applied sciences

MDPI

Article

An Improved Fuzzy Vector Signature with Reusability

Ilhwan Lim [1], Minhye Seo [2], Dong Hoon Lee [1] and Jong Hwan Park [3,*]

1 Graduate School of Information Security, Korea University, Seoul 02841, Korea; money683@korea.ac.kr (I.L.); donghlee@korea.ac.kr (D.H.L.)
2 Department of Cyber Security, Duksung Women's University, Seoul 01369, Korea; mhseo@duksung.ac.kr
3 Department of Computer Science, Sangmyung University, Seoul 03016, Korea
* Correspondence: jhpark@smu.ac.kr

Received: 24 July 2020; Accepted: 1 October 2020; Published: 14 October 2020

Abstract: Fuzzy vector signature (FVS) is a new primitive where a fuzzy (biometric) data w is used to generate a verification key (VK_w), and, later, a distinct fuzzy (biometric) data w' (as well as a message) is used to generate a signature $(\sigma_{w'})$. The primary feature of FVS is that the signature $(\sigma_{w'})$ can be verified under the verification key (VK_w) only if w is close to w' in a certain predefined distance. Recently, Seo et al. proposed an FVS scheme that was constructed (loosely) using a subset-based sampling method to reduce the size of helper data. However, their construction fails to provide the reusability property that requires that no adversary gains the information on fuzzy (biometric) data even if multiple verification keys and relevant signatures of a single user, which are all generated with correlated fuzzy (biometric) data, are exposed to the adversary. In this paper, we propose an improved FVS scheme which is proven to be reusable with respect to arbitrary correlated fuzzy (biometric) inputs. Our efficiency improvement is achieved by strictly applying the subset-based sampling method used before to build a fuzzy extractor by Canetti et al. and by slightly modifying the structure of the verification key. Our FVS scheme can still tolerate sub-linear error rates of input sources and also reduce the signing cost of a user by about half of the original FVS scheme. Finally, we present authentication protocols based on fuzzy extractor and FVS scheme and give performance comparison between them in terms of computation and transmission costs.

Keywords: fuzzy vector signature; reusability; biometric authentication

1. Introduction

Biometric information (e.g., fingerprint, iris, face, vein) has been used for user authentication [1–5] because of its uniqueness and immutability. Due to these properties, such biometric information can be used in place of a user secret key in an authentication system. When using biometric information as a security key, the user is not required to memorize or securely store anything to authenticate, which makes the process much more convenient and user-friendly. However, since biometric information is *noisy* and *non-uniformly* distributed, it differs greatly from what is known about cryptographic secret keys. In general, a secret key of an authentication system is largely set to be a uniformly random string of fixed length. Until now, a large body of research has been conducted to bridge this gap and enable biometric information to be used as a secret key in a cryptographic way.

To overcome the problem of noisy secret keys, researchers proposed the fuzzy extractor and fuzzy signature as two types of biometric cryptosystems. Comprised of two algorithms, **Gen** and **Rep**, a fuzzy extractor [6] is able to generate a uniformly random string of fixed length (i.e., a secret key) from fuzzy (biometric) data. The generation algorithm **Gen** takes as input sample f fuzzy (biometric) data w to generate a secret key r together with helper data p. The reproduction algorithm **Rep** takes as input

another sample of fuzzy (biometric) data w' close to w and p to reproduce r. If the difference between w and w' is less than a minimal threshold value, **Rep** can generate the same secret key r. On the other hand, consisting of three algorithms, **KG**, **Sign**, and **Vrfy**, the fuzzy signature [7] generates a signature by using fuzzy (biometric) data itself as a signing key. The key generation algorithm **KG** takes as input sample of fuzzy (biometric) data w to generate a verification key vk. The signing algorithm **Sign** takes as input another sample of biometric information w' to generate a signature σ. The verification algorithm **Vrfy** takes as input vk and σ and succeeds in verification if w and w' are close to within a fixed threshold.

Biometric cryptosystems are generally considered secure when each sample of fuzzy (biometric) data is used only once. However, in reality, a user may use the same biometric source (e.g., right index fingerprint) to authenticate their accounts for several applications as a matter of expediency. Since similar biometric information is used multiple times, a new security notion is required to guarantee both the privacy of the fuzzy (biometric) data at hand and the reusable security of biometric cryptosystems in this situation. In 2004, Boyen [8] introduced the *reusability* of a fuzzy extractor, which ensures no entropy loss to the secret key or biometric source even when relevant pairs of helper data or keys from similar forms of biometric information are revealed. Since then, many researchers have focused on studying this fuzzy extractor with reusability (a.k.a. reusable fuzzy extractor).

In order for a reusable fuzzy extractor to be widely used in practice, it should be able to tolerate more than a certain level of errors inherent in fuzzy (biometric) data. For example, iris readings have the average error rate of 20–30% [9–11]. A number of studies have proposed constructions that tolerate linear fraction of errors [8,12–15], but these schemes impose a strong requirement on the distribution of fuzzy (biometric) data, namely: (1) the distribution must have sufficiently high min-entropy ("Y" in High min-entropy in Table 1), or (2) any difference between two distinct biometric readings cannot significantly decrease the min-entropy of the fuzzy (biometric) data ("$\mathbf{H}_\infty[w_i|w_i - w_j] > m$" in Source Distribution in Table 1). Unfortunately, both of these expectations are somewhat unrealistic.

Table 1. Comparison with reusable biometric cryptosystems.

