IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 April 2023, accepted 8 June 2023, date of publication 19 June 2023, date of current version 26 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3287777

RESEARCH ARTICLE

A Provably Secure Lattice-Based Fuzzy Signature Scheme Using Linear Sketch

MINGMEI ZHENG[1], ZI-YUAN LIU[1], AND MASAHIRO MAMBO[2], (Member, IEEE)
[1]Graduate School of Natural Science and Technology, Kanazawa University, Kanazawa 920-1192, Japan
[2]Institute of Science and Engineering, Kanazawa University, Kanazawa 920-1192, Japan

Corresponding author: Mingmei Zheng (mmzheng@stu.kanazawa-u.ac.jp)

This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan; and in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI, Japan, under Grant JP21K11885.

ABSTRACT Fuzzy signatures (FS) are a kind of signature scheme that employs a noisy string (e.g., biometric data) as the secret key without requiring the user-specific auxiliary data. As the quantum computing era approaches, some research has been dedicated to developing quantum-resistant FS schemes, which can be classified into fuzzy extractor (FE) approach and linear sketch (LS) approach. However, the existing schemes utilizing FEs to obtain (variants of) fuzzy signatures require to produce the user-specific auxiliary information known as helper data to retrieve secret keys, leading to an additional computational cost. In light of the circumstance, we seek to construct a fuzzy signature scheme by employing a linear sketch, since this approach does not require the user-specific auxiliary data to derive secret keys. We modify the linear sketch which is an essential ingredient of the most practical fuzzy signature proposed by Katsumata et al. (CCS' 21). Then we combine it with Lyubashevsky's lattice-based signature scheme (EUROCRYPT' 12) to construct our lattice-based fuzzy signature scheme. Moreover, to further demonstrate the security of our proposed scheme, we provide a rigorous security proof in the random oracle model. Finally, the comparison indicates that our proposed FS scheme not only avoids the use of FE but also shows a promising tendency in efficiency among the existing quantum-resistant FS schemes.

INDEX TERMS Biometrics, fuzzy signatures, lattice-based signatures, quantum resistance.

I. INTRODUCTION

Digital signatures are an indispensable component of modern cryptography, which is the cornerstone of information security. It is widely used in the fields of communication, electronic commerce and national defense because of its non-repudiation, data integrity and unforgeability. Many kinds of signatures are proposed to satisfy the needs of the society towards security services in many different scenarios. A kind of special digital signature, called fuzzy signature [1], can offer better usability and security of the secret keys by using noisy data (e.g., biometric data) instead of traditional number-based passwords.

The associate editor coordinating the review of this manuscript and approving it for publication was Barbara Masucci.

As we all know, the security of modern cryptographic applications is usually based on the secret keys. Thus the users need to keep his/her secret keys carefully, and they may keep their secret keys on a USB token or a smart card and remember a password to activate it. Hence, carrying an additional device is unavoidable for the users in such cases. This limitation causes some inconvenience and reduces usability. One of the promising solutions is to use biometric data as secret keys, like fingerprints, iris and faces, since they are parts of our body, and unique for everyone. Biometrics has a wide range of applications, especially the cybersecurity and personal privacy. However, biometric data could not be directly applied into the cryptographic scheme (as the signing keys), since it is not uniformly distributed and fluctuates each time when it is captured. Therefore, there are several methods proposed to address this issue, such as fuzzy extractors

(FEs)[1] [2], [3], [4], [5], [6] and fuzzy signatures [1], [5], [7], [8]. FEs rely on the user-specific auxiliary information called helper data to retrieve its secret keys. In contrast, fuzzy signatures allow users to use their biometric data as secret keys to generate signatures without relying on user-specific auxiliary data. Such fuzzy signatures may bring a more straightforward way to solve the above problem. Therefore, we focus on fuzzy signatures in this paper.

Fuzzy signatures [1], known as generating a signature with a noisy string, are a kind of digital signature that utilizes a noisy string (e.g., biometric data) as a secret key to generate a signature without depending on the user-specific auxiliary data. For a fuzzy signature scheme, the key generation takes a noisy string x and a public parameter pp_{FS} produced by the setup algorithm as inputs, and outputs a public key pk. The signing algorithm takes a message m and another noisy string x' as inputs, and outputs a signature σ. The verification algorithm takes a message-signature pair (m, σ) and pk as inputs, and outputs 1 when σ is a valid signature on message m; Otherwise, outputs 0. That is, a message m signed by the fuzzy data x' can be verified by the public key pk generated by another fuzzy data x close to x'.

To our knowledge, using linear sketch [1], [7], [9] to construct a fuzzy signature is a more promising approach. The primitive, linear sketch, is an important building block used to cope with fuzzy data in a fuzzy signature scheme. Linear sketch (scheme), formally defined by [1], consists of three algorithms (LS.Setup, LS.Sketch, LS.DiffRec). The algorithm LS.Sketch takes a public parameter pp_{LS} produced by the algorithm LS.Setup and a fuzzy data x as inputs, and outputs a sketch c and a proxy key S (which is regarded as a secret key). The difference reconstruction algorithm LS.DiffRec takes pp_{LS}, c, and c' as inputs, and outputs the difference $\Delta S = S' - S$. Therefore, we can get a secret key S directly by exploiting the fuzzy data x from the algorithm LS.Sketch. In addition, there are few attentions on quantum-resistant fuzzy signatures constructed by linear sketch approach [8]. Hence, it is non-trivial to construct the quantum-resistant fuzzy signature scheme based on the linear sketch.

A. OUR CONTRIBUTION

The main contributions of our work are summarized as follows:

- In this work, we propose a lattice-based fuzzy signature scheme that is constructed by a modified linear sketch. More specifically, we provide a mapping h to replace the universal hash function $\mathcal{UH}$[2] used for the linear sketch proposed by Katsumata et al. [7] so as to make it applicable to a lattice-based scheme of [10]. Since the modification we made on the linear sketch of [7] keeps the original functionality and structure, our modified linear sketch inherits the conceptually clean construction from Katsumata et al. [7] which proposed the first fuzzy signature implemented efficiently and securely. That makes our proposed scheme not only present theoretical achievements, but also have the probability of being implemented.
- We modify the Lyubashevsky's lattice-based signature scheme [10], and combine it with the above modified linear sketch to obtain a fuzzy signature scheme whose security relies on a lattice-based hardness assumption. To further illustrate the security of our scheme, we give a rigorous security proof in the random oracle model. In addition, Table 1 indicates that our scheme obtains a promising result in efficiency among the existing lattice-based fuzzy signature schemes.

B. OUR APPROACH

The work aims to propose a lattice-based fuzzy signature scheme by utilizing linear sketch. Since most of the existing fuzzy signatures constructed by linear sketch [1], [7], [9], [13] are based on traditional number-theoretical assumptions, their linear sketch schemes cannot be directly employed in the lattice-based setting. We modify the linear sketch from the scheme of [7] which proposed the first fuzzy signature scheme that can be implemented securely and efficiently, since we hope that the novel construction of the modified linear sketch can not only make our scheme have the probability of being implemented, but also be applicable to the schemes from lattice. In our approach, we use a mapping h to replace an universal hash function of the algorithm LS.Sketch of linear sketch scheme in [7]. This modification makes the modified linear sketch not only capable of being applied into a lattice-based signature scheme, but also keep the original functionality and structure unchanged.

In the security proof part, there are two possible approaches of responding the signing queries from the forger during signing, either using the secret key, or directly generating signatures chosen randomly from a distribution without utilizing secret key. Even though our scheme is based on [10], we do not prove the security of our scheme in the same way as [10]. We choose the former method, since we could not easily produce signatures without the help of the proxy key $\mathbf{S}'$ in the signing queries. Hence, in order to successfully simulate a signature which can be indistinguishable from the actual signature during the signing queries, we choose the former method.

C. RELATED WORK

The concept of fuzzy signatures was first introduced by Takahashi et al. [1] which not only provided the formal definition of fuzzy signatures including two building blocks, fuzzy key setting and linear sketch, but also proposed a generic construction of fuzzy signature from an ordinary

[1]FE includes two algorithms (Gen, Rep), where Gen takes a noisy string x as input, and outputs a helper string P and an extracted key r; Algorithm Rep takes as inputs another noisy data x' and the helper string P, then reproduces r if x and x' are close enough.

[2]$\mathcal{UH} = \{UH : D \to R\}$ is called universal if for all distinct elements $x, x' \in D$, we have that $\Pr_{UH \leftarrow \mathcal{UH}}[UH(x) = UH(x')] \leq |R|^{-1}$.

TABLE 1. Comparison with the related works of lattice-based fuzzy signature schemes in the random oracle mode.

Scheme	Assumption	Method	Communication Cost	Computational Cost	
			Signature	Signing	Verification
SW21 [6]	SIVP	FE	$2V$	$t_H + 2t_M + 2t_P$	$t_H + 2t_M$
TLD21 [5]	LWE,SIS	FE	$2M + 3V$	$t_H + 3t_M + t_P$	$2t_H + 3t_M$
KK22 [8]	LWE	LS	$(2T + 4)V$	$2t_H + (4T + 2)t_M$	$2t_H + (2T + 1)t_M$
Our Σ_{FS}	SIS	LS	$3V$	$t_H + 7t_M$	$t_H + 6t_M$

[1] Let t_H, t_M be the time of computing one time of hash operation, and multiplication operation regarding matrices and vectors, respectively. Let t_P be the time of running one time of algorithms of FE connecting to helper data P to derive the secret key. Let M/V be matrix/vector, and let integer $T \geq 1$. FE/LS in the method represents fuzzy extractor/linear sketch;
[2] "Method" represents the way to construct these fuzzy signatures;
[3] We compare the size of signatures in terms of the number of "basic elements" represented by matrix and vector as in [11], [12], and we here omit the comparison in the size of public keys since all of them are the same.

signature satisfying homomorphic property regarding keys. Then Matsuda et al. [9] provided a relaxed version of fuzzy signatures by relaxing some requirements on the building blocks, like employing the ordinary signatures having the weaker form of homomorphic property, e.g., Waters signatures [14] were replaced with Schnorr signatures [15] in their proposed instantiations. In 2017, Yasuda et al. [16] claimed that the linear sketch of [1] and [9] is vulnerable to their "recovering attacks" since the treatment of real numbers in the linear sketch. After that, Takahashi et al. [13] gave a treatment of rounding-down operation (or called truncation) on the decimal part of real numbers to address that problem. However, such a method caused correctness loss in their proposed schemes.

Katsumata et al. [7] in 2021 proposed a simpler, more efficient and direct construction of fuzzy signatures by exploiting the Schnorr signatures [15] with a simpler linear sketch based on a mathematical object, called lattice. They showed that this fuzzy signature scheme can be efficiently and securely implemented with the help of some novel statistical techniques. They also gave the experimental results of using the real-world finger-vein database to show that the finger-vein from one hand is enough to construct secure and efficient fuzzy signatures. The work of [7] made a breakthrough for fuzzy signatures by widening theory oriented research into practical one.

Furthermore, little attention is paid on (variants of) quantum-resistant fuzzy signatures [5], [6], [8]. More concretely, a concrete instantiation of reusable fuzzy signature in [5] was built up on a reusable FE [17] based on learning with errors (LWE). The work in [6] proposed a variant of fuzzy signature scheme by utilizing a lattice-based signature scheme of [10] and a FE of [2]. Specifically, the scheme of [6] redefined the definition of fuzzy signatures proposed by [1] and relaxed the security model of [1] by relaxing the requirement of error distribution of fuzzy data.

