axioms

MDPI

Article

Fuzzy Signature from Computational Diffie–Hellman Assumption in the Standard Model

Yunhua Wen *, Tianlong Jin and Wei Li

School of Computer Science and Technology, Donghua University, Shanghai 201620, China; 2232987@mail.dhu.edu.cn (T.J.); weili@dhu.edu.cn (W.L.)
* Correspondence: yhwen@dhu.edu.cn

Abstract

Fuzzy signature ($\mathsf{SIG_F}$) is a type of digital signature that preserves the core functionalities of traditional signatures, while accommodating variations and non-uniformity in the signing key. This property enables the direct use of high-entropy fuzzy data, such as biometric information, as the signing key. In this paper, we define the $\mathfrak{m}$-existentially unforgeable under chosen message attack ($\mathfrak{m}$-EUF-CMA*) security of fuzzy signature. Furthermore, we propose a generic construction of fuzzy signature, which is composed of a homomorphic secure sketch (SS) with an error-recoverable property, a homomorphic average-case strong extractor (Ext), and a homomorphic and key-shift* secure signature scheme (SIG). By instantiating the foundational components, we present a $\mathfrak{m}$-EUF-CMA* secure fuzzy signature instantiation based on the Computational Diffie–Hellman (CDH) assumption over bilinear groups in the standard model.

Keywords: fuzzy signature; secure sketch; digital signature

MSC: 94A60

check for updates

Academic Editor: Abbe Mowshowitz

Received: 19 June 2025
Revised: 2 August 2025
Accepted: 4 August 2025
Published: 6 August 2025

Citation: Wen, Y.; Jin, T.; Li, W. Fuzzy Signature from Computational Diffie–Hellman Assumption in the Standard Model. *Axioms* **2025**, *14*, 613. https://doi.org/10.3390/axioms14080613

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

1.1. Digital Signature and Fuzzy Signature

A digital signature [1,2] is a fundamental cryptographic primitive that ensures integrity, authentication, and non-repudiation in digital communication. Traditional signature algorithms [3] generally require the signing keys to be uniformly random and precisely accurate. When using a digital signature for authentication, users need to take care to retain their signing key. They may store the signing key on devices such as a smart card [4,5] or a USB token [6,7]. Hence, it is unavoidable for users to carry an additional device.

Fuzzy signature [8] is an advanced digital signature scheme that permits non-uniformity and approximate matching of the signing key. This property enables fuzzy signatures to leverage fuzzy data (e.g., biometric information) as the signing key. Note that the same individual's biometric information [9–11] (e.g., fingerprints, iris scans, and facial features) may vary slightly with each capture due to environmental conditions, equipment differences, and measurement noise. Additionally, the distribution of biometric information is not uniform, but has high entropy.

Fuzzy signature ($\mathsf{SIG_F}$) consists of four algorithms, as depicted in Figure 1. The setup algorithm $\mathsf{Setup_F}$ takes the security parameter as input, and it outputs the public parameters pp. Given the public parameters pp and a fuzzy signing key sk, such as a sample of biometric data, the key-generation algorithm $\mathsf{KeyGen_F}$ computes the verification key vk.

The signing algorithm Sign_F takes sk' (a new sample of the same biometric data), a message m, and the public parameters pp as inputs, and it outputs a signature σ. The verification algorithm $\mathsf{Verify}_\mathsf{F}$ takes pp, vk, a message m, and a signature σ as inputs, and it outputs 1 (valid) or 0 (invalid). The signature σ is guaranteed to be valid when the signing keys sk and sk' are sufficiently close.

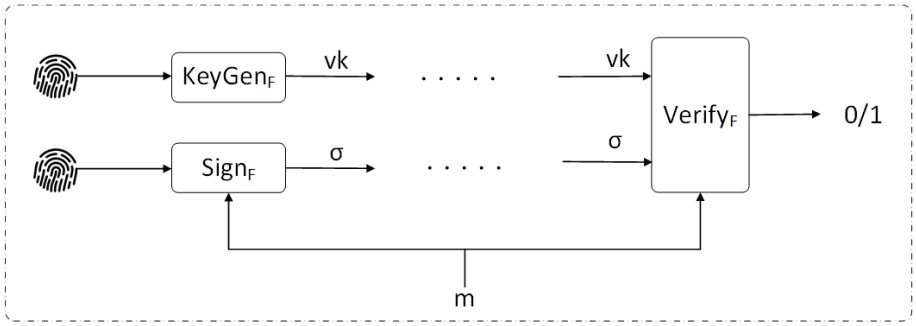

Figure 1. Fuzzy signature SIG_F. The public parameters pp are omitted for brevity.

The notion of fuzzy signature was first introduced by Takahashi et al. [8]. In their work, they proposed a general construction of fuzzy signature based on a tool called linear sketch and a signature with homomorphic properties regarding keys and signatures. However, their security model assumes that the differences between different samples drawn from identical biometric data are independent of the biometric data itself, which deviates from real-world applications. In addition, their fuzzy signature has a critical limitation: it requires biometric information to be uniformly distributed, a condition that is clearly unrealistic in practical scenarios. Furthermore, their fuzzy signature requires bilinear groups and has large public parameters (proportional to the security parameter), making it less practical for real-world deployment.

Matsuda et al. [12] relaxed the constraints imposed on the underlying linear sketch and homomorphic signature scheme. Through this relaxation, they constructed a new fuzzy signature scheme based on the Schnorr signature scheme [13] and an improved linear sketch. Their construction operates under the assumption that the fuzzy data has high average min-entropy, even in the presence of information leakage. Thanks to the use of the Schnorr signature scheme, their fuzzy signature scheme achieves greater efficiency. Nevertheless, its security is guaranteed only in the random oracle model.

In 2021, Katsumata et al. [14] proposed a simple and efficient generic construction of fuzzy signature based on linear sketch and the tweaked Schnorr signature scheme. In their work, the linear sketch is formalized using lattice theory [15]. The security of the scheme relies on the $\mathrm{DL}^{\mathrm{sketch}}$ assumption, which combines the hardness of the discrete logarithm problem (DLP) [16] with the security of the sketch mechanism. Analysis shows that under a low Conditional False Matching Rate (ConFMR), $\mathrm{DL}^{\mathrm{sketch}}$ reduces to the standard DLP, and remains unconditionally secure in the generic group model, even for higher ConFMR values. To connect theory with practice, the authors develop a statistical framework using extreme value analysis and t-tests to estimate the ConFMR for real biometric data. However, their fuzzy signature is secure only in the random oracle model [17]. Similarly to previous works, their fuzzy signature requires that the differences between different samples drawn from identical biometric data are independent of the biometric data itself, which is not compatible with real-world applications.

Song et al. [18] introduced a novel security model called $\mathfrak{m}$-existentially unforgeable under chosen message attack ($\mathfrak{m}$-EUF-CMA) security for fuzzy signature, which requires only that the fuzzy signing key sk possesses sufficient entropy [19] and that the error distribution across different biometric samples can be arbitrarily correlated to the biometric

data. In addition, they proposed a generic construction of fuzzy signature based on a fuzzy extractor [20] and a signature scheme with a simple key-generation process. By instantiating their framework with different signature schemes, they presented two concrete constructions: one based on the Computational Diffie–Hellman (CDH) assumption [21] in the standard model, and another based on lattice assumptions which is post-quantum secure in the random oracle model.

However, their signature generation algorithm requires the verification key vk as an additional input, which may impose a usability burden on the user. For instance, consider a scenario where a user signs messages using biometric data (e.g., a fingerprint or iris scan) as the signing key. If vk is required during signing, the user must securely store vk and may need to carry an external device (e.g., a smart card or USB token), which introduces usability challenges and potential security risks. Is such a device is compromised, it could undermine the overall security of the system. In contrast, if vk is not required during the signing process, the user can generate signatures solely based on biometric data, without the need to manage or protect the verification key. This leads to a more practical, user-friendly, and device-free signing experience.

With the advancement of quantum computing, growing attention has been devoted to the development of post-quantum secure cryptographic algorithms. Tian et al. [22] presented a new reusable fuzzy signature based on a reusable fuzzy extractor from LWE [23–25] and lattice-based digital signatures [26]. While their fuzzy signature scheme achieves post-quantum security, its security is established in the random oracle model.

In 2023, zheng et al. [27] proposed another efficient quantum-resistant fuzzy signature scheme based on the small integer solution problem (SIS) [28] by employing a modified linear sketch. Although their security proof is more rigorous than that of prior work, it can only be established under the random oracle model.

The aforementioned fuzzy signature schemes typically employ either a linear sketch or a fuzzy extractor as the core technique to bridge the gap between fuzzy biometric data and cryptographic primitives (e.g., signing keys).

A fundamental limitation of fuzzy signature schemes [8,12,14,27] based on linear sketches is the requirement that the errors across different biometric samples are statistically independent of the biometric data itself. This assumption is difficult to justify in practice, as real-world biometric variations are typically correlated with the underlying data, rendering such schemes impractical for real-life applications.

On the other hand, fuzzy extractor–based constructions face their own limitations. For instance, the scheme proposed in [22] is only proven secure in the random oracle model, while the one in [18] requires the verification key as input during signing, which introduces usability and security challenges.

Motivated by these observations, we are naturally led to the following question:

Does there exist a construction of a fuzzy signature scheme that (1) relaxes the assumptions on the distribution of fuzzy data, (2) does not require the verification key as input during signing, and (3) achieves security in the standard model without relying on random oracles?

1.2. Our Contributions

In this work, we provide an affirmative answer to the above question. The principal contributions of our work are as follows:

– First, we redefine the notion of fuzzy signature and propose a new security model, which we call $\mathfrak{m}$-existential unforgeability under chosen-message attack ($\mathfrak{m}$-EUF-CMA*), for fuzzy signatures. Compared with previous work [18], this new definition eliminates the requirement for the signing algorithm to take a verification key as input, thereby reducing the burden on users to maintain the verification key.

Furthermore, our new security model is stronger than the one proposed by Matsuda et al. [12], because our security model allows the adversary to determine the differences between different biometric samples.

- We formally define the error-recoverable property of the secure sketch and show that the syndrome-based secure sketch possesses the error recoverable property. We redefine the homomorphic property and define the key-shift* security of the signature scheme. In our key-shift* security model, the adversary can modify not only the signing key sk, but also the verification key vk. We also prove that the Waters signature scheme $\mathsf{SIG}_{\mathsf{Wat}}$ is homomorphic and key-shift* secure.
- We present a generic construction of a fuzzy signature scheme, built upon three core primitives: a homomorphic secure sketch (SS) with error recovery, a homomorphic average-case strong extractor (Ext), and a homomorphic signature scheme (SIG) satisfying key-shift security.
- Instantiating these primitives yields a fuzzy signature scheme that achieves m-EUF-CMA* security under the Computational Diffie–Hellman (CDH) assumption in the standard model over bilinear groups, and, notably, does not require the verification key as input to the signing algorithm.

In Table 1, we present an overview of our constructions compared with existing fuzzy signature schemes.

Table 1. An overview of representative fuzzy signature schemes. "SIG_F" is the abbreviation of fuzzy signature scheme. "Method" represents the primary cryptographic primitive utilized in the construction of SIG_F. "Correlation" explains the relationship between the error distribution e and the biometric information W, where e captures the deviation between the fuzzy signing key sk and sk' (i.e., two readings derived from identical biometric input W). "vk" indicates that the signing algorithm does not need vk as input. "Assumption" specifies the cryptographic assumption underlying the SIG_F. "Standard Model" asks whether the SIG_F achieves security in the standard model.

SIG_F	Method	Correlation	vk	Assumption	Standard Model
TMMHN15 [8]	Linear Sketch	Independent	✔	CDH	✔
MTMH16 [12]	Linear Sketch	Independent	✔	DL	✗
KK21 [14]	Linear Sketch	Independent	✔	DL	✗
ZLM23 [27]	Linear Sketch	Independent	✔	SIS	✗
SW21 [18]	Fuzzy Extractor	Arbitrary	✗	CDH	✔
TLDSY21 [22]	Fuzzy Extractor	Adversary-Selected	✔	LWE, SIS	✗
Our Scheme	Secure Sketch	Adversary-Selected	✔	CDH	✔

1.3. Our Approach

Our generic construction of fuzzy signature consists of a homomorphic secure sketch with an error-recoverable property, a homomorphic average-case strong extractor, and a SIG with a homomorphic property and a simple key-generation process.

In the setup algorithm $\mathsf{Setup}_\mathsf{F}$, the seed i of extractor Ext and public parameters pp, generated by the underlying signature scheme's setup algorithm Setup, serve as the public parameters pp_F of the entire fuzzy signature SIG_F. In the key-generation algorithm $\mathsf{KeyGen}_\mathsf{F}$, the public parameters pp_F and a biometric sample w are taken as input, and the verification key vk_F of SIG_F is generated. Specifically, w is first computed by the sketch generation algorithm SS.Gen to produce a sketch s, and then the extractor Ext is used to extract the secret key sk from w for the underlying signature scheme (SIG). Subsequently, the SIG's

simple key-generation process enables the verification key vk of the SIG to be derived. Therefore, the final verification key of the $\mathsf{SIG_F}$ is given by $vk_{\mathsf{F}} = (vk, \mathsf{s})$.