	Schemes	Secure Sketch	High Min-Entropy	Security Assumption	Error Tolerance Rate	Reusability	Source Distribution
FE	[8]	O	Y	+	linear	weak	$w_i = w + \delta_i$
	[16]	X	N	LWE	log	Strong	$w_i = w + \delta_i$
	[13]	O	Y	LWE	linear	Strong	$w_i = w + \delta_i$
	[15]	O	Y	DDH	linear	Strong	$w_i = w + \delta_i$
	[14]	O	Y	DDH	linear	Strong	$\mathbf{H}_\infty[w_i\|w_i - w_j] > m$
	[17]	X	N	X	sub-lin	Strong	(w, w_i)
	[18]	X	N	X	sub-lin	Strong	(w, w_i)
	[12]	X	Y	X	linear	Strong	(w, w_i)
FS	[19]	X	N	LWE	log	Strong	$w_i = w + \delta_i$
FVS	[20]	X	N	XDH	sub-lin	△	(T, k)
	Ours	X	N	XDH	sub-lin	Strong	(w, w_i)

• In Secure Sketch, If the scheme used the secure sketch, "O". Otherwise, "X". • In Reusability, "weak" means that the scheme is proven in the weak reusability model; • "△" means that the scheme does not provide a formal proof; • In Source Distribution, "$w_i = w + \delta_i$" means that, for a fuzzy (biometric) source w, the error δ_i is controlled by an adversary; • "(w, w_i)" means that the biomteric readings w and w_i are arbitrary correlated; • "(T, k)" means the (T, k)-block source in Reference [21]; • High Min-entropy means that min-entropy is higher than some value secure against brute-force attack. In High Min-entropy, "Y" is indicated if the scheme requires sufficiently high min-entropy of the input (biometric) data, and "N" otherwise; • In Security Assumption, "+" means that the scheme is information-theoretically secure.

Canetti et al. [17] relaxed these conditions, only requiring a subset of samples to have sufficient average min-entropy for given subset indices, and proposed a reusable fuzzy extractor that would tolerate a sub-linear fraction of errors. The construction is contingent on the existence of a powerful tool called *digital locker*, which is a non-standard assumption. Other reusable biometric cryptosystems [16,19] did not require either unrealistic requirements on the biometric distribution or contain non-standard assumptions, but they only tolerated a logarithmic fraction of errors.

Recently, a new primitive called fuzzy vector signature (FVS) [20] was proposed based on bilinear maps (i.e., pairings), which improved the error tolerance rate without any additional requirements on the distribution of biometric information. This scheme tolerates a sub-linear fraction of errors and also is based on standard assumptions, like the external Diffie-Hellman (XDH). It is also claimed to be reusable, but no formal proof for reusability was provided in Reference [20]. In this paper, we introduce the formal security model for reusability of fuzzy vector signature (FVS) and prove that our proposed scheme is reusable in the reusability model. By more strictly applying the subset-based sampling method [17], our scheme is also more efficient than Reference [20] from the perspective of the user and the authentication server. Specifically, it reduces not only the size of the signature and verification key but also the number of pairing operations required for verification. Section 5 outlines a detailed performance comparison.

1.1. Related Work

Reusable Fuzzy Extractor. The concept of a fuzzy extractor and secure sketch was first proposed by Dodis et al. [6]. Following this, Boyen et al. [8] introduced the notion of reusability, meaning that additional keys could be securely generated for any helper data, which is required to regenerate the key, even if the helper data or key pairs were exposed. Wen et al. [14] subsequently proposed a reusable fuzzy extractor based on the Decisional Diffie-Hellman (DDH) problem, which can tolerate a linear size of errors. However, their scheme requires that, for any two distinct readings of the same source, the differences between them should not leak significant information about the source—a requirement that is too strict, as each component of fuzzy (biometric) data is non-uniformly distributed. In response to this, Wen et al. [15] proposed a DDH-based reusable fuzzy extractor to remove the requirement [14] by changing the source distribution. Apart from these studies, Wen et al. [13] also proposed a reusable fuzzy extractor that allowed a linear fraction of errors based on the Learning with Errors (LWE) assumption [22]. However, their schemes [13–15] used a secure sketch as a method for controlling noise of data. A secure sketch is used for recovering w from w' if w and w' are close to within a fixed distance, but it nevertheless causes a small leakage of biometric information. Therefore, when considering reusability scenarios, it is noted that these schemes require an input source to have sufficiently high min-entropy to avoid brute-force attack even after security loss.

On the other side of the spectrum are reusable fuzzy extractors that did not suggest using secure sketches. Canetti et al. [17] proposed a reusable fuzzy extractor that is constructed using the subset-based sampling technique instead of a secure sketch to control a noisy data source. This scheme relied on the use of a digital locker to generate helper data. However, even if a digital locker was simply instantiated with a hash function, the size of helper data would increase significantly (e.g., about 0.8 GB to achieve an error tolerance rate of 20%). Cheon et al. [18] modified this scheme [17] to reduce the size of the helper data, but in turn, the computational cost for **Rep** increased significantly. Alamélou et al. [12] also improved [17] and suggested a fuzzy extractor that achieved a linear fraction of errors. However, they also used a digital locker and this scheme had an unrealistic requirement that each component of a fuzzy (biometric) source had to have significant min-entropy. Apon et al. [16] modified a non-reusable fuzzy extractor [23] based on the LWE assumption into a reusable one. However, their scheme can tolerate only a logarithmic fraction of errors due to the time-consuming process of reproducing a key even with a small number of errors.