To the best of our knowledge, there just exists a fuzzy signature scheme [8] which is against the quantum computers and constructed by the linear sketch. The work of [8] proposed a generic construction of fuzzy signatures constructed by a linear sketch and an ordinary signature scheme Σ_{SS}, and then instantiate it by giving a concrete LWE-based signature scheme and an instantiation of linear sketch. We hightlight that our scheme is not an instantiation of [8]. The construction of signing algorithm, verification algorithm, and the linear sketch of the generic construction in [8] are different from ours in various viewpoints. Specifically, in the signing algorithm of [8], it reused the key generation algorithm of Σ_{SS} to produce another pair of public/secret keys whose secret key was used as the input of the signing algorithm of Σ_{SS} and the linear sketch. For our scheme, we do not utilize the key generation algorithm again in our signing phase, and we just use an output of our linear sketch as the secret key. Moreover, compared to the scheme of [8], we give a rigorous security proof in the random oracle model.

In addition, although our scheme seems not to approach the optimal results in Table 1, it shows a promising tendency in efficiency among the existing lattice-based fuzzy signature schemes.

D. ROADMAP

The remainder of this paper is arranged as follows. Section II introduces some preliminaries used in this paper. Linear sketch, an essential building block of fuzzy signatures, is recalled in Section III. In Section IV, we propose our lattice-based fuzzy signature scheme and give its security analysis. Furthermore, Section V concludes this paper.

II. PRELIMINARIES

In this section, we recall some basic notations, results and definitions that will be used in the paper.

A. NOTATION

Throughout the paper, we denote $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{Z}$ by the set of real numbers, natural number and integers. We use log to denote the logarithm of base 2. $\kappa \in \mathbb{N}$ denotes the security parameter, and let q be a polynomial-size prime number. Vectors are denoted by bold lower-case letters (e.g., $\mathbf{x}$), and matrices are represented by bold capital letters (e.g., $\mathbf{X}$). Let all vectors be column vectors, and $\mathbf{x}^\top$ will be the transpose of the vector $\mathbf{x}$. $\|\mathbf{x}\|_2$ is denoted by the ℓ_2 norm of a vector $\mathbf{x}$. The notation of $\leftarrow$ denotes the random selection of the elements from

some sets or distributions. We denote deterministic polynomial time (resp. probabilistic polynomial time) by DPT (resp. PPT). A function $f(n)$ is negligible in n if for any positive c, large enough n, we have $f(n) < n^{-c}$. We use standard notation big-O and big-Ω to classify the growth of functions $f(x)$ and $g(x)$ which map positive integers to non-negative real numbers. We say that $f(n) = O(g(n))$ if there exist $c_1 > 0$ and $N_1 \in \mathbb{N}$ such that $f(n) \leq c_1 \cdot g(n)$ for all $n \geq N_1$. And we say that $f(n) = \Omega(g(n))$ if there exists $c_2 > 0$ and $N_2 \in \mathbb{N}$ such that $f(n) \geq c_2 \cdot g(n)$ for all $n \geq N_2$. Note that all operations include the elements in $\mathbb{Z}$ involved end with a reduction modulo q. That means we usually omit to write the modulo q in such equations. For example, the product of a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times n}$ and a vector $\mathbf{y} \in \mathbb{Z}^n$ is a vector in $\mathbb{Z}_q^n$.

B. DIGITAL SIGNATURES

We recall the definition of digital signature schemes.

Definition 1 (Signature Schemes): A signature scheme Σ is a triple $(\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ of PPT algorithms together with message space $\mathcal{M}$. It is correct if for any message $\mu \in \mathcal{M}$, the algorithm holds $\mathsf{Vrfy}(pk, \mu, \sigma) = 1$ except with negligible probability in κ over the choice of $(pk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$ and $\sigma \leftarrow \mathsf{Sign}(sk, \mu)$.

A signature scheme is said to be secure if there is only a negligible probability that any forger, after seeing signatures of messages of his choosing, can sign a message whose signature he has not already seen [18]. The standard security notion for digital signature schemes is existentially unforgeable under adaptative chosen message attacks (EUF-CMA) [11], [12] which is usually given as a game. It requires that a forger $\mathcal{F}$ could not be able to come up with a valid signature of a new message after he adaptively queries the messages. Formally, consider the following EUF-CMA game between a challenger $\mathcal{C}$ and a forger $\mathcal{F}$.

- **KGen:** The challenger $\mathcal{C}$ first runs $(pk, sk) \leftarrow \mathsf{KGen}(1^\kappa)$. It then sends the public key pk to the forger $\mathcal{F}$, and keeps secret key sk by itself.
- **Signing:** The forger $\mathcal{F}$ is allowed to query messages adaptively. When $\mathcal{F}$ asks the signature on any fresh message M, the challenger $\mathcal{C}$ computes and sends $\sigma_M \leftarrow \mathsf{Sign}(sk, M)$ to $\mathcal{F}$. The forger can repeat this queries in any polynomial time.
- **Forge:** Finally, the forger $\mathcal{F}$ outputs a message-signature pair (M^*, σ_{M^*}), and let Q be the set of all messages queried by $\mathcal{F}$. The challenger $\mathcal{C}$ outputs 1 if $M^* \notin Q$ and $\mathsf{Vrfy}(pk, M^*, \sigma_{M^*}) = 1$, else outputs 0.

A signature scheme $\Sigma = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ is EUF-CMA secure if there is no PPT forger wins the above EUF-CMA game with a non-negligible probability.

Definition 2 (EUF-CMA Security): Let κ be the security parameter. A signature scheme Σ is said to be existentially unforgeable against chosen message attacks if the advantage $\mathsf{Adv}_{\Sigma,\mathcal{F}}^{euf-cma}(1^\kappa) = \Pr[\mathcal{C} \text{ outputs } 1]$ is negligible in κ for all PPT adversaries.

C. LATTICE AND GAUSSIAN DISTRIBUTION

A (full-rank) m-dimension lattice $\mathcal{L}(\mathbf{B}) = \{\mathbf{Bz} : \mathbf{z} \in \mathbb{Z}^m\}$ is the set of all integer linear combinations of m linearly independent vectors $\mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_m] \in \mathbb{R}^{m\times m}$. There is a special lattice family called q-ary lattices, which contains $q\mathbb{Z}^m$ as a sublattice for some small integer q. Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ be a matrix with some positive $n, m, q \in \mathbb{Z}$, and consider the following m-dimension q-ary lattice:

$$\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{v} \in \mathbb{Z}^m : \mathbf{Av} = 0 \mod q\}.$$

Given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the small integer solution problem ($\mathsf{SIS}_{q,m,n,\beta}$ problem) asks to find a non-zero vector $\mathbf{v} \in \Lambda_q^\perp(\mathbf{A})$ such that $\mathbf{Av} = 0 \mod q$ and $\|\mathbf{v}\|_2 \leq \beta$. We then give the formal definition of $\mathsf{SIS}_{q,m,n,\beta}$ [19] as follows.

Definition 3: ($\mathsf{SIS}_{q,m,n,\beta}$ problem) Given a random matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, find a non-zero vector $\mathbf{v} \in \mathbb{Z}^m$ such that $\mathbf{Av} = 0 \mod q$ and $\|\mathbf{v}\|_2 \leq \beta$.

The following useful facts used in our paper are from [10], [11], [20], and [21].

Definition 4: The continuous Normal distribution over $\mathbb{R}^m$ centered at $\mathbf{v}$ with standard deviation σ is defined by the function $\rho_{\mathbf{v},\sigma}^m(\mathbf{x}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^m e^{\frac{-\|\mathbf{x}-\mathbf{v}\|^2}{2\sigma^2}}$.

The subscript $\mathbf{v}$ of the function $\rho_{\mathbf{v},\sigma}^m$ is omitted when $\mathbf{v}$ is taken to be 0. Note that for all $\mathbf{v} \in \mathbb{Z}^m$, $\rho_{\mathbf{v},\sigma}^m(\mathbb{Z}^m) = \rho_\sigma^m(\mathbb{Z}^m)$.

Definition 5: The discrete Normal distribution over $\mathbb{Z}^m$ centered at $\mathbf{v} \in \mathbb{Z}^m$ with standard deviation σ is defined as $D_{\mathbf{v},\sigma}^m(\mathbf{x}) = \frac{\rho_{\mathbf{v},\sigma}^m(\mathbf{x})}{\rho_{\mathbf{v},\sigma}^m(\mathbb{Z}^m)} = \frac{\rho_{\mathbf{v},\sigma}^m(\mathbf{x})}{\rho_\sigma^m(\mathbb{Z}^m)}$.

Lemma 1:
1) For any $k > 0$, $\Pr[|z| > k\sigma : z \leftarrow D_\sigma^1] \leq 2e^{\frac{-k^2}{2}}$
2) For any $\mathbf{z} \in \mathbb{Z}^m$, and $\sigma \geq \frac{3}{\sqrt{2\pi}}$, $D_\sigma^m(\mathbf{z}) \leq 2^{-m}$
3) For any $k > 1$, $\Pr[\|\mathbf{z}\| > k\sigma\sqrt{m} : \mathbf{z} \leftarrow D_\sigma^m] < k^m e^{\frac{m}{2}(1-k^2)}$.

Lemma 2: For any $\mathbf{v} \in \mathbb{Z}^m$, if $\sigma = \omega(\|\mathbf{v}\|\sqrt{\log m})$, then

$$\Pr[D_\sigma^m(\mathbf{z})/D_{\mathbf{v},\sigma}^m(\mathbf{z}) = O(1) : \mathbf{z} \leftarrow D_\sigma^m] = 1 - 2^{-\omega(\log m)},$$

and more specifically, for any $\mathbf{v} \in \mathbb{Z}^m$, if $\sigma = \alpha\|\mathbf{v}\|$ for any positive α, then

$$\Pr[D_\sigma^m(\mathbf{z})/D_{\mathbf{v},\sigma}^m(\mathbf{z}) \leq e^{12/\alpha+1/(2\alpha^2)} : \mathbf{z} \leftarrow D_\sigma^m] = 1 - 2^{-100}.$$

Lemma 3: For any positive integer $m \in \mathbb{Z}$, vector $y \in \mathbb{Z}^m$, and large enough $\sigma \geq \omega(\sqrt{\log m})$, we have that

$$\Pr_{x \leftarrow D_\sigma^m}[x = y] \leq 2^{1-m}.$$

Lemma 4: Let d be a small positive integer. For any $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ where prime integer q, positive integer n, and $m > 64 + n \cdot \frac{\log q}{\log(2d+1)}$, for randomly chosen $\mathbf{s} \leftarrow \{-d, \cdots, 0, \cdots, d\}^m$, then with probability $1 - 2^{-100}$, there exists another $\mathbf{s}' \leftarrow \{-d, \cdots, 0, \cdots, d\}^m$ such that $\mathbf{As} = \mathbf{As}'$.

Rejection sampling is a well-known technique introduced by John von Neumann [22] to sample from a target probability distribution f. Specifically, given a source bound to a different probability distribution g, and a sample x is drawn

from g and is accepted by probability $\frac{f(x)}{M \cdot g(x)}$, where $M \in \mathbb{R}^+$. We get the following theorem of rejection sampling from [10].