The signing algorithm $\mathsf{Sign_F}$ uses a new biometric sample w' as the signing key of fuzzy signature to sign the message m. Specifically, the process begins by computing a new sketch s' from w', while the extractor Ext is used to extract the new sk' as the secret key (signing key) for the underlying signature scheme SIG. Subsequently, the message m and the sketch s' are signed together using the SIG to obtain σ. The final signature σ_{F} is then given by (σ, s'). In $\mathsf{Verify_F}$, utilizing the values of s and s' generated during the $\mathsf{KeyGen_F}$ and $\mathsf{Sign_F}$, the error-recovery property of SS allows us to compute $\Delta\mathsf{w} = \mathsf{w}' - \mathsf{w}$. This enables the extractor Ext to calculate $\Delta sk = sk' - sk$. Next, thanks to the homomorphic property of the SIG, we can obtain the verification key vk' corresponding to sk', thereby ensuring the correctness of SIG. Consequently, the correctness of the entire fuzzy signature scheme $\mathsf{SIG_F}$ is also guaranteed. In addition, the security of our $\mathsf{SIG_F}$ is ensured by Ext and SIG. More precisely, the security of Ext guarantees the uniform randomness of the secret key in SIG, thereby guaranteeing the security of SIG. Furthermore, the $\mathfrak{m}$-EUF-CMA* security we defined for $\mathsf{Sign_F}$ can be reduced to the key-shift* security of SIG, which is further guaranteed by the homomorphic property and EUF-CMA security of SIG. Detailed information on this is presented in Section 4.

2. Preliminaries

Throughout this paper, we denote the set of integers by $\mathbb{Z}$ and the set of real numbers by $\mathbb{R}$. The notation "$u \leftarrow v$" denotes a deterministic assignment of the value v to the variable u. Given a finite set $\mathcal{D}$, we use $d \leftarrow_{\$} \mathcal{D}$ to indicate that d is drawn uniformly at random from $\mathcal{D}$. The cardinality of a set $\mathcal{D}$ is denoted by $|\mathcal{D}|$. For any bit strings a and b, we write $|a|$ for the bit-length of a, and write $a \parallel b$ to denote the concatenation of a and b. A probabilistic algorithm that operates in polynomial time is referred to as a PPT algorithm. For a natural number k, we write $[k]$ for the index set $\{1, 2, \ldots, k\}$. Let $\mathcal{A}$ be a randomized procedure; we express $\rho \leftarrow \mathcal{A}(x; r)$ to represent the outcome ρ when $\mathcal{A}$ is executed on input x with internal randomness r. A function $\mathsf{negl}(n)$ is called negligible if for every positive polynomial $q(\cdot)$, there exists an integer n_0, such that for all $n > n_0$, it holds that $\mathsf{negl}(n) \leq \frac{1}{q(n)}$.

2.1. Metric Spaces

Definition 1 (**Metric Space** [29]). *A metric space is a set $\mathcal{W}$ equipped with a distance function* $\mathrm{dis}\colon \mathcal{W} \times \mathcal{W} \to \mathbb{R}_{\geq 0}$, *where* dis *satisfies the following properties for all* $\mathsf{w}_1, \mathsf{w}_2, \mathsf{w}_3 \in \mathcal{W}$:

(i) $\mathrm{dis}(\mathsf{w}_1, \mathsf{w}_2) = 0$ *if, and only if,* $\mathsf{w}_1 = \mathsf{w}_2$;

(ii) $\mathrm{dis}(\mathsf{w}_1, \mathsf{w}_2) = \mathrm{dis}(\mathsf{w}_2, \mathsf{w}_1)$ *(symmetry);*

(iii) $\mathrm{dis}(\mathsf{w}_1, \mathsf{w}_3) \leq \mathrm{dis}(\mathsf{w}_1, \mathsf{w}_2) + \mathrm{dis}(\mathsf{w}_2, \mathsf{w}_3)$ *(triangle inequality).*

2.2. Min-Entropy and Statistical Distance

Definition 2 (**Min-Entropy** [29]). *Let ξ be a discrete random variable over a finite set $\mathcal{D}$. The min-entropy of ξ is defined as*

$$H_\infty(\xi) = -\log\left(\max_{d \in \mathcal{D}} \Pr[\xi = d]\right).$$

Definition 3 (**Average Min-Entropy** [20]). *Let A and B be two discrete random variables over finite sets $\mathcal{A}$ and $\mathcal{B}$, respectively. The average min-entropy of A given B is defined as*

$$\tilde{H}_\infty(A \mid B) = -\log\left(\mathbb{E}_{b \leftarrow B}\left[\max_{a \in \mathcal{A}} \Pr[A = a \mid B = b]\right]\right).$$

Definition 4 (**Statistical Distance** [20]). *Let U and V be two random variables taking values in a finite set $\mathcal{W}$. The statistical distance between U and V is defined as*

$$\mathrm{SD}(U, V) = \frac{1}{2} \sum_{\mathsf{w} \in \mathcal{W}} |\Pr[U = \mathsf{w}] - \Pr[V = \mathsf{w}]|.$$

2.3. Computational Diffie–Hellman (CDH) Assumption

Let $\mathcal{IG}$ be an algorithm that outputs $(p, g, \mathbb{G})$, where $\mathbb{G}$ is a cyclic group with order p, and $g \in \mathbb{G}$ is a generator.

Definition 5 (CDH Problem). *Given the four-tuple $(\mathbb{G}, g, g^a, g^b)$, compute g^{ab}, where $(p, g, \mathbb{G}) \leftarrow \mathcal{IG}$ and $a, b \in \mathbb{Z}_p$.*

Definition 6 (CDH Assumption). *The CDH assumption w.r.t. $(p, g, \mathbb{G})$ is ϵ-hard if, for any PPT adversary $\mathcal{A}$, the following holds:*

$$\mathsf{Adv}^{\mathrm{CDH}}_{\mathcal{A}} := \Pr[\mathcal{A}(\mathbb{G}, g, g^a, g^b) \to g^{ab}] \leq \epsilon,$$

where $(p, g, \mathbb{G}) \leftarrow \mathcal{IG}$ and $a, b \in \mathbb{Z}_p$.

2.4. Secure Sketch and Average-Case Strong Extractor

Definition 7 (**Secure Sketch** [20]). *A $(\mathcal{W}, \mathfrak{m}, \hat{\mathfrak{m}}, t)$-secure sketch (SS) consists of two efficient algorithms, defined as follows:*

- $\mathsf{SS.Gen}(\mathsf{w})$ *on input* $\mathsf{w} \in \mathcal{W}$, *outputs a sketch* $\mathsf{s} \in \mathcal{S}$.
- $\mathsf{SS.Rec}(\mathsf{w}', \mathsf{s})$ *on input* $\mathsf{w}' \in \mathcal{W}$ *and a sketch* $\mathsf{s} \in \mathcal{S}$, *outputs a recovered value* $\tilde{\mathsf{w}}$.

These algorithms must satisfy the following properties:

- Correctness : *When* $\mathrm{dis}(\mathsf{w}, \mathsf{w}') \leq t$, *then* $\mathsf{w} = \mathsf{SS.Rec}(\mathsf{w}', \mathsf{SS.Gen}(\mathsf{w})) = \tilde{\mathsf{w}}$.
- Security : *For any distribution W over $\mathcal{W}$, if* $H_\infty(W) > \mathfrak{m}$, $\tilde{H}_\infty(W|\mathsf{SS.Gen}(W)) > \hat{\mathfrak{m}}$.

A secure sketch is regarded as homomorphic if for all $\mathsf{u}, \mathsf{v} \in \mathcal{W}$, it holds that

$$\mathsf{SS.Gen}(\mathsf{u} + \mathsf{v}) = \mathsf{SS.Gen}(\mathsf{u}) + \mathsf{SS.Gen}(\mathsf{v}).$$

Definition 8 (Error-Recoverable Property). *Let* $\mathsf{SS} = (\mathsf{SS.Gen}, \mathsf{SS.Rec})$ *be a* $(\mathcal{W}, \mathfrak{m}, \hat{\mathfrak{m}}, t)$*-secure sketch and* $\mathsf{w}, \mathsf{w}' \in \mathcal{W}$, *such that* $\mathrm{dis}(\mathsf{w}, \mathsf{w}') \leq t$. *Let* $\Delta\mathsf{w} = \mathsf{w}' - \mathsf{w}$ *and* $\mathsf{s} = \mathsf{SS.Gen}(\mathsf{w})$, $\mathsf{s}' = \mathsf{SS.Gen}(\mathsf{w}')$. *We say that the secure sketch has the error-recoverable property if there is a deterministic function f_{rec}, such that* $\Delta\mathsf{w} = f_{rec}(\mathsf{s}', \mathsf{s})$.

Remark 1. *In our fuzzy signature scheme, the user needs to compute the error between the fuzzy data* w *sampled during the generation of the verification key and* w' *used for signing, by utilizing the values* s *and* s'. *Here,* s *is included in the verification key, while* s' *is part of the signature. Through this mechanism, the scheme can verify the validity of the signature. The details can be found in Section 4 of our paper. The error-recoverable property ensures that the user can perform the above operations reliably.*

Lemma 1. *The error-recoverable property mentioned above is satisfied by the syndrome-based secure sketch construction.*

Proof. This lemma is a straightforward consequence of the inherent properties of the syndrome-based secure sketch construction [20]. Recall that the syndrome-based secure sketch is as follows. The generation algorithm $\mathsf{SS.Gen}(\mathsf{w}) = \mathrm{syn}(\mathsf{w}) = \mathsf{s}$, and the recovery

algorithm $\mathsf{SS.Rec}(\mathsf{w}', \mathsf{s}) = \mathsf{w}' - \mathsf{decode}(\mathsf{syn}(\mathsf{w}') - \mathsf{s}) = \mathsf{w}' - \mathsf{decode}(\mathsf{s}' - \mathsf{s})$, where decode is an efficient and deterministic function to find the unique $e \in \mathcal{W}$, where $\mathsf{dis}(e, 0) \leq t$ such that $\mathsf{syn}(e) = \mathsf{syn}(\mathsf{w}') - \mathsf{s}$. By the correctness of the secure sketch, if $\mathsf{dis}(\mathsf{w}, \mathsf{w}') \leq t$, then $\mathsf{w} = \mathsf{SS.Rec}(\mathsf{w}', \mathsf{s}) = \mathsf{w}' - \mathsf{decode}(\mathsf{s}' - \mathsf{s})$. It is obvious that $\Delta\mathsf{w} = \mathsf{w}' - \mathsf{w} = \mathsf{decode}(\mathsf{s}' - \mathsf{s})$. The lemma follows. □

For completeness, we provide the details of the syndrome-based secure sketch in Appendix A.

Definition 9 (**Average-Case Strong Extractor** [20]). *A function* $\mathsf{Ext}\colon \mathcal{W} \times \mathcal{R} \to \mathcal{V}$ *is called an average-case* $(\mathcal{W}, m, \mathcal{V}, \varepsilon_{ext})$*-strong extractor with randomness* $R \in \mathcal{R}$ *if for every random variable* W *over* $\mathcal{W}$ *and any auxiliary information* Z*, the following holds:*

$$\mathrm{SD}\big(\big(\mathsf{Ext}(W, R), R, Z\big),\ \big(U, R, Z\big)\big) \leq \varepsilon_{ext},$$

where R *and* U *are independently and uniformly sampled from* $\mathcal{R}$ *and* $\mathcal{V}$*, respectively.*

Definition 10 (**Homomorphic Average-Case Strong Extractor**). *An average-case* $(\mathcal{W}, m, \mathcal{V}, \varepsilon)$*-strong extractor* $\mathsf{Ext}\colon \mathcal{W} \times \mathcal{R} \to \mathcal{V}$ *is said to be homomorphic if for all* $\mathsf{w}_1, \mathsf{w}_2 \in \mathcal{W}$ *and all seeds* $r \in \mathcal{R}$*, it satisfies*

$$\mathsf{Ext}(\mathsf{w}_1 + \mathsf{w}_2, r) = \mathsf{Ext}(\mathsf{w}_1, r) + \mathsf{Ext}(\mathsf{w}_2, r).$$

One instantiation of an average-case strong extractor [30] is defined as follows:

$$\mathsf{Ext}(\mathsf{w}, \mathbf{r}) = \mathsf{w}_0 + r_1\mathsf{w}_1 + \cdots + r_l\mathsf{w}_l, \tag{1}$$

where $\mathbf{r} = (r_1, \ldots, r_l) \in \mathbb{Z}_p^l$ denotes the seed, and $\mathsf{w} = (\mathsf{w}_0, \mathsf{w}_1, \ldots, \mathsf{w}_l) \in \mathbb{Z}_p^{l+1}$ represents the input. Note that

$$\begin{aligned}
\mathsf{Ext}(\mathsf{w}_{\mathbf{0}} + \mathsf{w}_{\mathbf{1}}, \mathbf{r}) &= (\mathsf{w}_{0,0} + \mathsf{w}_{1,0}) + r_1(\mathsf{w}_{0,1} + \mathsf{w}_{1,1}) + \cdots + r_l(\mathsf{w}_{0,l} + \mathsf{w}_{1,l}) \\
&= \mathsf{w}_{0,0} + r_1\mathsf{w}_{0,1} + \cdots + r_l\mathsf{w}_{0,l} + \mathsf{w}_{1,0} + r_1\mathsf{w}_{1,1} + \cdots + r_l\mathsf{w}_{1,l} \\
&= \mathsf{Ext}(\mathsf{w}_{\mathbf{0}}, \mathbf{r}) + \mathsf{Ext}(\mathsf{w}_{\mathbf{1}}, \mathbf{r}).
\end{aligned}$$

Then we have that Equation (1) is a homomorphic average-case strong extractor.