Fuzzy Signature. The concept of a fuzzy signature was proposed by Takahashi et al. [7]. The fuzzy signature does not need helper data, unlike the fuzzy extractor, because it only requires a valid correlation between two input fuzzy (biometric) data relevant to a verification key and a signature, which can be achieved using a linear sketch. However, in Reference [7], a strong assumption was required that input fuzzy (biometric) data should be uniformly distributed, which is then relaxed in Reference [24]. Afterwards, Yasuda et al. announced that a linear sketch of fuzzy extractors [7,24] is vulnerable to recovery attacks [25]. In Reference [26], the merge of Reference [7,24], Takahashi et al. proposed two instantiations of a fuzzy signature secure against recovery attacks. In 2019, Tian et al.

first introduced the notion of reusability in a fuzzy signature [19]. To construct a reusable fuzzy signature, they adopted a reusable fuzzy extractor in Reference [16]. Consequently, the reusability is limited only to the generation of verification keys, ignoring the privacy of signatures.

Fuzzy Vector Signature. Seo et al. first proposed a fuzzy vector signature [20] following the subset-based sampling method of Reference [17]. The fuzzy vector signature requires the signing parameter like helper data in fuzzy extractor, to be reusable, but the size of the signing parameter is much smaller than that of helper data in Reference [17]. In addition, a fuzzy vector signature can tolerate sub-linear errors, while a reusable fuzzy signature [19] can only tolerate logarithmic errors. However, the size of verification key in Reference [20] is still huge, which results in high computational costs for verification. In addition, security models for reusability are incomplete as in Reference [19], and, as a result, no formal proof of reusability is provided.

1.2. Source Distributions

A source distribution of reusable biometric cryptosystems can be categorized into four types, according to correlation between two repeating readings, as in Table 1.

"$w_i = w + \delta_i$" implies that the hamming distance between $w \in W$ and $w + \delta_i$ is less than a threshold value t, where W is an input source. Since δ_i is randomly chosen by an adversary, $w + \delta_i$ may not be included in W, which is a little far from reality. Especially in Reference [16,19], additional assumptions were made that W should be a small error distribution χ of Learning with Errors (LWE) problem and both w and $w + \delta_i$ should be in W to tolerate logarithmic errors.

"$\mathbf{H}_\infty[w_i|w_i - w_j] > m$" implies that, for any two distinct readings w_i and w_j in W, $\mathbf{H}_\infty[w_i|w_i - w_j] > m$ holds where m is a minimum level of security. In other words, the difference between w_i and w_j (i.e., $w_i - w_j$) should not leak too much information of w_i even if $w_i - w_j$ is correlated to w_i, which is a strong requirement for the input source.

"(w, w_i)" implies that any two distinct readings are arbitrarily correlated, which would be the most realistic assumption. However, as a trade-off, schemes based on this distribution require additional assumptions on the input source. For example, in Reference [17,18], any subset of $W = (W[1], \ldots, W[n])$ should have high min-entropy even if indices are exposed, and, in Reference [12], each component $W[j]$ should have a high min-entropy even if other components are exposed.

Unlike the above three types, a previous fuzzy vector signature [20] used a (T, k)-block source in Reference [21], although following the subset concept in Reference [17] for construction. (T, k)-block source means that for input sources $W_1, \ldots, W_T$, i-th source has a high min-entropy even though $i - 1$ readings are set to $w_1, \ldots, w_{i-1}$, i.e., $\mathbf{H}_\infty[W_i \mid W_1 = w_1, \ldots, W_{i-1} = w_{i-1}] > k$ for all $i \in [1, T]$ where k is a minimum level of security. Namely, $W_1, \ldots, W_T$ are not correlated. However, since fuzzy (biometric) data from the same source must be correlated, it is inappropriate to use the (T, k)-block source when considering reusability.

1.3. Contribution

In this paper, we propose a new fuzzy vector signature (FVS) scheme based on a subset concept in Reference [17] to deal with noise in fuzzy (biometric) data. In consequence, our scheme is reusable and can tolerate sub-linear errors without any additional requirements, such as sufficiently high min-entropy of the input source. Compared to the previous fuzzy vector signature [20], we eliminated redundant parts in both verification key and signature by trying a new approach to security proofs, which in turn improved the efficiency of the scheme. For instance, for 80-bit security with 20% error tolerance rates, we reduce the size of the signing parameter by 33%, from 48 KB to 32 KB, the signature by 50%, from 32 KB to 16 KB, and a verification key by 22%, from 1.61 GB to 1.25 GB, where the length of the fuzzy (biometric) data is $n = 512$ bits. Additionally, we reduce the number of pairing operations in verification by up to 33% from 18.5×10^6 to 12.3×10^6.

We also provide the formal security models to reflect reusability of the privacy of the verifier and signer (i.e., VK-privacy and SIG-privacy, respectively) and the unforgeability of the signature

(i.e., Reusability). And we prove these properties on the assumption that the repeating readings of fuzzy (biometric) data are arbitrarily correlated, which is more realistic than the (T, k)-block source used in Reference [20]. In addition, we analyze the performance of our FVS scheme in terms of transmission and computational costs by comparing it with previous reusable biometric cryptosystems [17,18,20]. In particular, a signature can be generated in 155 ms on the signer's side, which is almost twice as fast as in Reference [20] (for more details, see Section 5).