Theorem 1: Let V be a subset of $\mathbb{Z}^m$ in which all elements have norms less than T, σ be some element in $\mathbb{R}$ such that $\sigma = \omega(T\sqrt{\log m})$, and $h : V \to \mathbb{R}$ be a probability distribution. Then there exists a constant $M = O(1)$ such that the distribution of the following algorithm $\mathcal{A}$:

1) $\mathbf{v} \leftarrow h$
2) $\mathbf{z} \leftarrow D^m_{\mathbf{v},\sigma}$
3) output $(\mathbf{v}, \mathbf{z})$ with probability $\min\left(\frac{D^m_\sigma(\mathbf{z})}{MD^m_{\mathbf{v},\sigma}(\mathbf{z})}, 1\right)$

is within statistical distance $\frac{2^{-\omega(\log m)}}{M}$ of the distribution of the following algorithm $\mathcal{B}$:

1) $\mathbf{v} \leftarrow h$
2) $\mathbf{z} \leftarrow D^m_\sigma$
3) output $(\mathbf{v}, \mathbf{z})$ with probability $\frac{1}{M}$.

More concretely, if $\sigma = \alpha T$ for any positive α, then $M = e^{12/\alpha + 1/(2\alpha^2)}$, the output of algorithm $\mathcal{A}$ is within statistical distance $\frac{2^{-100}}{M}$ of the output of $\mathcal{B}$, and the probability that $\mathcal{A}$ outputs something is at least $\frac{1-2^{-100}}{M}$.

III. FUZZY SIGNATURES

Before recalling the definition of fuzzy signatures [7], [13], we first introduce its two important ingredients, fuzzy key setting and linear sketch. They are used to formalize how to deal with fuzzy data in a cryptographic scheme.

A. FUZZY KEY SETTING

The primitive, fuzzy key setting [1], [7], is an important building block of fuzzy signatures. A fuzzy key setting includes the below five parameters $(X, \mathcal{X}, \xi, \Phi, \epsilon)$, and it is used to formally treat fuzzy data in cryptographic schemes.

- **Fuzzy data space** X: This is the space to which a possible fuzzy data x belongs. Assume that X forms an Abelian group.
- **Distribution** $\mathcal{X}$: The distribution of fuzzy data over X. That is, $\mathcal{X} : X \to \mathbb{R}$.
- **Acceptance region function** $\xi : X \to 2^X$: This function maps from a fuzzy data $x \in X$ to a subspace $\xi(x) \subset X$, i.e., if $x' \in \xi(x)$, then x' is considered close to x. Two quantities, the false matching rate FMR[3] and the false non-matching rate FNMR[4] [23], are determined based on ξ. The FMR is defined as:

$$\mathsf{FMR} = \Pr[x, x' \leftarrow \mathcal{X} : x' \in \xi(x)].$$

- **Error distribution** Φ: The distribution models the measurement error of fuzzy data. Assume "universal error model" where the measurement error is independent of the users.
- **Error parameter** ϵ: The error parameter $\epsilon \in [0, 1]$ defines FNMR. That is,

$$\mathsf{FNMR} = \Pr[x \leftarrow \mathcal{X}; e \leftarrow \Phi : x + e \notin \xi(x)] \le \epsilon.$$

B. LINEAR SKETCH

Linear sketch was first formally defined by [1], and then the work of [7] proposed a simpler one and gave a specific construction. Linear sketch is an essential ingredient in the construction of fuzzy signatures [1], [7], [9], [13], and its main purpose is to "bridge" fuzzy data and cryptographic operations. It is related to the fuzzy key setting and consists of three algorithms. In the following, we describe the formal definition of linear sketch scheme [7].

Definition 6 (Linear Sketch): Let $\mathcal{K} = (X, \mathcal{X}, \xi, \Phi, \epsilon)$ be a fuzzy key setting with respect to a (finite) Abelian group $\Lambda = (\psi, +)$. A linear sketch scheme Σ_{LS} for $\mathcal{K}$ and Λ consists of the following three polynomial-time algorithms.

- $\mathsf{LS.Setup}(\mathcal{K}, \Lambda) \to pp_{LS}$: The setup algorithm takes the fuzzy key setting $\mathcal{K}$ and the description Λ as inputs, and outputs a public parameter pp_{LS}.
- $\mathsf{LS.Sketch}(pp_{LS}, x) \to (c, S)$: The deterministic sketch algorithm takes pp_{LS} and a fuzzy data $x \in X$ as inputs, and outputs a sketch c and a proxy key $S \in \psi$.
- $\mathsf{LS.DiffRec}(pp_{LS}, c, c') \to \Delta S$: The deterministic difference reconstruction algorithm takes as inputs the public parameter pp_{LS} and two sketches c and c' (c' is also output by the algorithm LS.Sketch), and outputs the difference $\Delta S \in \psi$.

Correctness. We say a linear sketch scheme Σ_{LS} for a fuzzy key setting $\mathcal{K}$ and Λ is correct if, for all $x, x' \in X$ such that $x' \in \xi(x)$ and all $pp_{LS} \leftarrow \mathsf{LS.Setup}(\mathcal{K}, \Lambda)$, if $(c, S) \leftarrow \mathsf{LS.Sketch}(pp_{LS}, x)$, and $(c', S') \leftarrow \mathsf{LS.Sketch}(pp_{LS}, x')$, then we have that $S' - S = \mathsf{LS.DiffRec}(pp_{LS}, c, c')$.

Linearity. We say a linear sketch scheme Σ_{LS} satisfies linearity if there exists a DPT algorithm M_c satisfying the following: for all $x \in X, e \leftarrow \Phi$, and $pp_{LS} \leftarrow \mathsf{LS.Setup}(\mathcal{K}, \Lambda)$, if $(c, S) \leftarrow \mathsf{LS.Sketch}(pp_{LS}, x)$ and $(c', \Delta S) \leftarrow \mathsf{M}_c(pp_{LS}, c, e)$, then we get that $\mathsf{LS.Sketch}(pp_{LS}, x + e) = (c', S + \Delta S)$.

C. FUZZY SIGNATURES

We now give the formal definition of fuzzy signatures [1], [7], [13], whose messages are signed by the fuzzy data x', and the corresponding signatures can be verified by public key pk generated by another fuzzy data x, where $x' \in \xi(x)$. More specifically, the secret key will not be explicitly defined in the scheme, since the fuzzy data x can be regarded as the same role of the secret key in the fuzzy signature scheme.

Definition 7 (Fuzzy Signatures): Let Σ_{FS} be a fuzzy signature scheme for a fuzzy key setting $\mathcal{K} = (X, \mathcal{X}, \xi, \Phi, \epsilon)$ with massage space $\mathcal{M}$ consisting of four algorithms.

- $\mathsf{FS.Setup}(1^\kappa, \mathcal{K}) \to pp_{FS}$: The setup algorithm takes the security parameter 1^κ and the fuzzy key setting $\mathcal{K}$ as inputs and outputs a public parameter pp_{FS}.

[3] FMR is the rate at which a biometric process mismatches biometric signals from two distinct individuals as coming from the same individual.

[4] FNMR is the rate at which a biometric matcher miscategorizes two captures from the same individual as being from different individuals.

- $\mathsf{FS.KGen}(pp_{FS}, x) \rightarrow pk_{FS}$: The key generation algorithm takes pp_{FS} and a fuzzy data $x \in X$ as inputs, and outputs a public key pk_{FS}.
- $\mathsf{FS.Sign}(pp_{FS}, x', M) \rightarrow \sigma_{FS}$: The signing algorithm takes pk_{FS}, a fuzzy data $x' \in X$, and a message $M \in \mathcal{M}$ as inputs, and outputs a signature σ_{FS}.
- $\mathsf{FS.Vrfy}(pp_{FS}, pk_{FS}, M, \sigma_{FS}) \rightarrow 0/1$: The verification algorithm takes pp_{FS}, pk_{FS}, and the message-signature pair (M, σ_{FS}) as inputs, output 1 (resp. 0) indicates that σ_{FS} is a valid (resp. invalid) signature of the message M under the public key pk_{FS}.

We recall the correctness and EUF-CMA security of fuzzy signatures [7]. Briefly, the correctness requires that a signature signed by a fuzzy data $x' \in \xi(x)$ can be verified by a public key pk_{FS} generated by the fuzzy data x, and parameter ϵ is connected to the probability

$$\Pr[x \leftarrow \mathcal{X}, e \leftarrow \Phi : x + e \in \xi(x)] \geq 1 - \epsilon.$$

Formally, the work [7] defined ϵ-correctness and EUF-CMA security of fuzzy signatures. A fuzzy signature scheme Σ_{FS} for a fuzzy key setting $\mathcal{K}$ is ϵ-correct if, for all $M \in \mathcal{M}$, $x \leftarrow X$, and $e \leftarrow \Phi$, the following holds

$$\Pr[\mathsf{FS.Vrfy}(pp_{FS}, pk_{FS}, M, \sigma_{FS}) = 1] \geqslant 1 - \epsilon,$$

where the probability is taken over the randomness of algorithms $pp_{FS} \leftarrow \mathsf{FS.Setup}(1^\kappa, \mathcal{K})$, $pk_{FS} \leftarrow \mathsf{FS.KGen}(pp_{FS}, x)$, and $\sigma_{FS} \leftarrow \mathsf{FS.Sign}(pp_{FS}, x + e, M)$.

EUF-CMA security of fuzzy signatures is similar to those of standard signatures except that the challenger uses $x' \in \xi(x)$ to respond signing queries rather than the original x used to generate the public key pk_{FS}. The security model of a fuzzy signature scheme Σ_{FS} for a fuzzy key setting $\mathcal{K}$ is defined by the following game, which is between a challenger $\mathcal{C}$ and a forger $\mathcal{F}$:

- **Setup:** The challenger $\mathcal{C}$ first runs $pp_{FS} \leftarrow \mathsf{FS.Setup}(1^\kappa, \mathcal{K})$, $x \leftarrow X$, and $pk_{FS} \leftarrow \mathsf{FS.KGen}(pp_{FS}, x)$, and then sends the public parameter pp_{FS}, and public key pk_{FS} to the forger $\mathcal{F}$.
- **Signing:** The forger $\mathcal{F}$ is allowed to query messages adaptively. When $\mathcal{F}$ asks the signature on any fresh message M, the challenger $\mathcal{C}$ randomly samples $e \leftarrow \Phi$ and computes $\sigma_M \leftarrow \mathsf{FS.Sign}(pp_{FS}, x + e, M)$. Then $\mathcal{C}$ sends the signature σ_M to $\mathcal{F}$. The forger can repeat this queries in polynomial time.
- **Forge:** Finally, the forger $\mathcal{F}$ outputs a message-signature pair (M^*, σ_{M^*}), and let Q be the set of all messages queried by $\mathcal{F}$. The challenger $\mathcal{C}$ outputs 1 if

$$M^* \notin Q \wedge \mathsf{FS.Vrfy}(pp_{FS}, pk_{FS}, M^*, \sigma_{M^*}) = 1.$$

Otherwise, it outputs 0.