2.5. Signature Scheme

Definition 11 (Signature Scheme). *A digital signature scheme* $\mathsf{SIG} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *consists of four probabilistic polynomial-time (PPT) algorithms.*

- $pp \leftarrow \mathsf{Setup}(1^\lambda)$*: On input the security parameter* λ*,* Setup *outputs public parameters* pp*.*
- $(vk, sk) \leftarrow \mathsf{KeyGen}(pp)$*: On input* pp*,* KeyGen *generates a key pair consisting of a verification key* $vk \in \mathcal{VK}$ *and a signing key* $sk \in \mathcal{SK}$*.*
- $\sigma \leftarrow \mathsf{Sign}(pp, sk, m)$*: On input* pp*,* sk*, and a message* $m \in \mathcal{M}$*,* Sign *outputs a signature* σ*.*
- $b \leftarrow \mathsf{Verify}(pp, vk, m, \sigma)$*: On input* pp*,* vk*,* m*, and* σ*,* Verify *outputs a bit* $b \in \{0, 1\}$ *indicating whether the signature is valid (1) or invalid (0).*

Correctness. For any message $m \in \mathcal{M}$, let $pp \leftarrow \mathsf{Setup}(1^\lambda)$, $(vk, sk) \leftarrow \mathsf{KeyGen}(pp)$, and signature $\sigma \leftarrow \mathsf{Sign}(pp, sk, m)$; then

$$\Pr[\mathsf{Verify}(pp, vk, m, \sigma) = 1] \geq 1 - \mathsf{negl}(\lambda).$$

Definition 12 (EUF-CMA Security)**.** *We say that a digital signature* SIG *achieves existentially unforgeable under adaptive chosen-message attack (EUF-CMA) security if for any PPT adversary, it holds that*

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

where $\mathsf{Expt}^{\mathsf{EUF-CMA}}_{\mathsf{SIG},\mathcal{A}}(1^\lambda)$ *is defined as follows.*

$\underline{\mathsf{Expt}^{\mathsf{EUF-CMA}}_{\mathsf{SIG},\mathcal{A}}(1^\lambda)}$*:*

1. *The challenger* $\mathcal{C}$ *runs* $pp \leftarrow \mathsf{Setup}(1^\lambda)$*, generates* $(vk, sk) \leftarrow \mathsf{KeyGen}(pp)$*, initializes an empty query set* $\mathcal{Q} = \emptyset$*, and sends* pp *and* vk *to the adversary* $\mathcal{A}$*.*
2. *Throughout the experiment,* $\mathcal{A}$ *may adaptively query a signing oracle:*
   - $\mathcal{A}$ *submits a message* $m_j \in \mathcal{M}$ *to* $\mathcal{C}$*.*
   - $\mathcal{C}$ *computes* $\sigma_j \leftarrow \mathsf{Sign}(pp, sk, m_j)$*, adds* m_j *to* $\mathcal{Q}$*, and returns* σ_j *to* $\mathcal{A}$*.*
3. *Finally,* $\mathcal{A}$ *outputs a forgery* (m^*, σ^*)*. The experiment outputs* 1 *if, and only if,*

$$m^* \notin \mathcal{Q} \quad \textit{and} \quad \mathsf{Verify}(pp, vk, m^*, \sigma^*) = 1.$$

   Otherwise, it outputs 0*.*

Definition 13 (Simple Key-Generation Process [8])**.** *We say that a digital signature scheme* $\mathsf{SIG} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *supports a* simple key-generation process *if the public parameters* $pp \leftarrow \mathsf{Setup}(1^\lambda)$ *implicitly define the signing key space* $\mathcal{SK}$*, and there exists a deterministic polynomial-time (*PPT*) algorithm* KG *such that* $\mathsf{KeyGen}(pp)$ *is functionally equivalent to the following procedure:*

$$\mathsf{KeyGen}(pp): \quad [sk \leftarrow_{\$} \mathcal{SK},\ vk \leftarrow \mathsf{KG}(pp, sk),\ \textit{return}\ (vk, sk)].$$

Definition 14 (Homomorphic Signature)**.** *A signature scheme* $\mathsf{SIG} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$*, equipped with a simple key-generation process* KG*, is said to be homomorphic if the following properties are satisfied:*

1. *For all* $pp \leftarrow \mathsf{Setup}(1^\lambda)$*, the signing key space* $\mathcal{SK}$ *constitutes an abelian group* $(\mathcal{SK}, +)$*.*
2. *For all* $sk, \Delta sk \in \mathcal{SK}$*, there exists a deterministic and efficient algorithm* A_{vk} *such that*

$$\mathsf{KG}(pp, sk + \Delta sk) = \mathsf{A}_{\mathsf{vk}}(pp, vk, \Delta sk),$$

   where $pp \leftarrow \mathsf{Setup}(1^\lambda)$*,* $vk \leftarrow \mathsf{KG}(pp, sk)$*.*
3. *Suppose the random space of the signing algorithm is* $\mathcal{R}$*. For all* $r \in \mathcal{R}$*,* $m \in \mathcal{M}$ *and* $sk, \Delta sk \in \mathcal{SK}$*, there exists a deterministic and efficient algorithm* $\mathsf{A}_{\mathsf{Sign}}$ *such that*

$$\mathsf{Sign}(pp, sk + \Delta sk, m, r) = \mathsf{A}_{\mathsf{Sign}}(pp, m, \sigma, \Delta sk),$$

   where $pp \leftarrow \mathsf{Setup}(1^\lambda)$*,* $\sigma \leftarrow \mathsf{Sign}(pp, sk, m, r)$*.*

Remark 2. *For simplicity, we sometimes omit* r *and say that there exists a deterministic and efficient algorithm* $\mathsf{A}_{\mathsf{Sign}}$ *such that*

$$\mathsf{Sign}(pp, sk + \Delta sk, m) = \mathsf{A}_{\mathsf{Sign}}(pp, m, \sigma, \Delta sk).$$

where $pp \leftarrow \mathsf{Setup}(1^\lambda)$*,* $\sigma \leftarrow \mathsf{Sign}(pp, sk, m)$*.*

Remark 3. *Our definition of a homomorphic signature differs from the definition provided in [8] in three aspects.*

1. *The first difference lies in the inputs required by the algorithm* $\mathsf{A}_{\mathsf{Sign}}$. *In our definition,* $\mathsf{A}_{\mathsf{Sign}}$ *takes* $pp, m, \sigma, \Delta sk$ *as input and generates a valid signature of* m *under the signing key* $sk + \Delta sk$. *However, in [8], the algorithm additionally requires* vk *as part of its input.*
2. *The second difference is that in our definition with the same random number* r, $\mathsf{Sign}(pp, sk + \Delta sk, m, r) = \mathsf{A}_{\mathsf{Sign}}(pp, m, \mathsf{Sign}(pp, sk, m, r), \Delta sk)$, *while in [8], the distribution of* $\{\sigma' \leftarrow \mathsf{Sign}(pp, sk + \Delta sk, m) : \sigma'\}$ *and* $\{\sigma \leftarrow \mathsf{Sign}(pp, sk, m); \sigma' \leftarrow \mathsf{A}_{\mathsf{Sign}}(pp, m, \sigma, \Delta sk) : \sigma'\}$ *is required to be identical.*
3. *The third difference is that our definition omits the requirement that for all* $sk, \Delta sk$ *and all pairs* (m, σ) *satisfying* $\mathsf{Verify}(pp, \mathsf{KG}(pp, sk), m, \sigma) = 1$, *it holds that* $\mathsf{Verify}(pp, \mathsf{KG}(pp, sk + \Delta sk), m, \mathsf{A}_{\mathsf{Sign}}(pp, m, \sigma, \Delta sk)) = 1$.

Next, we will formally define the key-shift* security of a signature scheme. In the traditional related-key attack (RKA) security experiment [31], only the signing key sk can be changed by the adversary $\mathcal{A}$. More precisely, the adversary is allowed to modify the secret key sk using a predefined set of transformations, and obtains signatures under the modified keys on adaptively chosen messages, while the verification key vk remains unchanged. The adversary's goal is to produce a valid forgery—a message–signature pair (m, σ)—such that σ verifies correctly under the corresponding verification key vk. If the adversary succeeds in producing such a forgery, it is said to have broken the scheme under related-key attacks. These transformations can include various functions, such as affine transformations, bitwise operations, or other structured modifications. A more detailed RKA security definition is given in Appendix B.

The key-shift security notion is a special case of RKA security, in which the adversary is restricted to applying only additive shifts to the secret key. That is, the adversary can choose a shift value Δsk and obtain signatures under $sk + \Delta sk$.

Our proposed key-shift* security strengthens the key-shift model by also allowing the adversary to submit not only a forged message–signature pair (m, σ), but also a corresponding secret key shift Δsk. The adversary succeeds if the signature σ verifies correctly with respect to the verification key associated with the shifted secret key $sk + \Delta sk$. This captures a stronger adversarial capability and provides a more refined security guarantee in settings where both the signing and verification keys may be subject to related-key manipulations.

Definition 15 (Key-Shift* Security)**.** *A signature scheme* $\mathsf{SIG} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *with a simple key-generation process* KG *is key-shift* secure if for all* PPT *adversaries* $\mathcal{A}$, *it holds that*

$$\mathsf{Adv}^{\mathsf{key-shift}^*}_{\mathsf{SIG},\mathcal{A}}(1^\lambda) := \Pr[\, \mathsf{Expt}^{\mathsf{key-shift}^*}_{\mathsf{SIG},\mathcal{A}}(1^\lambda) \Rightarrow 1 \,] \leqslant \mathsf{negl}(\lambda),$$

where $\mathsf{Expt}^{\mathsf{key-shift}^*}_{\mathsf{SIG},\mathcal{A}}(1^\lambda)$ *is defined as follows.*

$\mathsf{Expt}^{\mathsf{key-shift}^*}_{\mathsf{SIG},\mathcal{A}}(1^\lambda)$*:*

1. *The challenger* $\mathcal{C}$ *runs* $pp \leftarrow \mathsf{Setup}(1^\lambda)$, *generates* $(vk, sk) \leftarrow \mathsf{KeyGen}(pp)$, *initializes an empty query set* $\mathcal{Q} = \emptyset$, *and sends* pp *and* vk *to the adversary* $\mathcal{A}$.
2. *Throughout the experiment,* $\mathcal{A}$ *may adaptively query a signing oracle with pairs* $(m_j, \Delta sk_j)$, *where* $m_j \in \mathcal{M}$ *and* Δsk_j *is a key shift:*
   - $\mathcal{A}$ *submits* $(m_j, \Delta sk_j)$ *to* $\mathcal{C}$.
   - $\mathcal{C}$ *computes the signature* $\sigma_j \leftarrow \mathsf{Sign}(pp, sk + \Delta sk_j, m_j)$, *adds* m_j *to* $\mathcal{Q}$, *and returns* σ_j *to* $\mathcal{A}$.

3. *Finally, $\mathcal{A}$ outputs a forgery (m^*, σ^*) and a final shift Δsk^*. The experiment outputs* 1 *if, and only if,*

$$m^* \notin \mathcal{Q} \quad \text{and} \quad \mathsf{Verify}\big(pp, \mathsf{KG}(pp, sk + \Delta sk^*), m^*, \sigma^*\big) = 1.$$

Otherwise, it outputs 0.

Remark 4. *Compared to standard key-shift security, key-shift* security captures a stronger adversarial capability. In this work, key-shift* security is specifically designed to support the construction of fuzzy signature schemes, where secret keys are often derived from noisy or biometric data that may vary across different uses.*

Lemma 2. *Let* $\mathsf{SIG} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *be a signature scheme with a simple key-generation process* KG *and a signing key space $\mathcal{SK}$. If* SIG *is EUF-CMA secure and homomorphic, then* SIG *is key-shift* secure.*

Proof. We prove the key-shift* security of SIG via a reduction to its EUF-CMA security, leveraging the homomorphic property of the scheme SIG. Suppose there exists a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ that breaks the key-shift* security of SIG with a non-negligible advantage. Then, we construct a new adversary $\mathcal{B}$ that uses $\mathcal{A}$ as a subroutine to break the EUF-CMA security of SIG.