2. Preliminaries

2.1. Notation

Let λ be the security parameter and $\text{poly}(\lambda)$ denote a polynomial in variable λ. We use the acronym "PPT" to indicate probabilistic polynomial-time. Let $\mathcal{W}$ be a fuzzy (biometric) space with metric space $\mathcal{M}$ and $\mathcal{Z}$ be a set of arbitrary alphabet. We denote $w \leftarrow W$ as the process of sampling w according to a random variable W such that $W \in \mathcal{W}$. Let $w = \{w[1], \ldots, w[n]\}$ be a fuzzy vector of length n. We denote string concatenation with the symbol "||," and U represents an arbitrary random distribution.

2.2. Hamming Distance Metric

A metric space is a set $\mathcal{M}$ with a non-negative integer distance function **dis**: $\mathcal{M}^2 \rightarrow \mathbb{Z} \cup \{0\}$. The elements of $\mathcal{M}$ are vectors in $\mathcal{Z}^n$ for some alphabet sets $\mathcal{Z}$. For any $w, w' \in \mathcal{M}$, the hamming distance $\mathbf{dis}(w, w') = |\{i | w_i \neq w'_i\}|$ is defined as the number of components in which w and w' differ.

2.3. Min-Entropy

Let X and Y be random variables. The min-entropy of X is defined as $\mathbf{H}_\infty[X] = -\log_2\left(\max_{x \in X} \mathbf{P}[X = x]\right)$. For conditional distributions, the average min-entropy of X given Y is defined by $\tilde{\mathbf{H}}_\infty[X|Y] = -\log_2\left(\mathbf{E}_{y \in Y}\left[\max_{x \in X} \mathbf{P}[X = x|Y = y]\right]\right)$. For a given source $X = (X[1], \ldots, X[n])$, we say that X is a source with α-entropy ℓ-samples if

$$\tilde{\mathbf{H}}_\infty[X[j_1], \ldots, X[j_\ell]|j_1, \ldots, j_\ell] \geq \alpha,$$

where α and ℓ are determined by the security parameter λ.

2.4. Statistical Distance

The statistical distance between random variables X and Y with the same domain is defined by

$$\Delta(X, Y) = \frac{1}{2} \sum_x |\Pr[X = x] - \Pr[Y = x]|.$$

2.5. Universal Hash Function

A collection $\mathcal{H}$ of hash functions $H : U \rightarrow V$ is $1/|V|$-universal if for any $x_1, x_2 \in U$ such that $x_1 \neq x_2$, it holds that $\Pr_{H \leftarrow \mathcal{H}}[H(x_1) = H(x_2)] = \frac{1}{|V|}$.

Lemma 1. (Reference [6]) *Let A, B, C be random variables. Then,*

(a) For any $\xi > 0$, the conditional entropy $\mathbf{H}_\infty[A|B = b]$ is at least $\tilde{\mathbf{H}}_\infty[A|B] - \log(1/\xi)$ with a probability of at least $1 - \xi$ over the choice of b.

(b) If B has at most 2^λ possible values, then $\tilde{\mathbf{H}}_\infty[A|(B, C)] \geq \tilde{\mathbf{H}}_\infty[(A, B)|C] - \lambda \geq \tilde{\mathbf{H}}_\infty[A|C] - \lambda$. In particular, $\tilde{\mathbf{H}}_\infty[A|B] \geq \mathbf{H}_\infty[(A, B)] - \lambda \geq \mathbf{H}_\infty[A] - \lambda$.

Lemma 2. (Reference [27]) *Let (X, Z) be any two jointly distributed random variables and Z has at most 2^v possible values. Then, for any $\epsilon > 0$ it holds that $\Pr[\mathbf{H}_\infty[X|Z = z] \geq \mathbf{H}_\infty[X] - v - \log(1/\epsilon)] \geq 1 - \epsilon$.*

Lemma 3. *Let $\mathcal{H}$ be a family of universal hash functions $H : U \to V$, and let $(X_1, \ldots, X_q)$ be a joint distribution such that $\mathbf{H}_\infty[X_i] \geq \alpha$ for $i = 1, ..., q$ and $\alpha \geq q \cdot \log|V| + 3\log(1/\epsilon) + \Theta(1)$. Then, the distribution $(H_1, H_1(X_1), \ldots, H_q, H_1(X_q))$, where $(H_1, ..., H_q) \leftarrow \mathcal{H}^q$, is $2\epsilon q$-close to the uniform distribution over $(\mathcal{H} \times V)^q$.*

Proof. Proof of Lemma 3 is in Appendix A. □

2.6. Discrete Logarithm Assumption

Let $\mathbb{G}$ be a group of prime order q, and let g be a generator of $\mathbb{G}$. For any PPT algorithm $\mathcal{A}$, we define the advantage of $\mathcal{A}$, denoted by $\mathsf{Adv}^{\mathsf{DL}}_{\mathcal{A}}(\lambda)$, in solving the discrete logarithm (DL) problem as follows:

$$\mathsf{Adv}^{\mathsf{DL}}_{\mathcal{A}} = \Pr\left[a \leftarrow \mathcal{A}(\mathbb{G}, q, g, g^a); a \leftarrow_R \mathbb{Z}_q\right]$$

We say that the DL assumption holds in $\mathbb{G}$if, for all PPT algorithms $\mathcal{A}$ and any security parameter λ, $\mathsf{Adv}^{\mathsf{DL}}_{\mathcal{A}}(\lambda) < \epsilon(\lambda)$ for some negligible function ϵ.