If the challenger $\mathcal{C}$ outputs 1, we say that the forger $\mathcal{F}$ wins this game. The advantage of the forger $\mathcal{F}$ in the game is defined as

$$\mathsf{Adv}^{euf-cma}_{\Sigma_{\mathsf{FS}}, \mathcal{F}}(1^\kappa) = \Pr[\mathcal{C} \text{ outputs } 1].$$

IV. A FUZZY SIGNATURE SCHEME FROM LATTICE

In this section, we first provide an instantiation of linear sketch, which is modified from an existing linear sketch in [7]. Then we introduce our concrete fuzzy signature from lattice, and finally give a rigorous proof of our scheme.

A. THE MODIFIED LINEAR SKETCH

Katsumata et al. [7] utilized a mathematical object called lattice [24] to construct their specific linear sketch. Lattice has the property of discretization and linearity, which can properly represent the fuzzy data, and associate fuzzy data and cryptographic operations together.

In the following, we first recall the basic definition of lattice, and declare several primitives related to lattice. Then we give our mapping h which is used to replace the universal hash function in the linear sketch proposed by [7]. Furthermore, we give the detail explanation for this modification without compromising the functionality and structure of the original linear sketch in [7]. We finally provide the modified linear sketch scheme.

Let $m \in \mathbb{N}$ and a lattice $\mathcal{L} = \mathcal{L}(\mathbf{B}) = \{\mathbf{Bz} : \mathbf{z} \in \mathbb{Z}^m\}$ is the set of all integer linear combinations of basis $\mathbf{B} \in \mathbb{R}^{m \times m}$. For a vector $\mathbf{x} \in \mathbb{R}^m$, the closest lattice point of $\mathbf{x}$ in lattice $\mathcal{L}$ is a vector $\mathbf{y} \in \mathcal{L}$, denoted by

$$\mathsf{CV}_{\mathcal{L}}(\mathbf{x}) := \{\mathbf{y} : \|\mathbf{x} - \mathbf{y}\| \leqslant \|\mathbf{x} - \mathbf{Bz}\|\}$$

for any $\mathbf{z} \in \mathbb{Z}^m$. The Voronoi region of $\mathbf{y} \in \mathcal{L}$, denoted by $\mathsf{VR}_{\mathcal{L}}(\mathbf{y})$, is defined by

$$\mathsf{VR}_{\mathcal{L}}(\mathbf{y}) = \{\mathbf{x} : \mathbf{y} = \mathsf{CV}_{\mathcal{L}}(\mathbf{x})\}.$$

In addition, since the symmetry of lattices, we have

$$\mathsf{VR}_{\mathcal{L}}(\mathbf{y}) = \mathsf{VR}_{\mathcal{L}}(\mathbf{0}) + \mathbf{y}.$$

Let $g_{\mathcal{L}} : X \rightarrow \mathcal{L}$ be the function $g_{\mathcal{L}}(\mathbf{x}) = \mathbf{B}\lfloor \mathbf{B}^{-1}\mathbf{x} \rfloor$.

1) THE MAPPING h

Let vector $\mathbf{v} \leftarrow \mathbb{Z}^k$ where $k \in \mathbb{N}$, and $h : \mathbb{Z}_q^m \rightarrow \{-d, \cdots, 0, \cdots, d\}^{m \times k}$ be a mapping satisfying linearity (under the modulo $(2d + 1)$). More precisely, let

$$h(\mathbf{y}) = h_{(\mathbf{B}, \mathbf{v})}(\mathbf{y}) = \mathbf{B}^{-1}\mathbf{y}\mathbf{v}^\top \mod (2d + 1).$$

Hence we have that

$$h_{(\mathbf{B}, \mathbf{v})}(\mathbf{y}_1 + \mathbf{y}_2) = (h_{(\mathbf{B}, \mathbf{v})}(\mathbf{y}_1) + h_{(\mathbf{B}, \mathbf{v})}(\mathbf{y}_2)) \mod (2d + 1).$$

The above explains that linearity property of the function h is under modulo $(2d + 1)$, which leads to linearity of the modified linear sketch also under the same situation.

We use the mapping h to replace the universal hash function used in the linear sketch scheme proposed by [7], since this modification made on the mapping h not only makes the mapping h achieve the properties of pre-image resistance and collision resistance like the universal hash function in [7], but also makes the modified linear sketch capable of being adapted into the lattice-based setting. Hence, the modified linear sketch can benefit from the original construction of [7],

while being applicable to the lattice-based setting. The analysis of the mapping h satisfying the properties of pre-image resistance and collision resistance is in the following.

For the function $h_{(\mathbf{B},\mathbf{v})}(\mathbf{y}) = \mathbf{B}^{-1}\mathbf{y}v^{\top} \mod (2d+1)$, we assume that $\mathbf{B}^{-1}\mathbf{y} = (x_1, \cdots, x_m)^{\top}$, $v^{\top} = (v_1, \cdots, v_k)$. Let $h_{(\mathbf{B},\mathbf{v})}(\mathbf{y}) = Z = (z_{ij})^{m\times k}$ where $i \in \{1, 2, \cdots, m\}$, and $j \in \{1, 2, \cdots, k\}$. Thereby, we have that

$$z_{ij} = x_i v_j \mod (2d+1).$$

Since the hash value Z and $v^{\top}$ is public, there is a chance of $\lfloor \frac{q}{2d+1} \rceil / q$ getting the value of x_i (i.e., $\mathbf{B}^{-1}\mathbf{y}$). After that, we can easily get $\mathbf{y}$ by having a left multiplication on $\mathbf{B}^{-1}\mathbf{y}$ by matrix $\mathbf{B}$. Hence, with a probability of at most $\frac{1}{(2d+1)^m}$, one can reveal vector $\mathbf{y}$ by knowing the hash value $\mathbf{Z}$ such that

$$\mathbf{Z} = h_{(\mathbf{B},\mathbf{v})}(\mathbf{y}) = \mathbf{B}^{-1}\mathbf{y}\mathbf{v}^{\top} \mod (2d+1).$$

The parameters of our scheme are inherited from [10], so the probability of $\frac{1}{(2d+1)^m}$ is small enough.

We now analyze the probability of getting a collision pair of function h. Assume that there exist two different vectors $\mathbf{y}, \mathbf{y}'$ such that $h_{(\mathbf{B},\mathbf{v})}(\mathbf{y}) = h_{(\mathbf{B},\mathbf{v})}(\mathbf{y}')$. Rearranging the equation, we can get that

$$\begin{aligned} &\mathbf{B}^{-1}\mathbf{y} = \mathbf{B}^{-1}\mathbf{y}' && \mod (2d+1) \\ \Longrightarrow\ &\mathbf{B}^{-1}(\mathbf{y}-\mathbf{y}') = 0 && \mod (2d+1). \end{aligned}$$

Since $|\mathbf{B}^{-1}| \neq 0$, we obtain $\mathbf{y} = \mathbf{y}' \mod (2d+1)$. This shows that if $\mathbf{y}' = \mathbf{y} + \mathbf{k}(2d+1)$ for $\mathbf{k} \in \mathbb{Z}^m$, there exists a pair of collision $\mathbf{y}$ and $\mathbf{y}'$ such that $h_{(\mathbf{B},\mathbf{v})}(\mathbf{y}) = h_{(\mathbf{B},\mathbf{v})}(\mathbf{y}')$. However, we need to consider its specific application scenario. Since the mapping h is used in the algorithm $\mathsf{Sketch}(pp_{LS}, \mathbf{x})$ to produce the proxy key $\mathbf{S}$, and the input of h is actually kept private. Hence, if fixing a vector $\mathbf{y}$ (unknown to others), there is a chance of at most $\frac{1}{(2d+1)^m}$ to obtain the vector $\mathbf{y}'$ by guess such that $h_{(\mathbf{B},\mathbf{v})}(\mathbf{y}') = h_{(\mathbf{B},\mathbf{v})}(\mathbf{y})$. Thus, it is hard to find a collision pair of function h under this circumstance.

2) THE MODIFIED LINEAR SKETCH

Let $\mathcal{K} = (X, \mathcal{X}, \xi, \Phi, \epsilon)$ be a concrete fuzzy key setting with respect to a lattice $\mathcal{L}$, where $X = \mathbb{R}^m$, $\mathcal{X}$ has the property that if $\mathbf{x} \leftarrow \mathcal{X}$, then $\mathbf{B}^{-1}\mathbf{x} \in [0, q)^m$, the acceptance region function of vector $\mathbf{x}$ is $\xi(\mathbf{x}) = \xi_{\mathcal{L}}(\mathbf{x}) = \{\mathbf{x}' : \mathsf{CV}_{\mathcal{L}}(\mathbf{x}-\mathbf{x}') = 0\}$, and Φ is any efficiently samplable distribution over X such that $\mathsf{FNMR} \leq \epsilon$.

Combining all the above building blocks, the detail of the modified linear sketch Σ_{LS} is described as the Fig. 1. The auxiliary algorithm M_c in Fig. 1 is used for proving the linearity property of Σ_{LS}.

Since the proof process of correctness and linearity of Σ_{LS} in the Fig. 1 are similar to the proof in [7], we briefly introduce the proof process. Linearity of our mapping is under modulo $(2d+1)$, hence correctness and linearity of Σ_{LS} are also under the same situation.

Proof of correctness. For correctness, we need to prove that $\Delta\mathbf{S} = (\mathbf{S}' - \mathbf{S}) \mod 2d+1$ where $\Delta\mathbf{S} = \mathsf{LS.DiffRec}(pp_{LS}, \mathbf{c}, \mathbf{c}')$, $(\mathbf{c}, \mathbf{S}) \leftarrow \mathsf{LS.Sketch}(pp_{LS}, \mathbf{x})$, and $(\mathbf{c}', \mathbf{S}') \leftarrow \mathsf{LS.Sketch}(pp_{LS}, \mathbf{x}')$ for $\mathbf{x}, \mathbf{x}' \in X$ satisfying $\mathbf{x}' \in \xi(\mathbf{x})$, and $pp_{LS} \leftarrow \mathsf{LS.Setup}(\mathcal{K}, \Lambda)$. In the algorithm $\mathsf{LS.DiffRec}$ of Σ_{LS}, $\Delta y \leftarrow \mathsf{CV}_{\mathcal{L}}(\mathbf{c}-\mathbf{c}')$ can be written as

$$\begin{aligned} \Delta y &= \mathsf{CV}_{\mathcal{L}}(\mathbf{c}-\mathbf{c}') = \mathsf{CV}_{\mathcal{L}}((\mathbf{x}-\mathbf{y}) - (\mathbf{x}'-\mathbf{y}')) \\ &= \mathsf{CV}_{\mathcal{L}}(\mathbf{x}-\mathbf{x}') + \mathbf{y}' - \mathbf{y} = \mathbf{y}' - \mathbf{y} \end{aligned}$$

where $\mathsf{CV}_{\mathcal{L}}(\mathbf{x}-\mathbf{x}') = 0$ since $\mathbf{x}' \in \xi(\mathbf{x})$. Hence we have

$$\begin{aligned} \Delta\mathbf{S} &= h_{(\mathbf{B},\mathbf{v})}(\Delta\mathbf{y}) = h_{(\mathbf{B},\mathbf{v})}(\mathbf{y}' - \mathbf{y}) \\ &= (\mathbf{S}' - \mathbf{S}) \mod 2d+1. \end{aligned}$$

Proof of linearity. For linearity, we use the auxiliary algorithm M_c to prove it. Let

$$(\mathbf{c}, \mathbf{S}) = (\mathbf{x} - g_{\mathcal{L}}(\mathbf{x}), h_{(\mathbf{B},\mathbf{v})}(g_{\mathcal{L}}(\mathbf{x}))) = \mathsf{LS.Sketch}(pp_{LS}, \mathbf{x}),$$

and

$$\begin{aligned} (\mathbf{c}', \mathbf{S}') &= ((\mathbf{x}+\mathbf{e}) - g_{\mathcal{L}}(\mathbf{x}+\mathbf{e}), h_{(\mathbf{B},\mathbf{v})}(g_{\mathcal{L}}(\mathbf{x}+\mathbf{e}))) \\ &= \mathsf{LS.Sketch}(pp_{LS}, \mathbf{x}+\mathbf{e}). \end{aligned}$$

Enlightened by [7], we get that the following equation:

$$\begin{aligned} &(\mathbf{c}+\mathbf{e} - g_{\mathcal{L}}(\mathbf{c}+\mathbf{e}), h_{(\mathbf{B},\mathbf{v})}(g_{\mathcal{L}}(\mathbf{c}+\mathbf{e}))) \\ &= (\mathbf{c}', (\mathbf{S}' - \mathbf{S}) \mod 2d+1) \end{aligned} \tag{1}$$

where the first item of the above (1) is the output of $\mathsf{M}_c(pp_{LS}, \mathbf{c}, \mathbf{e})$ is sufficient to show linearity of Σ_{LS}. By applying the related elements of Σ_{LS} to the first item of the above (1), it is easy to get the equality. Here we complete the proof.