The adversary $\mathcal{B}$ interacts with its own EUF-CMA challenger and simulates the key-shift* experiment for $\mathcal{A}$ as follows:

1. Upon receiving the public parameters pp and verification key vk from its EUF-CMA challenger, $\mathcal{B}$ forwards (pp, vk) to $\mathcal{A}$.
2. For each j-th signing query $(m_j, \Delta sk_j)$ made by $\mathcal{A}$, $\mathcal{B}$ proceeds as follows:
   - $\mathcal{B}$ queries its signing oracle on m_j and receives a signature $\widetilde{\sigma_j} \leftarrow \mathsf{Sign}(pp, sk, m_j)$.
   - $\mathcal{B}$ computes $\sigma_j \leftarrow \mathsf{A}_{\mathsf{Sign}}(pp, m_j, \widetilde{\sigma_j}, \Delta sk_j)$ using the homomorphic signing algorithm, and returns σ_j to $\mathcal{A}$.
3. In the final phase, when $\mathcal{A}$ outputs a forgery (m^*, σ^*) and a shift Δsk^*, $\mathcal{B}$ computes

$$\widetilde{\sigma^*} \leftarrow \mathsf{A}_{\mathsf{Sign}}(pp, m^*, \sigma^*, -\Delta sk^*),$$

and submits $(m^*, \widetilde{\sigma^*})$ to its challenger. $\mathcal{B}$ then outputs whatever its challenger returns.

We now argue that the simulation is perfect. By the homomorphic property of SIG, we have

$$\mathsf{Sign}(pp, sk + \Delta sk_j, m_j) = \mathsf{A}_{\mathsf{Sign}}(pp, m_j, \widetilde{\sigma_j}, \Delta sk_j),$$

where $\widetilde{\sigma_j} = \mathsf{Sign}(pp, sk, m_j)$. Therefore, the signatures returned by $\mathcal{B}$ are distributed identically to those in the real key-shift* experiment.

Next, we will show that $\mathcal{B}$ has the same advantage of $\mathcal{A}$.

Note that if $\mathcal{A}$'s forgery (m^*, σ^*) is valid under the shifted key $sk + \Delta sk^*$, i.e.,

$$m^* \notin \mathcal{Q} \wedge \mathsf{Verify}(pp, \mathsf{KG}(pp, sk + \Delta sk^*), m^*, \sigma^*) = 1,$$

by the homomorphic property, we have

$$\widetilde{\sigma^*} = \mathsf{A}_{\mathsf{Sign}}(pp, m^*, \sigma^*, -\Delta sk^*) = \mathsf{Sign}(pp, sk + \Delta sk^* - \Delta sk^*, m^*) = \mathsf{Sign}(pp, sk, m^*).$$

This implies that $\widetilde{\sigma^*}$ is a valid signature of m^* under the signing key sk. As a result, if $\mathcal{A}$ wins the key-shift* experiment, then $\mathcal{B}$ wins the EUF-CMA experiment. So we have

$$\mathsf{Adv}^{\mathsf{key-shift}^*}_{\mathsf{SIG},\mathcal{A}}(1^\lambda) \leq \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathsf{SIG},\mathcal{B}}(1^\lambda) \leq \mathsf{negl}(\lambda).$$

□

3. Fuzzy Signature

A fuzzy signature scheme $\mathsf{SIG_F}$ differs from a traditional digital signature scheme SIG in that the signing key is no longer required to be uniformly random or precisely reproducible. Instead, the signing key in a fuzzy signature can be any high-entropy fuzzy data, such as biometric inputs (e.g., facial scans or fingerprints). The key-generation algorithm takes a fuzzy signing key $\mathsf{w} \in \mathcal{W}$ (e.g., an initial facial scan) and outputs a verification key vk_{F}. Any signature generated using $\mathsf{w}' \in \mathcal{W}$ (e.g., a later facial scan of the same person) will verify successfully under vk_{F}, provided that w and w' are sufficiently close according to an appropriate distance metric. Below, we formally define a fuzzy signature scheme $\mathsf{SIG_F}$.

Definition 16 (Fuzzy Signature)**.** *Let $\mathcal{W}$ be the fuzzy signing key space, $\mathcal{M}$ the message space, and $t \in \mathbb{N}$ a threshold parameter governing the acceptable distance between keys. A $(\mathcal{W}, \mathcal{M}, t)$-fuzzy signature scheme $\mathsf{SIG_F} = (\mathsf{Setup_F}, \mathsf{KeyGen_F}, \mathsf{Sign_F}, \mathsf{Verify_F})$ consists of four probabilistic polynomial-time (*PPT*) algorithms:*

- $pp_{\mathsf{F}} \leftarrow \mathsf{Setup_F}(1^\lambda)$*. Upon input of the security parameter 1^λ, this algorithm outputs public parameters pp_{F}.*
- $vk_{\mathsf{F}} \leftarrow \mathsf{KeyGen_F}(pp_{\mathsf{F}}, \mathsf{w})$*. Given the public parameters pp_{F} and a fuzzy signing key $\mathsf{w} \in \mathcal{W}$ (e.g., a biometric template), this algorithm generates a verification key vk_{F}.*
- $\sigma_{\mathsf{F}} \leftarrow \mathsf{Sign_F}(pp_{\mathsf{F}}, \mathsf{w}', m)$*. To sign a message $m \in \mathcal{M}$, the signer uses a (possibly different) fuzzy key $\mathsf{w}' \in \mathcal{W}$ close to the original w, along with pp_{F}, to produce a signature σ_{F}.*
- $b \leftarrow \mathsf{Verify_F}(pp_{\mathsf{F}}, vk_{\mathsf{F}}, m, \sigma_{\mathsf{F}})$*, where $b \in \{0,1\}$. This algorithm checks the validity of the signature σ_{F} on message m under the verification key vk_{F} and public parameters pp_{F}, returning* 1 *if valid and* 0 *otherwise.*

Correctness. For any $m \in \mathcal{M}$, $\mathsf{w}, \mathsf{w}' \in \mathcal{W}$, if $pp_{\mathsf{F}} \leftarrow \mathsf{Setup_F}(1^\lambda)$, $vk_{\mathsf{F}} \leftarrow \mathsf{KeyGen_F}(pp_{\mathsf{F}}, \mathsf{w})$, $\sigma_{\mathsf{F}} \leftarrow \mathsf{Sign}(pp_{\mathsf{F}}, \mathsf{w}', m)$, $\mathsf{dis}(\mathsf{w}, \mathsf{w}') \leq t$, we have

$$\Pr[\mathsf{Verify}(pp_{\mathsf{F}}, vk_{\mathsf{F}}, m, \sigma_{\mathsf{F}}) = 1] \geq 1 - \mathsf{negl}(\lambda).$$

Next, we present a new security model in which the adversary can adaptively determine the differences between biometric samples.

Definition 17 ($\mathfrak{m}$-EUF-CMA* Security)**.** *A $(\mathcal{W}, \mathcal{M}, t)$-fuzzy signature scheme $\mathsf{SIG_F}$ is said to be* $\mathfrak{m}$-existentially unforgeable under adaptive chosen-message attacks secure *($\mathfrak{m}$-EUF-CMA*) if no efficient adversary $\mathcal{A}$ can produce a valid forgery with non-negligible probability when the signing key is sampled from an arbitrary distribution W over $\mathcal{W}$ with min-entropy of at least $\mathfrak{m}$, i.e., $H_\infty(W) \geq \mathfrak{m}$.*

Formally, for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$, we require that

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

where $\mathsf{Expt}^{\mathfrak{m}\text{-EUF}-\mathsf{CMA}^*}_{\mathsf{SIG},\mathcal{A}}(1^\lambda)$ is defined as follows.

$\mathsf{Expt}^{\mathfrak{m}\text{-EUF}-\mathsf{CMA}^*}_{\mathsf{SIG},\mathcal{A}}(1^\lambda)$*:*

1. *The challenger $\mathcal{C}$ runs $pp_{\mathsf{F}} \leftarrow \mathsf{Setup_F}(1^\lambda)$; samples a signing key $\mathsf{w} \leftarrow W$, where W is a distribution over $\mathcal{W}$ with $H_\infty(W) \geq \mathfrak{m}$; computes $vk_{\mathsf{F}} \leftarrow \mathsf{KenGen_F}(pp_{\mathsf{F}}, \mathsf{w})$; and initializes an empty query set $\mathcal{Q} = \emptyset$. Next, it returns pp_{F} and vk_{F} to $\mathcal{A}$.*

2. *Over the course of the experiment, $\mathcal{A}$ is allowed to adaptively make signing oracle queries in the following form:*
   - *$\mathcal{A}$ sends a message $m_j \in \mathcal{M}$ and a shift $\Delta \mathsf{w}_j \in \mathcal{W}$ to $\mathcal{C}$.*
   - *If $\mathsf{dis}(\Delta \mathsf{w}_j, 0) > t$, $\mathcal{C}$ returns $\perp$ (invalid shift). Else, $\mathcal{C}$ invokes $\sigma_{\mathsf{F},j} \leftarrow \mathsf{Sign}_{\mathsf{F}}(pp_{\mathsf{F}}, \mathsf{w} + \Delta \mathsf{w}_j, m_j)$, adds m_j to the query set $\mathcal{Q}$, and returns $\sigma_{\mathsf{F},j}$ to $\mathcal{A}$.*
3. *In the final phase, $\mathcal{A}$ outputs a forgery $(m^*, \sigma_{\mathsf{F}}^*)$. The experiment returns 1 if $m^* \notin \mathcal{Q} \wedge \mathsf{Verify}_{\mathsf{F}}(pp, vk, m^*, \sigma_{\mathsf{F}}^*) = 1$; otherwise, it returns 0.*

Remark 5. *In our security model, the shift $\Delta \mathsf{w}_j$ between different samples of the same noisy source is adaptively chosen by the adversary $\mathcal{A}$. Note that, before the adversary submits a shift Δw_j, the adversary has received $pp_{\mathsf{F}}, vk_{\mathsf{F}}$, and $(\sigma_{\mathsf{F},1}, \cdots, \sigma_{\mathsf{F},j-1})$. Since vk_{F} and $(\sigma_{\mathsf{F},1}, \cdots, \sigma_{\mathsf{F},j-1})$ are dependent on W, Δw_j can be dependent on W. However, due to $\mathcal{A}$'s limited computational ability, the shift $\Delta \mathsf{w}_j$ cannot depend on the biometric data in an arbitrary manner. Nevertheless, our security model is still stronger than the security model in [12], which requires the errors between different samples to be independent of the noisy source.*

Remark 6. *From a theoretical perspective, the adversary-selected shift model subsumes the scenario of independent errors, provided that the adversary chooses the shifts randomly and independently. From a practical perspective, however, we argue that modeling only independent errors may not be sufficient to capture real-world threats. For instance, in biometric-based systems, an adversary could manipulate the sampling device to deterministically flip or fix certain bits of the biometric data. Consider, for example, the i-th bit of the biometric template:*

- *If the original bit is 1, it remains unchanged.*
- *If the original bit is 0, it is flipped to 1.*

This type of deterministic bit manipulation cannot be captured by independent random-error models, as the error is not random, but depends on the biometric template. In contrast, our adversary-selected shift model is specifically designed to account for such scenarios, thereby providing a stronger and more realistic security guarantee in the presence of active and targeted distortions.

4. Construction of Fuzzy Signature

4.1. Construction

Our fuzzy signature scheme $\mathsf{SIG}_{\mathsf{F}} = (\mathsf{Setup}_{\mathsf{F}}, \mathsf{KeyGen}_{\mathsf{F}}, \mathsf{Sign}_{\mathsf{F}}, \mathsf{Verify}_{\mathsf{F}})$, which is depicted in Figure 2, is composed of the following building blocks.

- A homomorphic $(\mathcal{W}, \mathfrak{m}, \widetilde{\mathfrak{m}}, t)$-secure sketch $\mathsf{SS} = (\mathsf{SS.Gen}, \mathsf{SS.Rec})$ with a sketching space $\mathcal{S}$ that satisfies the error-recoverable property.
- A key-shift* secure and homomorphic $\mathsf{SIG} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ with a secret key space $\mathcal{SK}$ and a message space $\mathcal{M} \times \mathcal{S}$.
- A homomorphic average-case $(\mathcal{W}, \widetilde{\mathfrak{m}}, \mathcal{SK}, \varepsilon_{ext})$-strong extractor Ext with seed space $\mathcal{I}$.