2.7. Bilinear Maps

Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be groups of some prime order q and a bilinear map (or pairing) $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ over $(\mathbb{G}_1, \mathbb{G}_2)$ be admissible if it satisfies the following properties:

1. Bilinear: The map is bilinear if $e(g^a, h^b) = e(g, h)^{ab}$ for all $g \in \mathbb{G}_1, h \in \mathbb{G}_2$.
2. Non-degenerate: $e(g, h) \neq 1$.
3. Computable: There is an efficient algorithm to compute $e(g, h)$ for all $g \in \mathbb{G}_1, h \in \mathbb{G}_2$.

Our fuzzy vector signature is constructed using a Type-3 pairing where $\mathbb{G}_1 \neq \mathbb{G}_2$ and there is no known efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$.

2.8. External Diffie-Hellman (XDH) Assumption

Let $\mathbb{G}_1, \mathbb{G}_2$ be groups with prime order q and g, h be generators of $\mathbb{G}_1, \mathbb{G}_2$, respectively. The XDH problem in $\mathbb{G}_1$ is defined as follows: given $D = (g, g^a, g^b, h) \in \mathbb{G}_1^3 \times \mathbb{G}_2$ and $T \in \mathbb{G}_1$ with a Type-3 pairing, the goal of an adversary $\mathcal{A}$ is to distinguish whether T is either g^{ab} or random R. For any PPT algorithm $\mathcal{A}$, the advantage in solving the XDH problem in $\mathbb{G}_1$ is defined as

$$\mathsf{Adv}^{\mathsf{XDH}}_{\mathcal{A}}(\lambda) = \left|\Pr\left[\mathcal{A}(D, g^{ab}) \to 1\right] - \Pr\left[\mathcal{A}(D, T) \to 1\right]\right|.$$

We say that an XDH assumption holds in $\mathbb{G}_1$ if, for any PPT algorithm $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{XDH}}_{\mathcal{A}}(\lambda)$ is negligible for λ.

3. Definitions

3.1. Syntax of Fuzzy Vector Signature

Let $\mathcal{W}$ be a fuzzy (biometric) space with the Hamming distance metric $\mathcal{M}$ and w is a sample of random variable $W \in \mathcal{W}$. A fuzzy vector signature (FVS) scheme [20] is defined by three algorithms (**Setup**, **Sign**, **Verify**) as follows:

- **Setup**$(1^\lambda, w, n, d, \ell, t)$: The setup algorithm takes as input the security parameter 1^λ, a sample of fuzzy (biometric) data $w \leftarrow W$, the length n of w, the number d of subsets, the number ℓ of elements included into each subset, and the maximum number t of errors that can be tolerated. It generates a signing parameter SP and verification key VK_w corresponding to w. Here, t/n is said to be the error tolerance rate.
- **Sign**(SP, w', m): The signature generation algorithm takes as input a signing parameter SP, a sample of fuzzy (biometric) data $w' \leftarrow W$ of length n, and a message m. It generates a signature $\sigma_{w'}$ corresponding to w'.

- **Verify**$(\mathsf{VK}_w, \sigma_{w'}, m)$: The verification algorithm takes as input a verification key VK_w, a signature $\sigma_{w'}$, and a message m. If a signature $\sigma_{w'}$ is valid under the condition that $\mathbf{dis}(w, w') \leq t$, it outputs 1; otherwise 0.

Correctness. Let δ be the probability that the **Verify** algorithm outputs 0, and let $\mathbf{dis}(w, w') \leq t$ for two sample of fuzzy (biometric) data $w, w' \in \mathcal{M}$. For the signing parameter SP and the verification key VK_w, if $\sigma_{w'} \leftarrow \mathbf{Sign}(\mathsf{SP}, w', m)$, then $\Pr[\mathbf{Verify}(\mathsf{VK}_w, \sigma_{w'}, m) = 1] \geq 1 - \delta$.

3.2. Security Models

We considered three security notions for FVS security: VK-privacy, SIG-privacy, and reusability.

3.2.1. VK-Privacy

VK-privacy blocks an adversary from obtaining any information about fuzzy input data w from a verification key VK_w. In other words, the adversary cannot distinguish between a real VK_w and a random R. We say that an FVS scheme is VK-private if the advantage that any PPT adversary $\mathcal{A}$ wins against the challenger $\mathcal{C}$ in the following game for any $j = 1, \ldots, q$ is negligible in λ:

- **Setup**: $\mathcal{A}$ selects target correlated random variables $W = (W_1, \ldots, W_q) \in \mathcal{W}^q$ and gives these to $\mathcal{C}$.
- **Generation**: $\mathcal{C}$ samples $(w_1, \ldots, w_q) \in (W_1, \ldots, W_q)$. $\mathcal{C}$ runs **Setup**$(1^\lambda, w_k, n, d, \ell, t)$ for $k = 1, \ldots, q$, and chooses random R. $\mathcal{C}$ chooses one of the modes, real or random experiment. For any $k \in [1, q]$, if the mode is real, $\mathcal{C}$ gives $(\mathsf{VK}_{w_1}, \ldots, \mathsf{VK}_{w_k}, \ldots, \mathsf{VK}_{w_q}, \mathsf{SP}_1, \ldots, \mathsf{SP}_q)$; otherwise $(\mathsf{VK}_{w_1}, \ldots, R, \ldots, \mathsf{VK}_{w_q}, \mathsf{SP}_1, \ldots, \mathsf{SP}_q)$.
- **Distinguishing**: For k, $\mathcal{A}$ outputs a bit $b \in \{0, 1\}$ that, respectively, represents a real or random experiment.

