsf{FS1}})$ for the fuzzy key setting $\mathcal{F}_1$ is constructed as in Figure 4 (left).

It was shown in (Takahashi et al., 2017) that this fuzzy signature scheme satisfies ε_e-correctness, and is secure if the computational Diffie-Hellman assumption holds in $\mathbb{G}$.

Second Scheme Σ_{FS2}. This scheme is constructed for the specific fuzzy key setting $\mathcal{F}_2$, from the combination of the linear sketch scheme $\mathcal{S}_{\mathtt{Hash}}$ and the Schnorr signature scheme (Schnorr, 1990). (For the fuzzy key setting $\mathcal{F}_2$ and the linear sketch scheme $\mathcal{S}_{\mathtt{Hash}}$, see Sections 6.1 and 6.2, respectively.)

Let $\mathcal{F}_2 = ((d, X), t, \mathcal{X}, \Phi, \varepsilon_e)$ be the fuzzy key setting that we specified in Section 6.1, and suppose the dimension of the fuzzy data space is n. Let $(\mathbb{G}, p)$ be the description of a group $\mathbb{G}$ of prime order $p = \Omega(2^k)$. 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 as guaranteed by Lemma 2. (We identify $\mathbb{F}_p$ with $\mathbb{Z}_p$.) Let $H : \{0,1\}^* \to \mathbb{Z}_p$ be a cryptographic hash function (modeled as a random oracle).

Using these building blocks, the second fuzzy signature scheme $\Sigma_{\mathsf{FS2}} = (\mathsf{Setup}_{\mathsf{FS2}}, \mathsf{KG}_{\mathsf{FS2}}, \mathsf{Sign}_{\mathsf{FS2}}, \mathsf{Ver}_{\mathsf{FS2}})$ for the fuzzy key setting $\mathcal{F}_2$ is constructed as in Figure 4 (right).

It was shown in (Takahashi et al., 2017) that this fuzzy signature scheme satisfies ε_e-correctness, and is secure in the random oracle model (where H is modeled as a random oracle) if the discrete logarithm assumption holds in $\mathbb{G}$.

[4]For all generators $g \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, it holds that $e(g^a, g^b) = e(g, g)^{ab} \in \mathbb{G}_T$.