$\underline{\mathsf{Setup}_{\mathsf{F}}(1^\lambda)}$:	$\underline{\mathsf{Sign}_{\mathsf{F}}(pp_{\mathsf{F}}, \mathsf{w}', m)}$:	$\underline{\mathsf{Verify}_{\mathsf{F}}(pp_{\mathsf{F}}, vk_{\mathsf{F}}, m, \sigma_{\mathsf{F}})}$:
$i \leftarrow_{\$} \mathcal{I}$, $pp \leftarrow \mathsf{Setup}(1^\lambda)$.	Parse $pp_{\mathsf{F}} = (i, pp)$.	Parse $pp_{\mathsf{F}} = (i, pp)$.
$pp_{\mathsf{F}} \leftarrow (i, pp)$.	$\mathsf{s}' \leftarrow \mathsf{SS.Gen}(\mathsf{w}')$.	Parse $vk_{\mathsf{F}} = (\mathsf{s}, vk)$, $\sigma_{\mathsf{F}} = (\mathsf{s}', \sigma)$.
Return pp_{F}.	$sk' \leftarrow \mathsf{Ext}(\mathsf{w}', i)$.	$M \leftarrow (m, \mathsf{s}')$.
$\underline{\mathsf{KeyGen}_{\mathsf{F}}(pp_{\mathsf{F}}, \mathsf{w})}$:	$M \leftarrow (m, \mathsf{s}')$.	$\Delta \mathsf{w} \leftarrow f_{rec}(\mathsf{s}, \mathsf{s}')$.
Parse $pp_{\mathsf{F}} = (i, pp)$.	$\sigma \leftarrow \mathsf{Sign}(pp, sk', M)$.	$\Delta sk \leftarrow \mathsf{Ext}(\Delta \mathsf{w}, i)$.
$\mathsf{s} \leftarrow \mathsf{SS.Gen}(\mathsf{w})$, $sk \leftarrow \mathsf{Ext}(\mathsf{w}, i)$.	$\sigma_{\mathsf{F}} \leftarrow (\mathsf{s}', \sigma)$.	$vk' \leftarrow \mathsf{A}_{\mathsf{vk}}(pp, vk, \Delta sk)$.
$vk \leftarrow \mathsf{KG}(pp, sk)$, $vk_{\mathsf{F}} \leftarrow (\mathsf{s}, vk)$.	Return σ_{F}.	$b \leftarrow \mathsf{Verify}(pp, vk', M, \sigma)$.
Return vk_{F}.		Return b.

Figure 2. Generic construction of fuzzy signature scheme $\mathsf{SIG}_{\mathsf{F}}$.

For better clarity and understanding of our construction, we have added Figure 3 to visually depict the key components and their interactions.

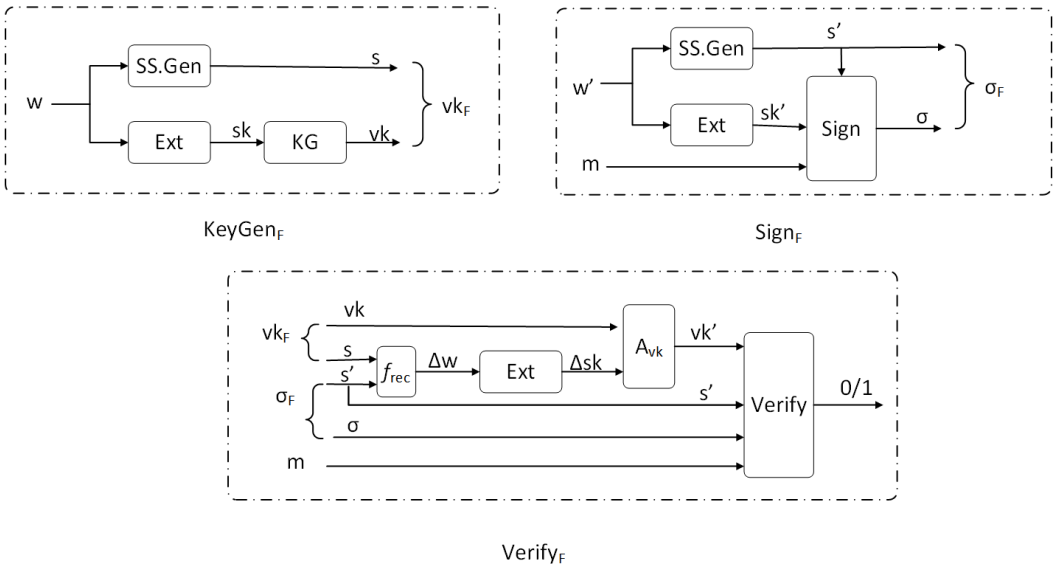

Figure 3. An overview of our construction of fuzzy signature. We have omitted pp for brevity.

4.2. Correctness

Theorem 1. *If* SS *is a homomorphic* $(\mathcal{W}, \mathfrak{m}, \widetilde{\mathfrak{m}}, t)$*-secure sketch with a sketch space* $\mathcal{S}$ *and the error-recoverable property,* Ext *is a homomorphic average-case* $(\mathcal{W}, \widetilde{\mathfrak{m}}, \mathcal{SK}, \varepsilon_{ext})$*-strong extractor with a seed space* $\mathcal{I}$*, and the signature scheme* SIG *is homomorphic with the secret key space* $\mathcal{SK}$ *and message space* $\mathcal{M} \times \mathcal{S}$*, then the constructed fuzzy signature scheme* $\mathsf{SIG_F}$ *is correct.*

Proof. To establish the correctness of the fuzzy signature scheme, we need to show that for any $m \in \mathcal{M}$ and $\mathsf{w}, \mathsf{w}' \in \mathcal{W}$, if $\mathsf{dis}(\mathsf{w}, \mathsf{w}') \leq t$, the verification algorithm accepts the signature. Specifically, it holds that $\mathsf{Verify_F}(pp_\mathsf{F}, vk_\mathsf{F}, m, \sigma_\mathsf{F}) = 1$, where $pp_\mathsf{F} \leftarrow \mathsf{Setup_F}(1^\lambda)$, $vk_\mathsf{F} \leftarrow \mathsf{KeyGen_F}(pp_\mathsf{F}, \mathsf{w})$, and $\sigma_\mathsf{F} \leftarrow \mathsf{Sign_F}(pp_\mathsf{F}, \mathsf{w}', m)$.

Note that verification algorithm accepts the signature if, and only if, the underlying signature scheme successfully verifies. That is, $\mathsf{Verify}(pp, vk', M, \sigma) = 1$, where $\sigma \leftarrow \mathsf{Sign}(pp, sk', M)$, $\mathsf{s}' \leftarrow \mathsf{SS.Gen}(\mathsf{w}')$, $sk' \leftarrow \mathsf{Ext}(\mathsf{w}', i)$, and $M \leftarrow (m, \mathsf{s}')$.

By the correctness of the underlying signature SIG, if $vk' = \mathsf{KG}(pp, sk')$, then $\mathsf{Verify}(pp, vk', M, \sigma) = 1$, since $\sigma \leftarrow \mathsf{Sign}(pp, sk', M)$. So we need to prove that if $\mathsf{dis}(\mathsf{w}, \mathsf{w}') \leq t$, $vk' = \mathsf{KG}(pp, sk')$.

Recall that $\Delta\mathsf{w} \leftarrow f_{rec}(\mathsf{s}, \mathsf{s}')$, $\Delta sk \leftarrow \mathsf{Ext}(\Delta\mathsf{w}, i)$, $vk' \leftarrow \mathsf{A_{vk}}(pp, vk, \Delta sk)$, where $\mathsf{s} \leftarrow \mathsf{SS.Gen}(\mathsf{w})$ and $\mathsf{s}' \leftarrow \mathsf{SS.Gen}(\mathsf{w}')$. Due to the the error-recoverable property of the secure sketch SS, for any $\mathsf{w}, \mathsf{w}' \in \mathcal{W}$, if $\mathsf{dis}(\mathsf{w}, \mathsf{w}') \leq t$, it holds that $\Delta\mathsf{w} = \mathsf{w}' - \mathsf{w} = f_{rec}(\mathsf{s}, \mathsf{s}')$.

Due to the homomorphic property of Ext, we have

$$sk' = \mathsf{Ext}(\mathsf{w}') = \mathsf{Ext}(\mathsf{w} + \Delta\mathsf{w}) = \mathsf{Ext}(\mathsf{w}) + \mathsf{Ext}(\Delta\mathsf{w}) = sk + \Delta sk.$$

Similarly, due to the homomorphic property of SIG, it follows that

$$vk' = \mathsf{A_{vk}}(pp, vk, \Delta sk) = \mathsf{KG}(pp, sk + \Delta sk) = \mathsf{KG}(pp, sk').$$

As a result, the fuzzy signature scheme $\mathsf{SIG_F}$ is correct. □

4.3. Security

Theorem 2. *If* SS *is a homomorphic* $(\mathcal{W}, \mathfrak{m}, \widetilde{\mathfrak{m}}, t)$*-secure sketch with a sketching space* $\mathcal{S}$ *and it satisfies the error-recoverable property,* Ext *is a homomorphic average-case* $(\mathcal{W}, \widetilde{\mathfrak{m}}, \mathcal{SK}, \varepsilon_{ext})$*-strong extractor with a seed space* $\mathcal{I}$*,* SIG *is a homomorphic signature scheme with a secret key space* $\mathcal{SK}$*, message space* $\mathcal{M} \times \mathcal{S}$ *and key-shift* security, then the construction of fuzzy signature* $\mathsf{SIG_F}$ *in Figure 2 is a* $(\mathcal{W}, \mathcal{M}, t)$*-fuzzy signature satisfying* $\mathfrak{m}$*-EUF-CMA* security. In addition,*

$$\mathsf{Adv}^{\mathfrak{m}\text{-}\mathsf{EUF}-\mathsf{CMA}^*}_{\mathsf{SIG_F},\mathcal{A}}(1^\lambda) \leq \mathsf{Adv}^{\mathsf{key}-\mathsf{shift}^*}_{\mathsf{SIG}}(1^\lambda) + \varepsilon_{ext}.$$

Proof. Let $\mathcal{A}$ be an arbitrary PPT adversary that attacks the $\mathfrak{m}$-EUF-CMA* security of fuzzy signature $\mathsf{SIG_F}$. We will prove this theorem by a sequence of games, where the first game $\mathsf{Game\ 0}$ is the original game $\mathsf{Expt}^{\mathfrak{m}\text{-}\mathsf{EUF}-\mathsf{CMA}^*}_{\mathsf{SIG_F},\mathcal{A}}(1^\lambda)$. For $\mathsf{i} \in \{0,1,2\}$, let $\mathsf{S_i}$ denote the event that $\mathcal{A}$ wins (i.e., the experiment returns 1) in $\mathsf{Game\ i}$. Our goal is to show that $\mathsf{Adv}^{\mathfrak{m}\text{-}\mathsf{EUF}-\mathsf{CMA}^*}_{\mathsf{SIG_F},\mathcal{A}}(1^\lambda) := \Pr[\mathsf{Expt}^{\mathfrak{m}\text{-}\mathsf{EUF}-\mathsf{CMA}^*}_{\mathsf{SIG_F},\mathcal{A}}(1^\lambda) \Rightarrow 1]$ is negligible. Differences between adjacent games are highlighted with underline.

Game 0. $\mathsf{Game\ 0}$ is identical to the $\mathfrak{m}$-EUF-CMA* experiment $\mathsf{Expt}^{\mathfrak{m}\text{-}\mathsf{EUF}\text{-}\mathsf{CMA}^*}_{\mathsf{SIG_F},\mathcal{A}}(1^\lambda)$. Specifically, it proceeds as follows:

1. Challenger $\mathcal{C}$ samples $i \leftarrow_{\$} \mathcal{I}$, invokes $pp \leftarrow \mathsf{Setup}(1^\lambda)$, sets $pp_{\mathsf{F}} \leftarrow (i, pp)$, then samples $\mathsf{w} \leftarrow W$, computes $\mathsf{s} \leftarrow \mathsf{SS.Gen}(\mathsf{w})$, $sk \leftarrow \mathsf{Ext}(\mathsf{w}, i)$, $vk \leftarrow \mathsf{KG}(pp, sk)$, and sets $vk_{\mathsf{F}} \leftarrow (\mathsf{s}, vk)$. Next, it initializes the query set $\mathcal{Q} = \emptyset$, which denotes the set of queries selected by $\mathcal{A}$. Afterwards, $\mathcal{C}$ returns $(pp_{\mathsf{F}}, vk_{\mathsf{F}})$ to $\mathcal{A}$.
2. The adversary $\mathcal{A}$ may adaptively make signing queries.
   - $\mathcal{A}$ submits a message $m_j \in \mathcal{M}$ and a shift $\Delta\mathsf{w}_j \in \mathcal{W}$ to $\mathcal{C}$.
   - If $\mathsf{dis}(\Delta\mathsf{w}_j, 0) > t$, $\mathcal{C}$ returns $\perp$. Else, $\mathcal{C}$ parses $pp_{\mathsf{F}} = (\mathsf{i}, pp)$, invokes $\mathsf{s}_j \leftarrow \mathsf{SS.Gen}(\mathsf{w} + \Delta\mathsf{w}_j)$ and $sk_j \leftarrow \mathsf{Ext}(\mathsf{w} + \Delta\mathsf{w}_j, i)$, then sets $M_j \leftarrow (m_j, \mathsf{s}_j)$, invokes $\sigma_j \leftarrow \mathsf{Sign}(pp, sk_j, M_j)$, sets $\sigma_{\mathsf{F},j} \leftarrow (\mathsf{s}_j, \sigma_j)$, adds m_j to the set $\mathcal{Q}$, and returns $\sigma_{\mathsf{F},j}$ to $\mathcal{A}$.
3. Finally, $\mathcal{A}$ submits a forgery $(m^*, \sigma^*_{\mathsf{F}})$. $\mathcal{C}$ parses $pp_{\mathsf{F}} = (i, pp)$, $vk_{\mathsf{F}} = (\mathsf{s}, vk)$, $\sigma^*_{\mathsf{F}} = (\mathsf{s}^*, \sigma^*)$, then sets $M^* \leftarrow (m^*, \mathsf{s}^*)$, and computes $\Delta\mathsf{w}^* \leftarrow f_{rec}(\mathsf{s}^*, \mathsf{s})$, $\Delta sk^* \leftarrow \mathsf{Ext}(\Delta\mathsf{w}^*, i)$ and $vk^* \leftarrow \mathsf{A_{vk}}(pp, vk, \Delta sk^*)$. If $m^* \notin \mathcal{Q} \wedge \mathsf{Verify}(pp, vk^*, M^*, \sigma^*) = 1$, the experiment outputs 1; else, it outputs 0.