Definition 1 (VK-privacy.). *An FVS scheme Π_f is VK-private if, for any PPT adversary $\mathcal{A}$ against VK-privacy, there exists a negligible function $\nu(\lambda)$ such that* $\mathsf{Adv}^{VK}_{\Pi_f, \mathcal{A}}(\lambda) \triangleq$

$$\Big| \Pr\left[\mathcal{A}(\mathsf{VK}_{w_1}, ..., \mathsf{VK}_{w_k}, ..., \mathsf{VK}_{w_q}, \mathsf{SP}_1, ..., \mathsf{SP}_q) = 1\right] - \Pr\left[\mathcal{A}(\mathsf{VK}_{w_1}, ..., R, ..., \mathsf{VK}_{w_q}, \mathsf{SP}_1, ..., \mathsf{SP}_q) = 1\right] \Big| \leq \nu(\lambda).$$

3.2.2. SIG-Privacy

SIG-privacy means that an adversary cannot ascertain any information on fuzzy input data w' from a signature $\sigma_{w'}$. The adversary cannot distinguish between a valid signature $\sigma_{w'}$ and a signature corresponding to a uniformly random fuzzy data u without a corresponding verification key. We say that an FVS scheme is SIG-private if the advantage that any PPT adversary $\mathcal{A}$ wins against the challenger $\mathcal{C}$ in the following game for $j = 1, \ldots, q$ is negligible in λ:

- **Setup**: $\mathcal{A}$ selects target correlated random variables $W = (W^*, W_1, \ldots, W_q) \in \mathcal{W}$ and gives it to $\mathcal{C}$. $\mathcal{C}$ chooses $(w^*, w_1, \ldots, w_q) \in (W^*, W_1, \ldots, W_q)$. Then, $\mathcal{C}$ runs **Setup**$(1^\lambda, w_j, n, d, \ell, t)$ for $j \in [1, q]$ and **Setup** $(1^\lambda, w^*, n, d, \ell, t)$. $\mathcal{C}$ sends $\{\mathsf{VK}_{w_j}, \mathsf{SP}_j\}_{j=1}^q$ and SP^* to $\mathcal{A}$.
- **Query**: $\mathcal{A}$ issues a random variable W' correlated with $(W^*, W_1, \ldots, W_q)$, a message m_k, and an index $j \in [1, q]$ of the signing parameter SP_j. $\mathcal{C}$ chooses $w' \leftarrow W'$ correlated with $(w^*, w_1, \ldots, w_q)$, runs **Sign** (SP_j, w', m_k), and gives the resulting signature to $\mathcal{A}$.
- **Challenge**: $\mathcal{A}$ issues a message m^*. $\mathcal{C}$ obtains $\sigma^* \leftarrow \mathbf{Sign}(\mathsf{SP}^*, w^*, m^*)$ and selects a bit $b \in \{0, 1\}$ at random. If $b = 0$, $\mathcal{C}$ sends σ^* to $\mathcal{A}$. Otherwise, $\mathcal{C}$ selects a random input u from a uniformly random distribution U, obtains $\sigma^* \leftarrow \mathbf{Sign}(\mathsf{SP}^*, u, m^*)$, and gives σ^* to $\mathcal{A}$.
- **Guess**: $\mathcal{A}$ outputs its guess $b' \in \{0, 1\}$. If $b' = b$, $\mathcal{A}$ wins the game.

Definition 2 (SIG-privacy.). *An FVS scheme Π_f is SIG-private if, for any PPT adversary $\mathcal{A}$ against SIG-privacy, there exists a negligible function $\nu(\lambda)$ such that*

$$\mathsf{Adv}^{SIG}_{\Pi_f,\mathcal{A}}(\lambda) \triangleq \left| \Pr\left[b = b'\right] - \frac{1}{2} \right| \leq \nu(\lambda).$$

3.2.3. Reusability

Reusability means that an adversary cannot generate a valid signature without knowing the target input source of data, even if the adversary is given verification keys and signing parameters correlated with the target input data. In addition, the adversary can get valid signatures that are verified with the target verification key or other (correlated) verification keys via signing oracles. We say that an FVS scheme is reusable if the advantage that any PPT adversary $\mathcal{A}$ wins against the challenger $\mathcal{C}$ in the following game is negligible in λ:

- **Setup**: $\mathcal{A}$ selects correlated random variables $(W_1, \ldots, W_q) \in \mathcal{W}^q$ and gives it to $\mathcal{C}$. $\mathcal{C}$ chooses $(w_1, \ldots, w_q) \in (W_1, \ldots, W_q)$ and runs **Setup**$(1^\lambda, w_j, n, d, \ell, t)$ for $j = 1, \ldots, q$. Then, $\mathcal{C}$ gives $\{\mathsf{VK}_{w_j}, \mathsf{SP}_j\}_{j=1}^q$ to $\mathcal{A}$.
- **Signing query**: $\mathcal{A}$ issues a random variable W' correlated with $(W_1, \ldots, W_q)$, a message m_k, and an index $j \in [1, q]$ of the signing parameter SP_j. $\mathcal{C}$ chooses $w' \leftarrow W'$ correlated with $(w_1, \ldots, w_q)$, and runs **Sign** $\left(\mathsf{SP}_j, w', m_k\right)$. $\mathcal{C}$ sends the resulting signature to $\mathcal{A}$.
- **Output**: $\mathcal{A}$ outputs (m^*, σ^*) such that σ^* was not the output of m^* queried. If **Verify**$\left(\mathsf{VK}_{w_j}, \sigma^*, m^*\right) = 1$ for some $j \in \{1, \ldots, q\}$, $\mathcal{A}$ wins the game.