We now illustrate why sketch $\mathbf{c}$ of Σ_{LS} does not leak the information of fuzzy data $\mathbf{x}$ and the proxy key $\mathbf{S}$. To ensure the privacy of fuzzy data $\mathbf{x}$, we require one quantity, the conditional false matching rate (ConFMR) as [7], to guarantee it. The definition of ConFMR is as follows:

$$\mathsf{ConFMR} = \Pr\left[\begin{array}{l|l} \mathbf{x}, \mathbf{x}' \leftarrow \mathcal{X} & \\ \mathbf{c} \leftarrow \mathbf{x} - g_{\mathcal{L}}(\mathbf{x}) & : \mathbf{x}' \in \xi(\mathbf{x}) \\ \mathbf{c}' \leftarrow \mathbf{x}' - g_{\mathcal{L}}(\mathbf{x}') & \end{array} \middle| \ \mathbf{c} = \mathbf{c}' \right].$$

Here we require $\mathsf{ConFMR} \approx 2^{-\kappa}$ is small, which indicates that with a low probability $2^{-\kappa}$, one can get a "collision" pair of $(\mathbf{x}, \mathbf{c})$, $(\mathbf{x}', \mathbf{c}')$ such that $\mathbf{c} = \mathbf{c}'$ and $\mathbf{x}' \in \xi(\mathbf{x})$. For the proxy key $\mathbf{S}$, since the randomness of the vector $\mathbf{v}$, the value of $\mathbf{S} \leftarrow h_{(\mathbf{B},\mathbf{v})}(\mathbf{y})$ is statistically close to an uniformly random element even given $\mathbf{c}$. Refer to [7] for more detail.

B. OUR SCHEME

In this section, we propose our concrete lattice-based fuzzy signature scheme. We first give the parameter setting. Most parameters in our scheme are inherited from the scheme in [10]. Let $\kappa \in \mathbb{N}$ be the security parameter, and q be prime integer. Let $m, n, k, d \in \mathbb{Z}^+$ and small $\eta > 1$. Let $M = O(1)$, and $\sigma \in \mathbb{R}^+$. Let hash function $H : \{0,1\}^* \to \{-1, 0, 1\}^k$, and matrix $\mathbf{E} \in \mathbb{Z}^{m\times k}$ where all entities are small integers.

$\mathsf{LS.Setup}(\mathcal{K}, \Lambda = (\mathbb{Z}_{2d+1}^{m\times k}, +))$	$\mathsf{LS.Sketch}(pp_{LS}, \mathbf{x})$	$\mathsf{LS.DiffRec}(pp_{LS}, \mathbf{c}, \mathbf{c}')$	$\mathsf{M}_c(pp_{LS}, \mathbf{c}, \mathbf{e})$
1 : The mapping $h_{(\mathbf{B},\mathbf{v})}$;	1 : $\mathbf{y} \leftarrow g_{\mathcal{L}}(\mathbf{x})$;	1 : $\Delta\mathbf{y} \leftarrow \mathsf{CV}_{\mathcal{L}}(\mathbf{c} - \mathbf{c}')$;	1 : $\mathbf{c}' \leftarrow \mathbf{c} + \mathbf{e} - g_{\mathcal{L}}(\mathbf{c} + \mathbf{e})$;
2 : Return $pp_{LS} = (\Lambda, h_{(\mathbf{B},\mathbf{v})})$.	2 : $\mathbf{c} \leftarrow \mathbf{x} - \mathbf{y}$;	2 : $\Delta\mathbf{S} \leftarrow h_{(\mathbf{B},\mathbf{v})}(\Delta\mathbf{y})$;	2 : $\hat{\mathbf{y}} \leftarrow g_{\mathcal{L}}(\mathbf{c} + \mathbf{e})$;
	3 : $\mathbf{S} \leftarrow h_{(\mathbf{B},\mathbf{v})}(\mathbf{y})$;	3 : Return $\Delta\mathbf{S}$.	3 : $\Delta\mathbf{S} \leftarrow h_{(\mathbf{B},\mathbf{v})}(\hat{\mathbf{y}})$;
	4 : Return $(\mathbf{c}, \mathbf{S})$.		4 : Return $(\mathbf{c}', \Delta\mathbf{S})$.

FIGURE 1. The modified linear sketch.

The modified linear sketch Σ_{LS} utilized in our lattice-based fuzzy signature scheme Σ_{FS} is given in the Section IV-A. Please see our concrete scheme Σ_{FS} in Fig. 2.

ϵ-Correctness. From the definition of the fuzzy key setting $\mathcal{K}$, we have

$$\Pr[\mathbf{x} \leftarrow \mathcal{X}, \mathbf{e} \leftarrow \Phi : \mathbf{x} + \mathbf{e} \in \xi(\mathbf{x})] \geq 1 - \epsilon.$$

Hence, to show correctness of our scheme, it is sufficient to shows that if $\mathbf{x}' \in \xi(\mathbf{x})$, then a signature generated by $\mathbf{x}'$ can be accepted under a public key pk_{FS} generated by $\mathbf{x}$.

We now consider the execution of the verification algorithm. From the scheme, we have that the actual distribution of $\mathbf{z}$ is $D^m_{\mathbf{S}'\mathbf{h},\sigma}$. From Theorem 1, we get that the actual distribution of $\mathbf{z}$ is statistically close to the distribution in which $\mathbf{z}$ is chosen from D^m_σ. Hence we tailored $\mathbf{z}$ to be distributed according to D^m_σ. By Lemma 1, we have that $\|\mathbf{z}\| \leq \eta\sigma\sqrt{m}$ with probability at least $1 - 2^{-m}$. Moreover, from Lemma 3, we get that finding a $\mathbf{y}' \in D^m_\sigma$ such that $\mathbf{y}' = \mathbf{y}$ is at most with a probability of 2^{1-m}. Hence $\mathbf{Y} = \mathbf{A}\mathbf{y}$ being public does not compromise the security of our scheme.

From correctness of linear sketch, we get that

$$\mathsf{LS.DiffRec}(pp_{LS}, \mathbf{c}, \mathbf{c}') = \Delta\mathbf{S} = (\mathbf{S}' - \mathbf{S}) \mod (2d + 1),$$

which indicates that there must exist a matrix $\mathbf{E} \in \mathbb{Z}^{m\times k}$ such that

$$\mathbf{S}' = \mathbf{S} + \Delta\mathbf{S} + (2d + 1)\mathbf{E} \in \{-d, \cdots, d\}^{m\times k} \tag{2}$$

Hence the ephemeral public key $\mathbf{T}'$ can be written as

$$\mathbf{T}' = \mathbf{A}\mathbf{S}' = \mathbf{T} + \mathbf{A}(\Delta\mathbf{S} + (2d + 1)\mathbf{E}).$$

Furthermore, since $\mathbf{z} = \mathbf{S}'\mathbf{h} + \mathbf{y}$, we easily get that

$$\mathbf{A}\mathbf{z} = \mathbf{A}(\mathbf{S}'\mathbf{h} + \mathbf{y}) = \mathbf{T}'\mathbf{h} + \mathbf{Y} \tag{3}$$

where $\mathbf{h} = H(\mathbf{Y}, \mathbf{m})$. Hence, such matrix $\mathbf{E}$ satisfying the above (2) also meets the equation of (3). Therefore, the signature $\sigma_{FS} = (\mathbf{z}, \mathbf{Y}, \mathbf{c}')$ is accepted by the verification algorithm. Hence σ_{FS} is a valid signature of the message $\mathbf{m}$.

We now give the way to calculate matrix $\mathbf{E} \in \mathbb{Z}^{m\times k}$. The parameters of our scheme are inherited from the scheme in [10] which gave two instantiations of $d = 1$ and $d = 31$, respectively. Even though the concrete value of $\mathbf{S}$, $\mathbf{S}'$ are unknown, we still know that the range of them are from $\{-d, \cdots, 0, \cdots, d\}^{m\times k}$, and $\Delta\mathbf{S}$ is public. Thus, from the above (2), we can get that the possible values of elements of matrix $\mathbf{E}$ are $\{-1, 0, 1\}$.

More concretely, for $d = 1$, if one element of $\Delta\mathbf{S}$ is -1, the corresponding element of $\mathbf{E}$ has two possible values, 1 with probability of $\frac{1}{3}$, and 0 with probability of $\frac{2}{3}$; If one element of $\Delta\mathbf{S}$ is 0, the corresponding element of $\mathbf{E}$ is also 0; If one element of $\Delta\mathbf{S}$ is 1, the corresponding element of $\mathbf{E}$ has two possible values, -1 with probability of $\frac{1}{3}$, and 0 with probability of $\frac{2}{3}$. The randomness of choosing $\mathbf{S}$ is used to calculate the above probability of the values of $\mathbf{E}$. Then we start to calculate the matrix $\mathbf{E}$. First of all, we set all the elements of $\mathbf{E}$ are 0. Then we calculate the value of

$$\mathbf{T}'\mathbf{h} + \mathbf{Y} = (\mathbf{T} + \mathbf{A}(\Delta\mathbf{S} + (2d + 1)\mathbf{E}))\cdot\mathbf{h} + \mathbf{Y} \tag{4}$$

to compare with the value of $\mathbf{A}\mathbf{z}$. Since $\Delta\mathbf{S}$ is public, we can get the positions of the elements $(-1, 0, 1)$ of $\Delta\mathbf{S}$. Hence, for the unmatched elements between the comparison result of the value of $\mathbf{A}\mathbf{z}$, and the above (4), if the values of the corresponding positions of the unmatched elements in $\Delta\mathbf{S}$ is -1 (1), we change 0 to 1 (-1) in the corresponding positions of $\mathbf{E}$. Therefore, we can just calculate the above (4) once to get the value of $\mathbf{E}$. For $d = 31$, it also has a similar pattern.