   Obviously,

$$\mathsf{Adv}^{\mathfrak{m}\text{-}\mathsf{EUF}-\mathsf{CMA}^*}_{\mathsf{SIG_F},\mathcal{A}}(1^\lambda) = \Pr[\mathsf{Expt}^{\mathfrak{m}\text{-}\mathsf{EUF}-\mathsf{CMA}^*}_{\mathsf{SIG_F},\mathcal{A}}(1^\lambda) \Rightarrow 1] = \Pr[\mathsf{S}_0]. \tag{2}$$

Game 1. This game behaves equivalently to $\mathsf{Game\ 0}$, apart from some conceptual changes.

More precisely,

1. The adversary $\mathcal{A}$ may adaptively make signing queries.
   - $\mathcal{A}$ submits a message $m_j \in \mathcal{M}$ and a shift $\Delta\mathsf{w}_j \in \mathcal{W}$ to $\mathcal{C}$.
   - If $\mathsf{dis}(\Delta\mathsf{w}_j, 0) > t$, $\mathcal{C}$ returns $\perp$. Else, $\mathcal{C}$ parses $pp_{\mathsf{F}} = (\mathsf{i}, pp)$, invokes $\underline{\mathsf{s}_j \leftarrow \mathsf{s} + \mathsf{SS.Gen}(\Delta\mathsf{w}_j)}$ and $\underline{sk_j \leftarrow sk + \mathsf{Ext}(\Delta\mathsf{w}_j, i)}$, then sets $M_j \leftarrow (m_j, \mathsf{s}_j)$, invokes $\sigma_j \leftarrow \mathsf{Sign}(pp, sk_j, M_j)$, sets $\sigma_{\mathsf{F},j} \leftarrow (\mathsf{s}_j, \sigma_j)$, adds m_j to the set $\mathcal{Q}$, and returns $\sigma_{\mathsf{F},j}$ to $\mathcal{A}$.

Lemma 3. $\Pr[\mathsf{S}_0] = \Pr[\mathsf{S}_1]$.

Proof. Due to the homomorphic property of Ext, it follows that

$$sk_j = \mathsf{Ext}(\mathsf{w} + \Delta\mathsf{w}_j, i) = \mathsf{Ext}(\mathsf{w}, i) + \mathsf{Ext}(\Delta\mathsf{w}_j, i) = sk + \mathsf{Ext}(\Delta\mathsf{w}_j, i).$$

Similarly, due to the homomorphic property of SS, it follows that

$$\mathsf{s}_j = \mathsf{SS.Gen}(\mathsf{w} + \Delta\mathsf{w}_j) = \mathsf{SS.Gen}(\mathsf{w}) + \mathsf{SS.Gen}(\Delta\mathsf{w}_j) = \mathsf{s} + \mathsf{SS.Gen}(\Delta\mathsf{w}_j).$$

Therefore, in both games, the signing key sk_j and the sketch s_j used in response to each query are computed correctly according to the homomorphic property. Since all oracle responses are identically distributed in Game_0 and Game_1, it follows that $\Pr[\mathsf{S}_0] = \Pr[\mathsf{S}_1]$. The lemma is thus established. □

Game 2. This game is identical to Game 1, except that in step 1, sk is uniformly chosen from $\mathcal{SK}$ other than $sk \leftarrow \mathsf{Ext}(\mathsf{w}, i)$. More specifically,

1. Challenger $\mathcal{C}$ samples $i \leftarrow_{\$} \mathcal{I}$, invokes $pp \leftarrow \mathsf{Setup}(1^\lambda)$, sets $pp_{\mathsf{F}} \leftarrow (i, pp)$, then samples $\mathsf{w} \leftarrow W$, computes $\mathsf{s} \leftarrow \mathsf{SS.Gen}(\mathsf{w})$, randomly chooses $sk \leftarrow_{\$} \mathcal{SK}$, computes $vk \leftarrow \mathsf{KG}(pp, sk)$, and sets $vk_{\mathsf{F}} \leftarrow (\mathsf{s}, vk)$. Next, it initializes the query set $\mathcal{Q} = \emptyset$, which denotes the set of queries selected by $\mathcal{A}$. Afterwards, $\mathcal{C}$ returns $(pp_{\mathsf{F}}, vk_{\mathsf{F}})$ to $\mathcal{A}$.

Lemma 4. $|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]| \leq \varepsilon_{ext}$.

Proof. Note that $H_\infty(W) \geq \mathfrak{m}$; by the security of the $(\mathcal{W}, \mathfrak{m}, \widetilde{\mathfrak{m}}, t)$-secure sketch, we have $\widetilde{H}_\infty(W|\mathsf{SS}(W) \geq \widetilde{\mathfrak{m}}$. Since Ext is a average-case $(\mathcal{W}, \widetilde{\mathfrak{m}}, \mathcal{SK}, \varepsilon_{ext})$-strong extractor with a seed space $\mathcal{I}$, then we have

$$\mathrm{SD}((\mathsf{Ext}(\mathsf{w}, i), i, \mathsf{s} = \mathsf{SS.Gen}(\mathsf{w})), (u, i, \mathsf{s} = \mathsf{SS.Gen}(\mathsf{w}))) \leq \varepsilon_{ext},$$

where $u \leftarrow_{\$} \mathcal{SK}$.

Next we show that if there exists an adversary $\mathcal{A}$ that the difference between the probability of $\mathcal{A}$ winning in Game 1 and the probability of $\mathcal{A}$ winning in Game 2 is non-negligible, then we can construct an algorithm $\mathcal{B}$ that can distinguish $(\mathsf{Ext}(\mathsf{w}, i), i, \mathsf{s} = \mathsf{SS.Gen}(\mathsf{w}))$ and $(u, i, \mathsf{s} = \mathsf{SS.Gen}(\mathsf{w}))$ with a non-negligible advantage. Assume that $\mathcal{B}$ is given (d, i, s), where d is either $\mathsf{Ext}(\mathsf{w}, i)$ or an element $u \leftarrow_{\$} \mathcal{SK}$. The aim of $\mathcal{B}$ is to tell which case it is. Adversary $\mathcal{B}$ proceeds as follows:

1. $\mathcal{B}$ invokes $pp \leftarrow \mathsf{Setup}(1^\lambda)$, sets $pp_{\mathsf{F}} \leftarrow (i, pp)$, then sets $sk = d$, computes $vk \leftarrow \mathsf{KG}(pp, sk)$, and sets $vk_{\mathsf{F}} \leftarrow (\mathsf{s}, vk)$. Next, it initializes the query set $\mathcal{Q} = \emptyset$, which denotes the set of queries selected by $\mathcal{A}$. Afterwards, $\mathcal{C}$ returns $(pp_{\mathsf{F}}, vk_{\mathsf{F}})$ to $\mathcal{A}$.
2. $\mathcal{A}$ may adaptively interact with the signing oracle in the following manner.
   - $\mathcal{A}$ submits a message $m_j \in \mathcal{M}$ and a shift $\Delta\mathsf{w}_j \in \mathcal{W}$ to $\mathcal{B}$.
   - If $\mathsf{dis}(\Delta\mathsf{w}_j, 0) > t$, $\mathcal{B}$ returns $\perp$. Else, $\mathcal{B}$ parses $pp_{\mathsf{F}} = (i, pp)$, invokes $\mathsf{s}_j \leftarrow \mathsf{s} + \mathsf{SS.Gen}(\Delta\mathsf{w}_j)$ and $sk_j \leftarrow sk + \mathsf{Ext}(\Delta\mathsf{w}_j, i)$, then sets $M_j \leftarrow (m_j, \mathsf{s}_j)$, invokes $\sigma_j \leftarrow \mathsf{Sign}(pp, sk_j, M_j)$, sets $\sigma_{\mathsf{F},j} \leftarrow (\mathsf{s}_j, \sigma_j)$, adds m_j to the set $\mathcal{Q}$, and returns $\sigma_{\mathsf{F},j}$ to $\mathcal{A}$.
3. In the finalization phase, $\mathcal{A}$ outputs a forgery $(m^*, \sigma^*_{\mathsf{F}})$. $\mathcal{B}$ parses $pp_{\mathsf{F}} = (i, pp)$, $vk_{\mathsf{F}} = (\mathsf{s}, vk)$, $\sigma^*_{\mathsf{F}} = (\mathsf{s}^*, \sigma^*)$, then sets $M^* \leftarrow (m^*, \mathsf{s}^*)$, and computes $\Delta\mathsf{w}^* \leftarrow f_{rec}(\mathsf{s}^*, \mathsf{s})$, $\Delta sk^* \leftarrow \mathsf{Ext}(\Delta\mathsf{w}^*, i)$, and $vk^* \leftarrow \mathsf{A}_{\mathsf{vk}}(pp, vk, \Delta sk^*)$. If $m^* \notin \mathcal{Q} \wedge \mathsf{Verify}(pp, vk^*, M^*, \sigma^*) = 1$, $\mathcal{B}$ returns 1; else, $\mathcal{B}$ returns 0.

If $d = \mathsf{Ext}(\mathsf{w}, i)$, then $\mathcal{B}$ accurately simulates Game 1 for $\mathcal{A}$; if $d = u$, where $u \leftarrow_{\$} \mathcal{SK}$, then $\mathcal{B}$ perfectly simulates Game 2 for $\mathcal{A}$. As a result, we have

$$|\Pr[\mathsf{S}_1] - \Pr[\mathsf{S}_2]| \leq \mathrm{SD}((\mathsf{Ext}(\mathsf{w}, i), i, \mathsf{s}), (u, i, \mathsf{s})) \leq \varepsilon_{ext}.$$

The lemma follows. □

Lemma 5. $|\Pr[S_2]| \leq \mathsf{Adv}^{\mathsf{key-shift}^*}_{\mathsf{SIG}}(1^\lambda)$.

Proof. We prove this lemma by showing that if there exists a PPT adversary $\mathcal{A}$ who wins in Game 2, then we can construct a PPT adversary $\mathcal{B}$ who can win in the key-shift* experiment of the underlying signature SIG.

1. Upon receiving (pp, vk) from its key-shift* challenger, $\mathcal{B}$ samples $i \leftarrow_\$ \mathcal{I}$, $\mathsf{w} \leftarrow W$, then calculates $\mathsf{s} = \mathsf{SS.Gen}(\mathsf{w})$, sets $pp_\mathsf{F} \leftarrow (i, pp)$, $vk_\mathsf{F} \leftarrow (\mathsf{s}, vk)$, and returns $(pp_\mathsf{F}, vk_\mathsf{F})$ to adversary $\mathcal{A}$.
2. When $\mathcal{A}$ makes a signing query on $(m_j, \Delta\mathsf{w}_j) \in \mathcal{M} \times \mathcal{W}$, $\mathcal{B}$ responds to $\mathcal{A}$'s query in the following manner:
   - If $\mathsf{dis}(\Delta\mathsf{w}_j, 0) > t$, $\mathcal{B}$ returns $\perp$. Else, $\mathcal{B}$ computes $\Delta sk_j \leftarrow \mathsf{Ext}(\Delta\mathsf{w}_j, i)$ and $\mathsf{s}_j = \mathsf{s} + \mathsf{SS.Gen}(\Delta\mathsf{w}_j)$, sets $M_j = (m_j, \mathsf{s}_j)$, and sends $(M_j, \Delta sk_j)$ to its key-shift* challenger.
   - Upon receiving σ_j from its key-shift* challenger, $\mathcal{B}$ sets $\sigma_{\mathsf{F},j} = (\mathsf{s}_j, \sigma_j)$ and returns $\sigma_{\mathsf{F},j}$ to $\mathcal{A}$.
3. In the finalization phase, $\mathcal{A}$ outputs a forgery $(m^*, \sigma^*_\mathsf{F})$ to $\mathcal{B}$. $\mathcal{B}$ parses $\sigma^*_\mathsf{F} = (\mathsf{s}^*, \sigma^*)$, sets $M^* \leftarrow (m^*, \mathsf{s}^*)$, computes $\Delta\mathsf{w}^* \leftarrow f_{rec}(\mathsf{s}^*, \mathsf{s})$ and $\Delta sk^* \leftarrow \mathsf{Ext}(\Delta\mathsf{w}^*, i)$, submits (M^*, σ^*) and a shift Δsk^* to its challenger, and returns what its challenger returns.