Definition 3 (Reusability). *An FVS scheme Π_f is reusable in chosen message attacks if, for any PPT adversary $\mathcal{A}$ making at most q_s signature queries, there is a negligible function $\nu(\lambda)$ such that*

$$\mathsf{Adv}^{REU}_{\Pi_f,\mathcal{A}}(\lambda) \triangleq \Pr\left[\mathcal{A} \text{ wins}\right] \leq \nu(\lambda).$$

4. Fuzzy Vector Signature

4.1. Construction

Let $\mathcal{G} = (p, g, h, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ be a bilinear group, where $g \in \mathbb{G}_1$ and $h \in \mathbb{G}_2$ are generators. Let $\mathcal{W}$ be a (biometric) space with the Hamming distance metric, and let $W \in \mathcal{W}$ be a random variable that is the user source. Given W, we consider two fuzzy (biometric) samples such that $w, w' \in W$. In this section, we present our FVS scheme that consists of the following three algorithms: **Setup**, **Sign**, and **Verify**.

- **Setup**$(1^\lambda, w, n, d, \ell, t)$ For a security parameter λ, the setup algorithm generates a bilinear group $\mathcal{G}$, and picks a hash function $H : \{0,1\}^* \to \mathbb{Z}_p$. The setup algorithm generates a signing parameter (SP) as follows:

  1. Pick random elements $x_i, y_i \in \mathbb{Z}_p$ for $i \in [1, n]$ and set $X_i = g^{x_i}, Y_i = g^{y_i} \in \mathbb{G}_1$ for $i \in [1, n]$.
  2. Generate a random element $g_1 \in \mathbb{G}_1$.
  3. Output $\mathsf{SP} = \left(g, g_1, H, \{X_i, Y_i\}_{i \in [1,n]}\right)$.

  Let n be the length of a fuzzy (biometric) data w, and d be the number of entire subsets, ℓ be the number of elements included in each subset, and t be the maximum number of errors among n elements, indicating an error tolerance rate. Given a fuzzy (biometric) data $w = (w[1], \ldots, w[n]) \leftarrow W$, the setup algorithm generates a verification key VK_w as follows:

  1. Randomly select a set $I_j \subset \{1, \ldots, n\}$ where $|I_j| = \ell$ for $j \in [1, d]$.
  2. Set a subset $\overrightarrow{v_j} = \{w[i] | i \in I_j\}$ of a vector w for $j \in [1, d]$.

3. Select random elements $r_j \in \mathbb{Z}_p$ for $j \in [1, d]$.
4. Set $vk_j = \left(\left(\prod_{i \in I_j} h^{x_i + w[i] y_i} \right)^{r_j}, h^{r_j} \right) \in \mathbb{G}_2^2$ for $j \in [1, d]$.

The verification key VK_w is given by

$$\mathsf{VK}_w = \left(\{vk_1, \ldots, vk_d\}, \{I_1, \ldots, I_d\} \right).$$

- **Sign**(SP, w', m)**.** To sign a message m under fuzzy (biometric) data $w' = (w'[1], \ldots, w'[n]) \leftarrow W$, the signing algorithm generates a signature $\sigma_{w'}$ as follows:

1. Choose random elements $s, k \in \mathbb{Z}_p^*$.
2. Set $\sigma_{(1,i)} = \left(X_i Y_i^{w'[i]} \right)^s$ for $i \in [1, n]$.
3. Set $\sigma_2 = g^s \in \mathbb{G}_1$.
4. Set $\sigma_3 = g_1^s \in \mathbb{G}_1$.
5. Set $\sigma_4 = H(g^k, g_1^k, \sigma_2, \sigma_3, \{\sigma_{(1,i)}\}_{i \in [1,n]}, m)$
6. Set $\sigma_5 = k + \sigma_4 \cdot s \in \mathbb{Z}_p$.
7. Output $\sigma_{w'} = \left(\{\sigma_{(1,i)}\}_{i \in [1,n]}, \sigma_2, \sigma_3, \sigma_4, \sigma_5 \right)$.

- **Verify**$(\mathsf{VK}_w, \sigma_{w'}, m)$**.** To verify a signature $\sigma_{w'} = \left(\{\sigma_{(1,i)}\}_{i \in [1,n]}, \sigma_2, \sigma_3, \sigma_4, \sigma_5 \right)$ on a message m under the verification key $\mathsf{VK}_w = \left(\{vk_1, \ldots, vk_d\}, \{I_1, \ldots, I_d\} \right)$, the verification algorithm proceeds as follows:

1. Set $vk_j = (vk_{1,j}, vk_{2,j})$ for $j \in [1, d]$.
2. Set $cnt = 0$.
3. While $cnt = 0$ for $j = 1, \ldots, d$:

   (1) Compute $A = \prod_{i \in I_j} \sigma_{(1,i)}$.
   (2) If $e(\sigma_2, vk_{1,j}) = e(A, vk_{2,j})$, $cnt = 1$.

   If $cnt = 0$, output 0.
4. Otherwise, set $B = g^{\sigma_5} \sigma_2^{-\sigma_4}$ and $B_1 = g_1^{\sigma_5} \sigma_3^{-\sigma_4}$.
5. If $\sigma_4 = H\left(B, B_1, \sigma_2, \sigma_3, \{\sigma_{(1,i)}\}_{i \in [1,n]}, m\right)$, output 1; otherwise 0.