C. SECURITY PROOF

Before giving the full security proof, we give a statement to illustrate that compared to [10], why our scheme can skip one signing hybrid (which is the Hybrid 2 in [10]) in the security proof process, since we use another approach to prove the security of our scheme.

Statement. Our fuzzy signature scheme Σ_{FS} is based on the scheme of [10], but we do not adopt the same method as [10] to prove the security of our scheme.

For the scheme in [10], there are two possible methods of responding the signing queries of the forger during signing, either using the simulated secret key to generate signatures (**method A**), or programming the random oracle accordingly and generating $\mathbf{z}$ directly from the distribution D^m_σ without utilizing the secret key (**method B**). More precisely, **method A** can just utilize Hybrid 1 of [10] (which is using a randomly chosen element to replace the actual hash value), and another one, **method B**, employs two hybrids shown in [10]. These two hybrids of [10] both obtain the property of the distribution of signatures generated by these two hybrids independent of the secret key. For Hybrid 1, it utilizes the rejection sampling to approach it as the actual signing algorithm, and signatures

FS.Setup$(1^\kappa, \mathcal{K})$

1 : $pp_{LS} \leftarrow \mathsf{LS.Setup}(\mathcal{K}, \Lambda)$;
2 : $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$;
3 : Return $pp_{FS} = (pp_{LS}, \mathbf{A})$.

FS.KGen$(pp_{FS}, \mathbf{x})$

1 : $(pp_{LS}, \mathbf{A}) \leftarrow pp_{FS}$;
2 : $(\mathbf{c}, \mathbf{S}) \leftarrow \mathsf{LS.Sketch}(pp_{LS}, \mathbf{x})$;
3 : $\mathbf{T} = \mathbf{AS}$;
4 : Return $pk_{FS} = (\mathbf{T}, \mathbf{c})$.

FS.Sign$(pp_{FS}, \mathbf{x}', \mathbf{m})$

1 : $(pp_{LS}, \mathbf{A}) \leftarrow pp_{FS}$;
2 : $(\mathbf{c}', \mathbf{S}') \leftarrow \mathsf{LS.Sketch}(pp_{LS}, \mathbf{x}')$;
3 : $\mathbf{y} \leftarrow D_\sigma^m$;
4 : Let $\mathbf{Y} = \mathbf{Ay}$;
5 : $\mathbf{h} \leftarrow H(\mathbf{Y}, \mathbf{m})$;
6 : $\mathbf{z} \leftarrow \mathbf{S}'\mathbf{h} + \mathbf{y}$;
7 : Return $\sigma_{FS} = (\mathbf{z}, \mathbf{Y}, \mathbf{c}')$ with probability P.
8 : Note that : $P = \min\left(\frac{D_\sigma^m(\mathbf{z})}{MD^m_{\mathbf{S}'\mathbf{h},\sigma}(\mathbf{z})}, 1\right)$

FS.Vrfy$(pp_{FS}, pk_{FS}, \mathbf{m}, \sigma_{FS})$

1 : $(pp_{LS}, \mathbf{A}) \leftarrow pp_{FS}$;
2 : $\Delta\mathbf{S} \leftarrow \mathsf{LS.DiffRec}(pp_{LS}, \mathbf{c}, \mathbf{c}')$;
3 : $\mathbf{T}' = \mathbf{T} + \mathbf{A} \cdot (\Delta\mathbf{S} + (2d+1)\mathbf{E})$;
4 : Compute $\mathbf{h} \leftarrow H(\mathbf{Y}, \mathbf{m})$;
5 : **if** $\|\mathbf{z}\| \le \eta\sigma\sqrt{m} \wedge \exists$ *an* $\mathbf{E}$ *such that* $\mathbf{Az} = \mathbf{T}'\mathbf{h} + \mathbf{Y}$
6 : Return 1;
7 : **else**
8 : Return 0.

FIGURE 2. Our fuzzy signature scheme from lattice.

generated by Hybrid 2 are randomly chosen without employing the secret key. Moreover, by appropriately choosing the parameters of σ and M in the scheme of [10], the statistical distance between the distribution of signatures from Hybrid 1 and from Hybrid 2 in [10] is small which indicates the difference caused by signatures generated from these two hybrids are slight.

Moveover, the scheme of [10] can actually use the above two methods in its proof. The author in [10] chose **method B** because he would like to be able to still use the same lemma of security proof with another section which proposed another schemes that can just use **method B** to proof their schemes.

As for our scheme, we prefer to use **method A** since the "ephemeral" public key $\mathbf{T}'$, which is utilized in our verification algorithm, cannot be directly obtained like the public key $\mathbf{T}$ in the proof of [10]. More specifically, the linearity of our mapping h is satisfied under modulo $2d+1$, so the "ephemeral" public key $\mathbf{T}'$ cannot be derived just by employing the public key and the signature. In the signing queries, we simulate the secret key to generate signatures which can be indistinguishable from the actual one. That is why we skip Hybrid 2 of [10], since we manage to simulate the signing key to generate signatures in the signing queries phase by using **method A** which can just employ Hybrid 1 of [10].

In addition, we use the auxiliary algorithm M_c in the Hybrid 1 to get the sketch $\mathbf{c}'$ without knowing the knowledge of the fuzzy data x'. Then we construct an algorithm solving the $\mathsf{SIS}_{q,m,n,\beta}$ assumption by simulating adversary against EUF-CMA security game by running algorithm M_c. We now prove the security of our signature scheme as follows.

Theorem 2: If there is a polynomial-time forger who makes at most s queries to the signing oracle and g queries to the random oracle H, and breaking the EUF-CMA security with probability δ, then there exists a polynomial-time algorithm solving the $\mathsf{SIS}_{q,m,n,\beta}$ problem for $\beta = (2\eta\sigma + 2dk)\sqrt{m}$ with probability $\approx \frac{\delta^2}{2(g+s)}$.

This theorem is proved in a sequence of two lemmas. In the Lemma 5, we illustrate that the actual signing algorithm can be replaced with Hybrid 2, and the statistical distance between these two outputs is at most $\epsilon' = s(s+g)2^{-n+1}$. For the Lemma 6, we assume that a forger produces a forgery with probability δ when the signing algorithm is replaced with Hybrid 2. Then we can use it to recover $\mathbf{v}$ such that $\|\mathbf{v}\| \le (2\eta\sigma + 2dk)\sqrt{m}$ and $\mathbf{Av} = 0$ with probability at least $\frac{\delta^2}{2(g+s)}$. Please see two signing hybrids in Fig. 3.

Lemma 5: Let $\mathcal{D}$ be a distinguisher which can query the random oracle H and either the actual signing algorithm or Hybrid 2. If he can make g queries to random oracle H and s queries to the signing algorithm that he can access to, then for all but a $e^{-\Omega(n)}$ fraction of all possible matrices $\mathbf{A}$, the advantage of the distinguisher $\mathcal{D}$ in distinguishing the actual signing algorithm from the one in Hybrid 2 is at most $s(s+g)2^{-n+1}$.

Proof: First, we show the outputs of the actual signing algorithm and Hybrid 1 exactly follow the same distribution. Instead of directly using the Sketch algorithm of Σ_{LS} again with input of fuzzy data x' to generate the sketch $\mathbf{c}'$ in the actual signing algorithm, we use the auxiliary algorithm M_c of Σ_{LS} with inputs of $\mathbf{c}$ and $\mathbf{e}$. Since the linearity of Σ_{LS}, the distribution of $\mathbf{c}'$ generated in the actual signing algorithm and in the Hybrid 1 are identical.

We then declare that the distinguisher $\mathcal{D}$ has the advantage of at most $s(s+g)2^{-n+1}$ to distinguish an output of Hybrid 1 from an output of Hybrid 2. The only difference between these two Hybrids is the output of the random oracle H. In Hybrid 2, the outputs of H are randomly chosen from $\{-1, 0, 1\}^k$ and then programmed as the response of $H(\mathbf{Y}, \mathbf{m}) = H(\mathbf{Az} - \mathbf{T}'\mathbf{h}, \mathbf{m}) = \mathbf{h}$ without checking $(\mathbf{Y}, \mathbf{m})$ is set or not. For each time the Hybrid 2 is called, the probability of getting a vector $\mathbf{y}$ such that $\mathbf{Ay}$ is equal to the one queried before is at most 2^{-n+1}. By [10], we know that with

Hybrid 1	Hybrid 2
$\mathbf{e} \leftarrow \Phi$;	$\mathbf{e} \leftarrow \Phi$;
$(\mathbf{c}', \Delta\mathbf{S}) \leftarrow \mathsf{M}_c(pp_{LS}, \mathbf{c}, \mathbf{e})$;	$(\mathbf{c}', \Delta\mathbf{S}) \leftarrow \mathsf{M}_c(pp_{LS}, \mathbf{c}, \mathbf{e})$;
$\mathbf{S}' = \mathbf{S} + \Delta\mathbf{S} + (2d+1)\mathbf{E} \in \{-d, \cdots, 0, \cdots, d\}^{m\times k}$;	$\mathbf{S}' = \mathbf{S} + \Delta\mathbf{S} + (2d+1)\mathbf{E} \in \{-d, \cdots, 0, \cdots, d\}^{m\times k}$;
$\mathbf{T}' = \mathbf{A}\mathbf{S}'$;	$\mathbf{T}' = \mathbf{A}\mathbf{S}'$;
$\mathbf{y} \leftarrow D_\sigma^m$;	$\mathbf{y} \leftarrow D_\sigma^m$;
Let $\mathbf{Y} = \mathbf{A}\mathbf{y}$;	Let $\mathbf{Y} = \mathbf{A}\mathbf{y}$;
$\mathbf{h} \leftarrow H(\mathbf{Y}, \mathbf{m})$;	$\mathbf{h} \leftarrow \{-1, 0, 1\}^k$;
$\mathbf{z} \leftarrow \mathbf{S}'\mathbf{h} + \mathbf{y}$;	$\mathbf{z} \leftarrow \mathbf{S}'\mathbf{h} + \mathbf{y}$;
Output $(\mathbf{z}, \mathbf{Y}, \mathbf{c}')$ with probability $\min\left(\frac{D_\sigma^m(\mathbf{z})}{MD^m_{\mathbf{S}'\mathbf{h},\sigma}(\mathbf{z})}, 1\right)$;	Output $(\mathbf{z}, \mathbf{Y}, \mathbf{c}')$ with probability $\min\left(\frac{D_\sigma^m(\mathbf{z})}{MD^m_{\mathbf{S}'\mathbf{h},\sigma}(\mathbf{z})}, 1\right)$;
Program $H(\mathbf{Y}, \mathbf{m}) = H(\mathbf{A}\mathbf{z} - \mathbf{T}'\mathbf{h}, \mathbf{m}) = \mathbf{h}$.	Program $H(\mathbf{Y}, \mathbf{m}) = H(\mathbf{A}\mathbf{z} - \mathbf{T}'\mathbf{h}, \mathbf{m}) = \mathbf{h}$.