It is obvious that $\mathcal{B}$ perfectly simulates Game 2 for $\mathcal{A}$. As a result, if $\mathcal{A}$ wins in Game 2, then $\mathcal{B}$ wins the key-shift* experiment. Consequently,

$$|\Pr[S_2]| \leq \mathsf{Adv}^{\mathsf{key-shift}^*}_{\mathsf{SIG}}(1^\lambda).$$

□

Taking Equation (2) and Lemmas 3–5 together, we have

$$\mathsf{Adv}^{\mathsf{m\text{-}EUF-CMA}^*}_{\mathsf{SIG}_\mathsf{F},\mathcal{A}}(1^\lambda) \leq \mathsf{Adv}^{\mathsf{key-shift}^*}_{\mathsf{SIG}}(1^\lambda) + \varepsilon_{ext}.$$

Theorem 2 follows. □

5. Instantiation

In this section, we instantiate the generic construction presented in Section 4 using the Waters signature scheme. We choose the Waters signature scheme for this instantiation as it is one of the few schemes that simultaneously supports both the homomorphic property and key-shift* security—two desirable features that are seldom found together in existing signature schemes. To this end, we first recall $\mathsf{SIG}_\mathsf{Wat}$, as depicted in Figure 4. Subsequently, we prove that $\mathsf{SIG}_\mathsf{Wat}$ is both homomorphic and key-shift* secure.

Waters [32] proved that the Waters signature scheme $\mathsf{SIG}_\mathsf{Wat}$ is EUF-CMA secure under the computational Diffie-Hellman (CDH) assumption in $\mathbb{G}$ (Further details regarding $\mathbb{G}$ can be found in Appendix C).

Theorem 3 ([32,33]). *Suppose there exists a forger $\mathcal{A}$ that (t, q, ϵ)-breaks the EUF-CMA security of $\mathsf{SIG}_\mathsf{Wat}$. Then, there exists an algorithm $\mathcal{B}$ (ϵ', t')-solving the computational Diffie–Hellman problem in $\mathbb{G}$ in time $t' \approx t$ with success probability $\epsilon' \geq \epsilon \cdot O\left(\frac{1}{\sqrt{\ell q}}\right)$.*

Lemma 6. $\mathsf{SIG}_\mathsf{Wat}$ *is homomorphic. More precisely,* $\mathsf{A}_\mathsf{vk}(pp, vk, \Delta sk) = vk \cdot g^{\Delta sk}$, $\mathsf{A}_\mathsf{Sign}(pp, m, (\sigma_1, \sigma_2), \Delta sk) = (h^{\Delta sk} \cdot \sigma_1, \sigma_2)$.

$\underline{\mathsf{Setup}_{\mathsf{Wat}}(1^\lambda)}$:	$\underline{\mathsf{KeyGen}_{\mathsf{Wat}}(pp)}$:
$\mathcal{BG} := (p, \mathbb{G}, \mathbb{G}_T, g, e) \leftarrow \mathsf{BGGen}(1^\lambda)$.	$sk \leftarrow_\$ \mathbb{Z}_p$.
$h, u', u_1, \ldots, u_l \leftarrow_\$ \mathbb{G}$.	$vk \leftarrow g^{sk}$.
$pp \leftarrow (\mathcal{BG}, h, u', (u_i)_{i\in[l]})$.	Return (vk, sk).
Return pp.	
$\underline{\mathsf{Sign}_{\mathsf{Wat}}(pp, sk, M)}$:	$\underline{\mathsf{Verify}_{\mathsf{Wat}}(pp, vk, M, \sigma)}$:
Parse M as $(M_1 \parallel \cdots \parallel M_l) \in \{0,1\}^l$.	$(\sigma_1, \sigma_2) \leftarrow \sigma$.
$r \leftarrow_\$ \mathbb{Z}_p$.	Parse M as $(M_1 \parallel \cdots \parallel M_l) \in \{0,1\}^l$.
$\sigma_1 \leftarrow h^{sk} \cdot (u' \cdot \prod_{i\in[l]} u_i^{M_i})^r$.	If $e(\sigma_2, u' \cdot \prod_{i\in[l]} u_i^{M_i}) \cdot e(vk, h) = e(\sigma_1, g)$.
$\sigma_2 \leftarrow g^r$.	Then return 1 else return 0.
Return $\sigma \leftarrow (\sigma_1, \sigma_2)$.	

Figure 4. Waters signature scheme $\mathsf{SIG}_{\mathsf{Wat}}$.

Proof. We now show that $\mathsf{SIG}_{\mathsf{Wat}}$ is homomorphic by demonstrating that it satisfies Definition 14. It is straightforward to see that $\mathsf{SIG}_{\mathsf{Wat}}$ has a simple key-generation process,

$$vk = \mathsf{KG}(pp, sk) = g^{sk}.$$

1. Note that $sk \leftarrow_\$ \mathbb{Z}_p$. The signing key space $\mathcal{SK} = \mathbb{Z}_p$, equipped with addition modulo p, constitutes an abelian group.
2. Note that $\mathsf{KG}(pp, sk + \Delta sk) = g^{sk+\Delta sk} = g^{sk} \cdot g^{\Delta sk} = vk \cdot g^{\Delta sk}$, so there exsits a deterministic and efficient algorithm A_{vk}, such that

$$\mathsf{A}_{\mathsf{vk}}(pp, vk, \Delta sk) = vk \cdot g^{\Delta sk} = \mathsf{KG}(pp, sk + \Delta sk).$$

3. Given a message M and a fixed random number $r \leftarrow_\$ \mathbb{Z}_p$, the signature σ of M under signing key sk with random number r is

$$\sigma = (\sigma_1 = h^{sk} \cdot (u' \cdot \prod_{i\in[l]} u_i^{M_i})^r, \sigma_2 \leftarrow g^r).$$

The signature $\widetilde{\sigma}$ of M under signing key $sk + \Delta sk$ with the same random number r is

$$\widetilde{\sigma} = (\widetilde{\sigma_1} = h^{sk+\Delta sk} \cdot (u' \cdot \prod_{i\in[l]} u_i^{M_i})^r, \widetilde{\sigma_2} \leftarrow g^r).$$

Note that

$$\widetilde{\sigma_1} = h^{sk+\Delta sk} \cdot (u' \cdot \prod_{i\in[l]} u_i^{M_i})^r = h^{sk} \cdot h^{\Delta sk} \cdot (u' \cdot \prod_{i\in[l]} u_i^{M_i})^r = h^{\Delta sk} \cdot \sigma_1,$$

and

$$\sigma_2 = \widetilde{\sigma_2}.$$

So there exsits a deterministic and efficient algorithm $\mathsf{A}_{\mathsf{Sign}}$, such that

$$\mathsf{A}_{\mathsf{Sign}}(pp, m, (\sigma_1, \sigma_2), \Delta sk) = (h^{\Delta sk} \cdot \sigma_1, \sigma_2) = \mathsf{Sign}(pp, sk + \Delta sk, m, r).$$

The lemma follows. □

It follows from Theorem 3 that $\mathsf{SIG}_{\mathsf{Wat}}$ achieves EUF-CMA security under the CDH assumption. Moreover, Lemma 6 demonstrates that $\mathsf{SIG}_{\mathsf{Wat}}$ possesses the homomorphic property. In addition, Lemma 2 states that any signature scheme that is both homomorphic and EUF-CMA secure also satisfies key-shift* security. Therefore, by combining Theorem 3, Lemma 6, and Lemma 2, we derive the following corollary.

Corollary 1. $\mathsf{SIG}_{\mathsf{Wat}}$ *is key-shift* secure under the computational Diffie–Hellman (CDH) assumption in* $\mathbb{G}$.

We use the homomorphic average-case strong extractor Ext in Equation (1), the syndrome-based secure sketch introduced in Subsect. 2.4, and the Waters signature scheme $\mathsf{SIG}_{\mathsf{Wat}}$ as fundamental components to instantiate the generic construction presented in Figure 2. The detailed construction is illustrated in Figure 5. This results in a fuzzy signature scheme $\mathsf{SIG}_{\mathsf{F}}$ that is secure under the CDH assumption over a bilinear group in the standard model. Moreover, the signing algorithm of this scheme does not require the verification key as extra input.

$\underline{\mathsf{Setup}_{\mathsf{F}}(1^\lambda)}$:	$\underline{\mathsf{KeyGen}_{\mathsf{F}}(pp_{\mathsf{F}}, \mathsf{w})}$:
$i \leftarrow_{\$} \mathcal{I}$.	$(i, pp) \leftarrow pp_{\mathsf{F}}$.
$\mathcal{BG} := (p, \mathbb{G}, \mathbb{G}_T, g, e) \leftarrow \mathsf{BGGen}(1^\lambda)$.	$\mathsf{s} \leftarrow \mathsf{syn}(\mathsf{w}), sk \leftarrow \mathsf{Ext}(\mathsf{w}, i)$.
$h, u', u_1, \ldots, u_l \leftarrow_{\$} \mathbb{G}$.	$vk \leftarrow g^{sk}, vk_{\mathsf{F}} \leftarrow (\mathsf{s}, vk)$.
$pp \leftarrow (\mathcal{BG}, h, u', (u_i)_{i\in[l]})$.	Return vk_{F}.
Return $pp_{\mathsf{F}} \leftarrow (i, pp)$.	
$\underline{\mathsf{Sign}_{\mathsf{F}}(pp_{\mathsf{F}}, \mathsf{w}', m)}$:	$\underline{\mathsf{Verify}_{\mathsf{F}}(pp_{\mathsf{F}}, vk_{\mathsf{F}}, m, \sigma_{\mathsf{F}})}$:
$(i, pp) \leftarrow pp_{\mathsf{F}}$	$(i, pp) \leftarrow pp_{\mathsf{F}}, (\mathsf{s}, vk) \leftarrow vk_{\mathsf{F}}, (\mathsf{s}', \sigma) \leftarrow \sigma_{\mathsf{F}}$.
$\mathsf{s}' \leftarrow \mathsf{syn}(\mathsf{w}'), sk' \leftarrow \mathsf{Ext}(\mathsf{w}', i)$.	$(\sigma_1, \sigma_2) \leftarrow \sigma, M \leftarrow (m, \mathsf{s}')$.
$M \leftarrow (m, \mathsf{s}')$.	Parse M as $(M_1 \parallel \cdots \parallel M_l) \in \{0,1\}^l$.
Parse M as $(M_1 \parallel \cdots \parallel M_l) \in \{0,1\}^l$.	$\Delta\mathsf{w} \leftarrow f_{rec}(\mathsf{s}', \mathsf{s})$.
$r \leftarrow_{\$} \mathbb{Z}_p$.	$\Delta sk \leftarrow \mathsf{Ext}(\Delta\mathsf{w}, i)$.
$\sigma_1 \leftarrow h^{sk'} \cdot (u' \cdot \prod_{i\in[l]} u_i^{M_i})^r$.	$vk' \leftarrow vk \cdot g^{\Delta sk}$.
$\sigma_2 \leftarrow g^r$.	If $e(\sigma_2, u' \cdot \prod_{i\in[l]} u_i^{M_i}) \cdot e(vk', h) = e(\sigma_1, g)$,
$\sigma \leftarrow (\sigma_1, \sigma_2), \sigma_{\mathsf{F}} \leftarrow (\mathsf{s}', \sigma)$.	$b = 1$, else $b = 0$.
Return σ_{F}.	Return b.

Figure 5. Instantiation of fuzzy signature scheme $\mathsf{SIG}_{\mathsf{F}}$ from secure sketch.

Corollary 2. *The instantiation illustrated in Figure 5 constitutes a* $(\mathcal{W}, \mathcal{M}, t)$-$\mathsf{SIG}_{\mathsf{F}}$, *for which* $\mathfrak{m}$*-EUF-CMA* security is established in the standard model, assuming the hardness of the CDH problem in bilinear groups.*

6. Future Work

One main limitation of our scheme is its reliance on pairing operations, which may impact efficiency in practice. Constructing a more efficient fuzzy signature scheme—potentially based on alternative assumptions such as lattices or hash-based primitives—is an important direction for future work.

Moreover, while our security model allows the adversary to choose errors adaptively, designing a fuzzy signature scheme in which the errors can be arbitrarily correlated with the biometric data itself remains an open and challenging problem.

Author Contributions: Conceptualization, Y.W.; Methodology, Y.W. and W.L.; Validation, T.J.; Investigation, T.J.; Writing—original draft, Y.W. and T.J.; Writing—review & editing, Y.W. and W.L.; Supervision, Y.W. and W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (Grant No. 62102077) and the Shanghai Natural Science Foundation (Grant No. 24ZR1401300).

Data Availability Statement: Data are contained within the article.