FIGURE 3. Signing hybrids.

probability at least $1 - e^{-\Omega(n)}$, the matrix $\mathbf{A}$ can be written in "Herimte Normal Form" as $\mathbf{A} = [\bar{\mathbf{A}}\|\mathbf{I}]$. Then, for any $\mathbf{t} \in \mathbb{Z}_q^n$,

$$\begin{aligned}\Pr[\mathbf{A}\mathbf{y} = \mathbf{t} : \mathbf{y} \in D_\sigma^m] &= \Pr[\mathbf{y}_1 = (\mathbf{t} - \bar{\mathbf{A}}\mathbf{y}_0) : \mathbf{y} \in D_\sigma^m] \\ &\le \max_{\mathbf{t}' \in \mathbb{Z}_q^n} \Pr[\mathbf{y}_1 = \mathbf{t}' : \mathbf{y}_1 \leftarrow D_\sigma^n] \\ &\le 2^{-n+1}\end{aligned}$$

where $\mathbf{y} = [\mathbf{y}_0\|\mathbf{y}_1]^\top$. Since $\mathcal{D}$ can call random oracle H g times and the signing algorithm s times, there is at most $s + g$ values of $(\mathbf{Y}, \mathbf{m})$ set. Thus, for each time Hybrid 2 accessed, the probability of getting a collision is at most $(s+g)2^{-n+1}$. Therefore, the probability that a collision occurs after s queries from Hybrid 2 is at most $s(s+g)2^{-n+1}$. Hence, the statistical distance between the output of the actual signing algorithm and Hybrid 2 is at most $s(s+g)2^{-n+1}$. □

Lemma 6: Suppose that there exists a polynomial-time forger $\mathcal{F}$ who makes at most s queries to the signer in Hybrid 2, g queries to the random oracle H, and succeeds in forging with probability δ. Then there exists a polynomial-time algorithm $\mathcal{B}$ that for a given $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, finds a non-zero vector $\mathbf{v} \in \mathbb{Z}^m$ such that $\|\mathbf{v}\| \le (2\eta\sigma + 2dk)\sqrt{m}$ and $\mathbf{A}\mathbf{v} = 0$ with probability at least $\frac{\delta^2}{2(g+s)}$.

Proof: We now give the construction of algorithm $\mathcal{B}$, which simulates the attack environment for $\mathcal{F}$, and solves solving $\mathsf{SIS}_{q,m,n,\beta}$ assumption with probability at least $\frac{\delta^2}{2(g+s)}$. The algorithm $\mathcal{B}$ receives a challenge instance $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, and computes $pp_{LS} \leftarrow \mathsf{LS.Setup}(\mathcal{K}, \Lambda)$. $\mathcal{B}$ randomly chooses $\mathbf{x} \in X$, computes $\mathbf{c} = \mathbf{x} - g_{\mathcal{L}}(\mathbf{x})$, and $\mathbf{S} = h_{(\mathbf{B},\mathbf{v})}(g_{\mathcal{L}}(\mathbf{x}))$ where the function $g_{\mathcal{L}}$ and $h_{(\mathbf{B},\mathbf{v})}$ are defined in IV-A. Then algorithm $\mathcal{B}$ computes $\mathbf{T} = \mathbf{A}\mathbf{S}$. Let $(pp_{LS}, \mathbf{A}, \mathbf{T}, \mathbf{c})$ be public and keep $\mathbf{S}$ private.

When $\mathcal{F}$ asks to see a signature of certain message, $\mathcal{B}$ runs the signing algorithm of Hybrid 2 to produce a signature. Let $D_H = \{-1, 0, 1\}^k$ denote the range of the random oracle H, and let $t = g + s$ be the bound on the number of times the random oracle H is called or programmed during the attack from $\mathcal{F}$. The algorithm $\mathcal{B}$ will conduct as follows: $\mathcal{B}$ first picks up the values $\mathbf{r}_1, \cdots, \mathbf{r}_t \leftarrow D_H$ that will correspond to the responses of the random oracle H. Note that a random oracle query can be made by the forger $\mathcal{F}$ directly, or it can be programmed by the signing algorithm when the forger $\mathcal{F}$ makes some signing queries on some messages. Thus, during signing or when $\mathcal{F}$ makes queries to the random oracle, the random oracle H will be programmed by $\mathcal{B}$, and the response of H will be the first unused $\mathbf{r}_i$ in the list $(\mathbf{r}_1, \cdots, \mathbf{r}_t)$ every time. At the same time, $\mathcal{B}$ keeps a table of all queries to the random oracle H, so when the same query is made twice, the previously answered $\mathbf{r}_i$ will be replied. After making at most $s + g$ random oracle queries, $\mathcal{F}$ outputs a forged signature $(\hat{\mathbf{z}}, \hat{\mathbf{Y}}, \hat{\mathbf{c}}')$ on message $\hat{\mathbf{m}}$.

Recall that with probability δ, $\mathcal{F}$ will output a message $\hat{\mathbf{m}}$ with its corresponding signature $(\hat{\mathbf{z}}, \hat{\mathbf{Y}}, \hat{\mathbf{c}}')$ such that $\|\hat{\mathbf{z}}\| \le \eta\sigma\sqrt{m}$ and $\mathbf{A}\hat{\mathbf{z}} = \hat{\mathbf{T}}'\mathbf{h} + \hat{\mathbf{Y}}$ where $H(\hat{\mathbf{Y}}, \hat{\mathbf{m}}) = H(\mathbf{A}\hat{\mathbf{z}} - \hat{\mathbf{T}}'\hat{\mathbf{h}}, \hat{\mathbf{m}}) = \hat{\mathbf{h}}$, $\hat{\mathbf{T}}' = \mathbf{A}\hat{\mathbf{S}}'$, and

$$\hat{\mathbf{S}}' = \mathbf{S} + \Delta\hat{\mathbf{S}}' + (2d+1)\mathbf{E}_1 \in \{-d, \cdots, 0, \cdots, d\}^{m\times k}. \tag{5}$$

The above (5) shows the existence of the secret key $\hat{\mathbf{S}}'$, since the secret key $\mathbf{S}$, $\Delta\hat{\mathbf{S}}' \leftarrow \mathsf{LS.DiffRec}(pp_{LS}, \mathbf{c}, \hat{\mathbf{c}}')$, and the matrix $\mathbf{E}_1 \in \mathbb{Z}^{m\times k}$ that can be calculated are known by the algorithm $\mathcal{B}$. If the random oracle H was not queried or programmed on some input $\mathbf{w} = \hat{\mathbf{Y}} = \mathbf{A}\hat{\mathbf{z}} - \hat{\mathbf{T}}'\hat{\mathbf{h}}$, then $\mathcal{F}$ only has a probability of $\frac{1}{|D_H|}$ to produce a vector $\hat{\mathbf{h}}$ such that $\hat{\mathbf{h}} = H(\mathbf{w}, \hat{\mathbf{m}})$, so $\hat{\mathbf{h}}$ is one of the $\mathbf{r}_i$'s with probability $1 - \frac{1}{|D_H|}$. Thus the probability that $\mathcal{F}$ succeeds in forging and $\hat{\mathbf{h}}$ is one of the $\mathbf{r}_i$'s, is at least $\delta - \frac{1}{|D_H|}$. Let j be such that $\hat{\mathbf{h}} = H(\hat{\mathbf{Y}}, \hat{\mathbf{m}}) = \mathbf{r}_j$.

In the above case, $\mathcal{B}$ records the forged message-signature pair $((\hat{\mathbf{z}}, \hat{\mathbf{Y}}, \hat{\mathbf{c}}'), \hat{\mathbf{m}})$, and then generates fresh random elements $\mathbf{r}_j^*, \cdots, \mathbf{r}_t^* \leftarrow D_H$. Then $\mathcal{B}$ returns the forger $\mathcal{F}$ the same randomness tape and answers to the random oracle H as the previous run until j-th query. By the General Forking Lemma

of Bellare and Neven [25], we have that the probability that $\mathbf{r}_j^* \neq \mathbf{r}_j$ and $\mathcal{F}$ uses $\mathbf{r}_j^*$ this random oracle response in his forgery, is at least

$$\left(\delta - \frac{1}{|D_H|}\right)\left(\frac{\delta - \frac{1}{|D_H|}}{t} - \frac{1}{|D_H|}\right),$$

thus with the above probability, $\mathcal{F}$ outputs another forged signature $(\mathbf{z}^*, \mathbf{Y}^*, \mathbf{c}'^*)$ on the message $\hat{\mathbf{m}}$, where $\mathbf{Y}^* = \hat{\mathbf{Y}}$, $\mathbf{r}_j^* = \mathbf{h}^* = H(\mathbf{Y}^*, \hat{\mathbf{m}})$. Hence, the algorithm $\mathcal{B}$ gets that

$$\mathbf{A}\hat{\mathbf{z}} - \hat{\mathbf{T}}'\hat{\mathbf{h}} = \mathbf{A}\mathbf{z}^* - \mathbf{T}'^*\mathbf{h}^* \tag{6}$$

where $\mathbf{T}'^* = \mathbf{A}\mathbf{S}'^*$, and

$$\mathbf{S}'^* = \mathbf{S} + \Delta\mathbf{S}'^* + (2d+1)\mathbf{E}_2 \in \{-d, \cdots, d\}^{m\times k}. \tag{7}$$

The above (7) shows the existence of the secret key $\mathbf{S}'^*$, since the secret key $\mathbf{S}$, $\Delta\mathbf{S}'^* \leftarrow \mathsf{LS.DiffRec}(pp_{LS}, \mathbf{c}, \mathbf{c}'^*)$, and the matrix $\mathbf{E}_2 \in \mathbb{Z}^{m\times k}$ that can be calculated are known by the algorithm $\mathcal{B}$. Then $\mathcal{B}$ rearranges terms in the above (6) and plugging in $\hat{\mathbf{T}}' = \mathbf{A}\hat{\mathbf{S}}'$, $\mathbf{T}'^* = \mathbf{A}\mathbf{S}'^*$, then we have that

$$\mathbf{A}(\hat{\mathbf{z}} - \mathbf{z}^* + \mathbf{S}'^*\mathbf{h}^* - \hat{\mathbf{S}}'\hat{\mathbf{h}}) = 0.$$

Thus, the algorithm $\mathcal{B}$ outputs $\hat{\mathbf{z}} - \mathbf{z}^* + \mathbf{S}'^*\mathbf{h}^* - \hat{\mathbf{S}}'\hat{\mathbf{h}}$ as its own solution of the $\mathsf{SIS}_{q,m,n,\beta}$ instance. Since $\|\hat{\mathbf{z}}\|, \|\mathbf{z}^*\| \leq \eta\sigma\sqrt{m}$, and $\|\hat{\mathbf{S}}'\hat{\mathbf{h}}\|, \|\mathbf{S}'^*\mathbf{h}^*\| \leq dk\sqrt{m}$, we have that

$$\|\hat{\mathbf{z}} - \mathbf{z}^* + \mathbf{S}'^*\mathbf{h}^* - \hat{\mathbf{S}}'\hat{\mathbf{h}}\| \leq (2\eta\sigma + 2dk)\sqrt{m}.$$

Now, we analyze the probability of

$$\hat{\mathbf{z}} - \mathbf{z}^* + \mathbf{S}'^*\mathbf{h}^* - \hat{\mathbf{S}}'\hat{\mathbf{h}} \neq 0. \tag{8}$$