Acknowledgments: We appreciate the editors and reviewers for their time and consideration in evaluating this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Syndrome-Based Secure Sketch

Recall that an efficiently decodable $[n, k, 2t+1]_{\mathcal{F}}$ linear error-correcting code $\mathcal{C}$ can correct up to t errors. This code is a linear subspace of $\mathcal{F}^n$ with dimension k. The parity-check matrix H of $\mathcal{C}$ is an $(n-k) \times n$ matrix whose rows span the orthogonal space $\mathcal{C}^{\perp}$. For any vector $\mathsf{w} \in \mathcal{F}^n$, the syndrome of w is defined as

$$\mathrm{syn}(\mathsf{w}) := H\mathsf{w}.$$

It is easy to verify that

$$\mathsf{w} \in \mathcal{C} \Leftrightarrow \mathrm{syn}(\mathsf{w}) = 0.$$

For any codeword $c \in \mathcal{C}$ and error vector e, we have

$$\mathrm{syn}(c+e) = \mathrm{syn}(c) + \mathrm{syn}(e) = \mathrm{syn}(e).$$

A linear error-correcting code implies a syndrome-based secure sketch [20], which operates as follows:

- **Secure Sketch (SS):** $\mathrm{SS}(\mathsf{w}) := \mathrm{syn}(\mathsf{w}) = s$.
- **Reconstruction (Rec):** $\mathrm{Rec}(\mathsf{w}', s) := \mathsf{w}' - \mathrm{Decode}(\mathrm{syn}(\mathsf{w}') - s)$.

Appendix B. RKA Security

Here, we revisit the definition of RKA security for signature schemes, as proposed by Bellare et al. [31].

Definition A1 (RKA Security)**.** *A signature scheme* $\mathsf{SIG} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *is* Φ*-RKA secure if for all* PPT *adversaries* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\Phi-\mathsf{RKA}}_{\mathsf{SIG},\mathcal{A}}(1^\lambda) := \Pr[\, \mathsf{Expt}^{\Phi-\mathsf{RKA}}_{\mathsf{SIG},\mathcal{A}}(1^\lambda) \Rightarrow 1 \,] \leqslant \mathsf{negl}(\lambda),$$

where $\mathsf{Expt}^{\Phi-\mathsf{RKA}}_{\mathsf{SIG},\mathcal{A}}(1^\lambda)$*, defined as follows, is an experiment played between an adversary* $\mathcal{A}$ *and a challenger* $\mathcal{C}$*.*

$\underline{\mathsf{Expt}^{\Phi-\mathsf{RKA}}_{\mathsf{SIG},\mathcal{A}}(1^\lambda)}$*:*

1. *The challenger* $\mathcal{C}$ *runs* $pp \leftarrow \mathsf{Setup}(1^\lambda)$*, computes* $(vk, sk) \leftarrow \mathsf{KeyGen}(pp)$*, then initializes* $\mathcal{Q} = \emptyset$*, which denotes the set of queries selected by* $\mathcal{A}$*. Afterwards,* $\mathcal{C}$ *returns* pp *and* vk *to* $\mathcal{A}$*.*
2. *Over the course of the experiment,* $\mathcal{A}$ *is allowed to adaptively make signing oracle queries in the following form:*

- $\mathcal{A}$ *sends a message* $m_j \in \mathcal{M}$ *and a funcition* $\phi_j \in \Phi$ *to the challenger* $\mathcal{C}$.
- $\mathcal{C}$ *computes* $\sigma_j \leftarrow \mathsf{Sign}(pp, \phi_j(sk), m_j)$, *adds* m_j *to the query set* $\mathcal{Q}$, *and replies to* $\mathcal{A}$ *with* σ_j.

3. *In the final phase,* $\mathcal{A}$ *outputs a candidate forgery in the form of a message–signature pair* (m^*, σ^*). *The experiment returns 1 if* $m^* \notin \mathcal{Q} \wedge \mathsf{Verify}(pp, vk, m^*, \sigma^*) = 1$; *otherwise, it returns 0.*

Appendix C. Bilinear Groups

We define a tuple $\mathcal{BG} = (p, \mathbb{G}, \mathbb{G}_T, g, e)$ to be a symmetric bilinear group if the following conditions are satisfied:

- p is a prime number;
- $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order p;
- g is a generator of $\mathbb{G}$;
- $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map computable in time polynomial in $|p|$, satisfying the following:
  - **Bilinearity:** For all $g \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have
  $$e(g^a, g^b) = e(g, g)^{ab}.$$
  - **Non-degeneracy:** For all generators g of $\mathbb{G}$, $e(g, g)$ is not the identity in $\mathbb{G}_T$.

For simplicity, we let BGGen denote an algorithm—known as a "bilinear group generator"—which, on input 1^λ, returns a bilinear group description $\mathcal{BG} = (p, \mathbb{G}, \mathbb{G}_T, g, e)$ such that $|p| = \Theta(\lambda)$.

References

1. Kaur, R.; Kaur, A. Digital signature. In Proceedings of the 2012 International Conference on Computing Sciences, Phagwara, India, 14–15 September 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 295–301. [CrossRef]
2. Yin, H.L.; Fu, Y.; Li, C.L.; Weng, C.X.; Li, B.H.; Gu, J.; Lu, Y.S.; Huang, S.; Chen, Z.B. Experimental quantum secure network with digital signatures and encryption. *Natl. Sci. Rev.* **2023**, *10*, nwac228. [CrossRef] [PubMed]
3. Alzubi, J.A. Blockchain-based Lamport Merkle digital signature: Authentication tool in IoT healthcare. *Comput. Commun.* **2021**, *170*, 200–208. [CrossRef]
4. Jiang, Q.; Ma, J.; Li, G.; Li, X. Improvement of robust smart-card-based password authentication scheme. *Int. J. Commun. Syst.* **2015**, *28*, 383–393. [CrossRef]
5. Lu, H.K. Network smart card review and analysis. *Comput. Netw.* **2007**, *51*, 2234–2248. [CrossRef]
6. Wala'a, M.A.; Abusaimeh, H. Modified USB Security Token for User Authentication. *Comput. Inf. Sci.* **2015**, *8*, 51–63. [CrossRef]
7. Jiang, L.; Li, X.; Cheng, L.; Guo, D. Identity authentication scheme of cloud storage for user anonymity via USB token. In Proceedings of the 2013 International Conference on Anti-Counterfeiting, Security and Identification (ASID), Shanghai, China, 25–27 October 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1–6. [CrossRef]
8. Takahashi, K.; Matsuda, T.; Murakami, T.; Hanaoka, G.; Nishigaki, M. A signature scheme with a fuzzy private key. In *Applied Cryptography and Network Security, Proceedings of the International Conference on Applied Cryptography and Network Security, New York, NY, USA, 2–5 June 2015*; Malkin, T., Kolesnikov, V., Lewko, A., Polychronakis, M., Eds.; LNCS; Springer: Cham, Switzerland, 2015; Volume 9092, pp. 105–126. [CrossRef]
9. Tan, J.; Bauer, L.; Bonneau, J.; Cranor, L.F.; Thomas, J.; Ur, B. Can Unicorns Help Users Compare Crypto Key Fingerprints? In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI'17, Denver, CO, USA, 6–11 May 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 3787–3798. [CrossRef]
10. Yoon, S.; Jain, A.K.. Longitudinal study of fingerprint recognition. *Proc. Natl. Acad. Sci. USA* **2015**, *112*, 8555–8560. [CrossRef]
11. Omolara, A.E.; Jantan, A.; Abiodun, O.I.; Arshad, H.; Mohamed, N.A. Fingereye: Improvising security and optimizing ATM transaction time based on iris-scan authentication. *Int. J. Electr. Comput. Eng.* **2019**, *9*, 1879–1886. [CrossRef]
12. Matsuda, T.; Takahashi, K.; Murakami, T.; Hanaoka, G. Fuzzy signatures: Relaxing requirements and a new construction. In *Applied Cryptography and Network Security, Proceedings of the International Conference on Applied Cryptography and Network Security, Guildford, UK, 19–22 June 2016*; Manulis, M., Sadeghi, A.R., Schneider, S., Eds.; LNCS; Springer: Cham, Switzerland, 2016; Volume 9696, pp. 97–116. [CrossRef]

13. Schnorr, C.P. Efficient identification and signatures for smart cards. In *Advances in Cryptology—CRYPTO' 89 Proceedings, Proceedings of the 9th Annual International Cryptology Conference, Santa Barbara, CA, USA, 20–24 August 1989*; Brassard, G., Ed.; LNCS; Springer: Cham, Switzerland, 1990; Volume 435, pp. 239–252. [CrossRef]
14. Katsumata, S.; Matsuda, T.; Nakamura, W.; Ohara, K.; Takahashi, K. Revisiting fuzzy signatures: Towards a more risk-free cryptographic authentication system based on biometrics. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, CCS '21, Virtual, 15–19 November 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 2046–2065. [CrossRef]
15. Peikert, C. A decade of lattice cryptography. *Found. Trends® Theor. Comput. Sci.* **2016**, *10*, 283–424. [CrossRef]
16. Odlyzko, A. Discrete logarithms in finite fields and their cryptographic significance. In *Proceedings of the Advances in Cryptology*; Beth, T., Cot, N., Ingemarsson, I., Eds.; Springer: Cham, Switzerland, 1984; Volume 209, *LNCS*, pp. 224–314. [CrossRef]
17. Koblitz, N.; Menezes, A.J. The random oracle model: A twenty-year retrospective. *Des. Codes Cryptogr.* **2015**, *77*, 587–610. .. [CrossRef]
18. Song, J.; Wen, Y. A generic construction of fuzzy signature. In *Information Security and Cryptology, Proceedings of the International Conference on Information Security and Cryptology, Virtual,12–14 August 2021*; Yu, Y., Yung, M., Eds.; LNCS; Springer: Cham, Switzerland, 2021; Volume 13007, pp. 23–41. [CrossRef]
19. Bein, B. Entropy. *Best Pract. Res. Clin. Anaesthesiol.* **2006**, *20*, 101–109. [CrossRef] [PubMed]
20. Dodis, Y.; Reyzin, L.; Smith, A.D. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *EUROCRYPT 2004, Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004*; Cachin, C., Camenisch, J., Eds.; LNCS; Springer: Cham, Switzerland, 2004; Volume 3027, pp. 523–540. [CrossRef]
21. Joux, A.; Nguyen, K. Separating decision Diffie–Hellman from computational Diffie–Hellman in cryptographic groups. *J. Cryptol.* **2003**, *16*, 239–247. [CrossRef]
22. Tian, Y.; Li, Y.; Deng, R.; Sengupta, B.; Yang, G. Lattice-based remote user authentication from reusable fuzzy signature. *J. Comput. Secur.* **2021**, *29*, 273–298. [CrossRef]
23. Apon, D.; Cho, C.; Eldefrawy, K.; Katz, J. Efficient, reusable fuzzy extractors from LWE. In *Cyber Security Cryptography and Machine Learning, Proceedings of the first International Conference, CSCML 2017, Beer-Sheva, Israel, 29–30 June 2017*; Dolev, S., Lodha, S., Eds.; LNCS; Springer: Cham, Switzerland, 2017; Volume 10332, pp. 1–18. [CrossRef]
24. Wen, Y.; Liu, S. Reusable fuzzy extractor from LWE. In *Information Security and Privacy, Proceedings of the ACISP 2018, Wollongong, NSW, Australia, 11–13 July 2018*; Susilo, W., Yang, G., Eds.; LNCS; Springer: Cham, Switzerland, 2018; Volume 10946, pp. 13–27. [CrossRef]
25. Zhou, Y.; Liu, S.; Han, S. Robustly Reusable Fuzzy Extractor from Isogeny. *Theor. Comput. Sci.* **2024**, *1008*, 114677. [CrossRef]
26. Lyubashevsky, V. Lattice signatures without trapdoors. In *Advances in Cryptology—EUROCRYPT 2012, Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, 15–19 April 2012*; Springer: Cham, Switzerland, 2012; Volume 7237, pp. 738–755. [CrossRef]
27. Zheng, M.; Liu, Z.; Mambo, M. A Provably Secure Lattice-Based Fuzzy Signature Scheme Using Linear Sketch. *IEEE Access* **2023**, *11*, 62510–62521. [CrossRef]
28. Wang, S.; Zhu, Y.; Ma, D.; Feng, R. Lattice-based key exchange on small integer solution problem. *Sci. China Inf. Sci.* **2014**, *57*, 1–12. [CrossRef]
29. Wen, Y.; Liu, S. Robustly reusable fuzzy extractor from standard assumptions. In *Advances in Cryptology—ASIACRYPT 2018, Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, 2–6 December 2018*; Peyrin, T., Galbraith, S., Eds.; LNCS; Springer: Cham, Switzerland, 2018; Volume 11274, pp. 459–489. [CrossRef]
30. Shoup, V. *A Computational Introduction to Number Theory and Algebra*; Cambridge University Press: Cambridge, UK, 2006.
31. Bellare, M.; Cash, D.; Miller, R. Cryptography Secure against Related-Key Attacks and Tampering. In *Advances in Cryptology—ASIACRYPT 2011, Proceedings of the 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, Republic of Korea, 4–8 December 2011*; Lee, D.H., Wang, X., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2011; Volume 7073, pp. 486–503. [CrossRef]

32. Waters, B. Efficient identity-based encryption without random oracles. In *Advances in Cryptology—EUROCRYPT 2005, Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005*; Springer: Cham, Switzerland, 2005; Volume 3494, pp. 114–127. [CrossRef]
33. Hofheinz, D.; Jager, T.; Knapp, E. Waters Signatures with Optimal Security Reduction. In Public Key Cryptography—PKC 2012, Proceedings of the 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, 21–23 May 2012; Fischlin, M., Buchmann, J., Manulis, M., Eds.; Springer: Cham, Switzerland, 2012; Volume 7293, pp. 66–83. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.