From Lemma 4, we know that for any $\mathbf{S} \leftarrow \{-d, \cdots, 0, \cdots, d\}^{m\times k}$, there is at least a probability of $1 - 2^{-100}$ existing another secret key $\mathbf{S}_1 \in \{-d, \cdots, 0, \cdots, d\}^{m\times k}$ such that all the columns of $\mathbf{S}_1$, except the column i, are the same as $\mathbf{S}$, and $\mathbf{AS} = \mathbf{AS}_1$. Hence, assume that $\hat{\mathbf{S}}_1'$ and $\mathbf{S}_1'^*$ are two another secret keys corresponding to $\hat{\mathbf{S}}'$ and $\mathbf{S}'^*$, respectively. Therefore, if $\hat{\mathbf{z}} - \mathbf{z}^* + \mathbf{S}'^*\mathbf{h}^* - \hat{\mathbf{S}}'\hat{\mathbf{h}} = 0$, then we have at least $\hat{\mathbf{z}} - \mathbf{z}^* + \mathbf{S}_1'^*\mathbf{h}^* - \hat{\mathbf{S}}'\hat{\mathbf{h}} \neq 0$ or $\hat{\mathbf{z}} - \mathbf{z}^* + \mathbf{S}'^*\mathbf{h}^* - \hat{\mathbf{S}}_1'\hat{\mathbf{h}} \neq 0$. Morever, the signing key $\mathbf{S} = h_{(\mathbf{B},\mathbf{v})}(g_{\mathcal{L}}(\mathbf{x}))$ where elements $\mathbf{x}$ and $\mathbf{v}$ are chosen at random, hence $\mathbf{S}$ can be regarded as being chosen randomly. From the way of computing $\hat{\mathbf{S}}'$ and $\mathbf{S}'^*$, we get that $\hat{\mathbf{S}}'$ and $\mathbf{S}'^*$ can also be treated as being chosen at random. Thus we will have a non-zero answer with probability at least $\frac{1}{2}$, since the forger $\mathcal{F}$ does not know which signing keys we used, and each key ($\mathbf{S}$, $\hat{\mathbf{S}}'$ or $\mathbf{S}'^*$) has an equal probability to be chosen since they can be regarded as being randomly chosen. Hence, if the forger $\mathcal{F}$ succeeds in forging in this EUF-CMA game with probability δ, then the algorithm $\mathcal{B}$ can use him to recover a non-zero vector $\mathbf{v} = \hat{\mathbf{z}} - \mathbf{z}^* + \mathbf{S}'^*\mathbf{h}^* - \hat{\mathbf{S}}'\hat{\mathbf{h}}$ such that $\mathbf{Av} = 0$ and $\|\mathbf{v}\| \leq (2\eta\sigma + 2dk)\sqrt{m}$ with probability at least

$$\left(\frac{1}{2} - 2^{-100}\right)\left(\delta - 2^{-100}\right)\left(\frac{\delta - 2^{-100}}{g+s} - 2^{-100}\right) \approx \frac{\delta^2}{2(g+s)}.$$

This finally completes the proof. □

V. CONCLUSION

In this work, we proposed a lattice-based fuzzy signature scheme which is constructed by linear sketch. We modified the linear sketch proposed by [7], and combined it with the signature scheme of [10] to obtain our lattice-based fuzzy signature scheme. Specifically, the modified linear sketch can not only benefit from the original construction of linear sketch in [7] which can be implemented securely and efficiently, but also be capable of being employed into the lattice-based setting. Moreover, Table 1 shows that our scheme has a promising tendency in efficiency among the existing lattice-based fuzzy signature schemes. In addition, our proposed fuzzy signature scheme is provably secure in the random oracle model. As our future work, we plan to construct the generic fuzzy signature scheme from linear sketch.

REFERENCES

[1] K. Takahashi, T. Matsuda, T. Murakami, G. Hanaoka, and M. Nishigaki, "A signature scheme with a fuzzy private key," in *Proc. ACNS*, in Lecture Notes in Computer Science, vol. 9092, 2015, pp. 105–126, doi: 10.1007/978-3-319-28166-7_6.

[2] Y. Dodis, L. Reyzin, and A. D. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. EUROCRYPT*, in Lecture Notes in Computer Science, vol. 3027, 2004, pp. 523–540, doi: 10.1007/978-3-540-24676-3_31.

[3] X. Boyen, "Reusable cryptographic fuzzy extractors," in *Proc. CCS*, Oct. 2004, pp. 82–91, doi: 10.1145/1030083.1030096.

[4] N. Li, F. Guo, Y. Mu, W. Susilo, and S. Nepal, "Fuzzy extractors for biometric identification," in *Proc. ICDCS*, Jun. 2017, pp. 667–677, doi: 10.1109/ICDCS.2017.107.

[5] Y. Tian, Y. Li, R. H. Deng, B. Sengupta, and G. Yang, "Lattice-based remote user authentication from reusable fuzzy signature," *J. Comput. Secur.*, vol. 29, no. 3, pp. 273–298, May 2021, doi: 10.3233/JCS-191370.

[6] J. Song and Y. Wen, "A generic construction of fuzzy signature," in *Proc. Inscrypt*, in Lecture Notes in Computer Science, vol. 13007, 2021, pp. 23–41, doi: 10.1007/978-3-030-88323-2_2.

[7] S. Katsumata, T. Matsuda, W. Nakamura, K. Ohara, and K. Takahashi, "Revisiting fuzzy signatures: Towards a more risk-free cryptographic authentication system based on biometrics," in *Proc. CCS*, Nov. 2021, pp. 2046–2065, doi: 10.1145/3460120.3484586.

[8] A. Kaafarani and S. Katsumata, "Post-quantum signature scheme using biometrics or fuzzy data," U.S. Patent 0 103 375 A1, Jan. 31, 2022. [Online]. Available: https://patentimages.storage.googleapis.com/2d/62/5f/1db456bc4c5f09/US20220103375A1.pdf

[9] T. Matsuda, K. Takahashi, T. Murakami, and G. Hanaoka, "Fuzzy signatures: Relaxing requirements and a new construction," in *Proc. ACNS*, in Lecture Notes in Computer Science, vol. 9696, 2016, pp. 97–116, doi: 10.1007/978-3-319-39555-5_6.

[10] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Proc. EUROCRYPT*, in Lecture Notes in Computer Science, vol. 7237, 2012, pp. 738–755, doi: 10.1007/978-3-642-29011-4_43.

[11] J. Zhang, Y. Chen, and Z. F. Zhang, "Programmable hash functions from lattices: Short signatures and IBEs with small key sizes," in *Proc. CRYPTO*, 2016, pp. 303–332, doi: 10.1007/978-3-662-53015-3_11.

[12] L. Ducas and D. Micciancio, "Improved short lattice signatures in the standard model," in *Proc. CRYPTO*, 2014, pp. 335–352, doi: 10.1007/978-3-662-44371-2_19.

[13] K. Takahashi, T. Matsuda, T. Murakami, G. Hanaoka, and M. Nishigaki, "Signature schemes with a fuzzy private key," *Int. J. Inf. Secur.*, vol. 18, no. 5, pp. 581–617, Oct. 2019, doi: 10.1007/s10207-019-00428-z.

[14] B. Waters, "Efficient identity-based encryption without random oracles," in *Proc. EUROCRYPT*, in Lecture Notes in Computer Science, vol. 3494, 2005, pp. 114–127, doi: 10.1007/11426639_7.

[15] C. P. Schnorr, "Efficient signature generation by smart cards," *J. Cryptol.*, vol. 4, no. 3, pp. 161–174, Jan. 1991, doi: 10.1007/BF00196725.

[16] M. Yasuda, T. Shimoyama, M. Takenaka, N. Abe, S. Yamada, and J. Yamaguchi, "Recovering attacks against linear sketch in fuzzy signature schemes of ACNS 2015 and 2016," in *Proc. ISPEC*, in Lecture Notes in Computer Science, vol. 10701, 2017, pp. 409–421, doi: 10.1007/978-3-319-72359-4_24.
[17] D. Apon, C. Cho, K. Eldefrawy, and J. Katz, "Efficient, reusable fuzzy extractors from LWE," in *Proc. CSCML*, in Lecture Notes in Computer Science, vol. 10332, 2017, pp. 1–18, doi: 10.1007/978-3-319-60080-2_1.
[18] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM J. Comput.*, vol. 17, no. 2, pp. 281–308, Apr. 1988, doi: 10.1137/0217017.
[19] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. ACM*, 2008, pp. 197–206, doi: 10.1145/1374376.1374407.
[20] W. Banaszczyk, "New bounds in some transference theorems in the geometry of numbers," *Math. Ann.*, vol. 296, no. 1, pp. 625–635, Dec. 1993, doi: 10.1007/BF01445125.
[21] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," *SIAM J. Comput.*, vol. 37, no. 1, pp. 267–302, Jan. 2007, doi: 10.1137/S0097539705447360.
[22] J. von Neumann, "Various techniques used in connection with random digits," in *Monte Carlo Method* (National Bureau of Standards Applied Mathematics), vol. 12. Washington, DC, USA: U.S. Government Printing Office, 1951, pp. 36–38.
[23] M. E. Schuckers, *Computational Methods in Biometric Authentication, Statistical Methods for Performance Evaluation*. London, U.K.: Springer, Jan. 2010, doi: 10.1007/978-1-84996-202-5.
[24] C. Peikert, "A decade of lattice cryptography," *Found. Trends Theor. Comput. Sci.*, vol. 10, no. 4, pp. 283–424, 2016.
[25] M. Bellare and G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," in *Proc. CCS*, 2006, pp. 390–399, doi: 10.1145/1180405.1180453.

MINGMEI ZHENG received the M.S. degree from the College of Mathematics and Informatics, Fujian Normal University, China, in 2019. She is currently pursuing the Ph.D. degree with the Division of Electrical Engineering and Computer Science, Kanazawa University, Japan. Her research interests include applied cryptography, especially the quantum-resistant cryptography from lattice, and digital signatures.

ZI-YUAN LIU received the B.E. degree from the Department of Computer Science, National Tsing Hua University, Taiwan, in 2016, and the M.E. degree from the Department of Computer Science, National Chengchi University, Taiwan, in 2018, where he is currently pursuing the joint Ph.D. degree with the Department of Computer Science, and the Division of Electrical Engineering and Computer Science, Kanazawa University, Japan. His research interests include applied cryptography, particularly in the areas of asymmetric searchable encryption and identity-based encryption.

MASAHIRO MAMBO (Member, IEEE) received the B.Eng. degree from Kanazawa University, Kanazawa, Japan, in 1988, and the M.S.Eng. and Dr.Eng. degrees in electronic engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1990 and 1993, respectively. In 2011, after working with the Japan Advanced Institute of Science and Technology, Tohoku University, Sendai, Japan, and the University of Tsukuba, Tsukuba, Japan, he joined Kanazawa University. He is currently a Professor with the Faculty of Electrical, Information and Communication Engineering, Institute of Science and Engineering. His research interests include information security, software protection, and privacy protection. He has served as the Co-Editor-in-Chief for the *International Journal of Information Security*, and the Steering Committee Chair of the International Conference on Information Security and the Chair of the Technical Committee on Information Security in Engineering Sciences Society of the Institute of Electronics, Information and Communication Engineers (ISEC in ESS of IEICE).