4.2. Setting the Number of Subsets

Let δ be the probability that the **Verify** algorithm outputs 0, meaning that it fails to verify a signature $\sigma_{w'}$ using VK_w. Thus, if $\delta = 1/2$ is set, then a signer would produce a signature two times with overwhelming probability of generating a valid signature. Following [17], we show that the number d of subsets is determined by setting the value δ. Given (n, ℓ), we assume that our FVS scheme wants to tolerate at most t errors among n elements. During verification, the probability that $e(\sigma_2, vk_{1,j}) = e(A, vk_{2,j})$ is at least $(1 - \frac{t}{n})^\ell$ for each $j = 1, \ldots, d$. Thus, the probability that the **Verify** algorithm outputs 0 is at most $\left(1 - (1 - \frac{t}{n})^\ell\right)^d$, meaning that no (biometric) vector $\overrightarrow{v}_j$ matches the vector corresponding with w'. If we set the failure probability δ as

$$\left(1 - \left(1 - \frac{t}{n}\right)^\ell\right)^d \approx \delta,$$

the approximation $e^x \approx x + 1$ gives the relations, such as $\left(1 - \left(1 - \frac{t}{n}\right)^\ell\right)^d \approx \left(1 - e^{-\frac{t\ell}{n}}\right)^d \approx \exp\left(-de^{-\frac{t\ell}{n}}\right)$. Consequently, we obtain the relation

$$d \approx e^{\frac{t\ell}{n}} \cdot \ln \frac{1}{\delta},$$

as required to determine the number d of subsets.

4.3. Correctness

Assume that $\mathbf{dis}(w, w') = \gamma \leq t$. Then, the probability that $e(\sigma_2, vk_{1,j}) = e(A, vk_{2,j})$ for any $j \in [1, d]$ is at least $(1 - \frac{\gamma}{n})^{\ell}$. In this case, since $(1 - \frac{\gamma}{n})^{\ell} \geq (1 - \frac{t}{n})^{\ell}$ because of $\gamma \leq t$, we can derive the probability that the **Verify** algorithm outputs 0 as

$$\left(1 - \left(1 - \frac{\gamma}{n}\right)^{\ell}\right)^{d} \leq \left(1 - \left(1 - \frac{t}{n}\right)^{\ell}\right)^{d} \approx \delta,$$

by the above approximation. As a result, if $\mathbf{dis}(w, w') = \gamma \leq t$, $\Pr[\mathbf{Verify}(\mathsf{VK}_w, \sigma_{w'}, m) = 1] \geq 1 - \delta$.

4.4. Security

Theorem 1. *If $\mathcal{W}$ is a family of sources over Z^n with α-entropy ℓ-samples, then the FVS scheme is VK-private for such a $\mathcal{W}$.*

Proof. Before proving the VK-privacy, we first prove that a family $\mathcal{H}$ of functions $\{f_{\mathsf{SP}} : \mathcal{Z}^{\ell} \to \mathbb{G}_2\}$ generating the partial verification key is a $1/p$-universal hash function for a fixed SP. In our FVS scheme, given a vector $\overrightarrow{v_j} = \{w[i] | i \in I_j\}$ as input, the function $f_{\mathsf{SP}}(\overrightarrow{v_j})$ is defined as follows:

$$f_{\mathsf{SP}}(\overrightarrow{v_j}) = \Big(\prod_{i \in I_j} h^{x_i + w[i] y_i}\Big)^{r_j},$$

where r_j is a randomly chosen exponent $\in \mathbb{Z}_p$. Since r_j is raised for each input, it is easy to see that two outputs of the function could be equal with probability $1/p$. Thus, it holds that for two distinct inputs $\overrightarrow{v_i}$ and $\overrightarrow{v_j}$, $\Pr\left[f_{\mathsf{SP}}(\overrightarrow{v_i}) = f_{\mathsf{SP}}(\overrightarrow{v_j})\right] = \frac{1}{p} = \frac{1}{|\mathbb{G}_2|}$; thus, $\mathcal{H} = \{f_{\mathsf{SP}}\}$ is $1/p$-universal.

Next, we prove that a verification key VK_w corresponding with $w \in W$ is almost close to uniform distribution based on Lemma 3. After that, we extend the result to the case of a polynomial number of verification keys. Let $W = (W[1], ..., W[n])$ be a random variable of a source with α-entropy ℓ-samples. If $V_j = \{W[i] | i \in \mathcal{I}_j\}$, we see that $(V_1, ..., V_d)$ is a joint distribution of d subsets of $W \in \mathcal{W}$ such that $\tilde{\mathbf{H}}_{\infty}[V_j | \mathcal{I}_j] \geq \alpha$ for random sets of indices $(\mathcal{I}_1, ..., \mathcal{I}_d)$. In addition, as mentioned above, each function f_{SP} is a $1/p$-universal hash function with respect to a distinct $j \in [1, d]$.

By Lemma 2(a), if a random set of indices $\mathcal{I}_j$ is set to be I_i, then $\mathbf{H}_{\infty}[V_j | \mathcal{I}_j = I_j] \geq \tilde{\mathbf{H}}_{\infty}[V_j | \mathcal{I}_j] - \log(1/\xi)$ with a probability of at least $1 - \xi$. Let $f_{\mathsf{SP}}(V_j)$ be a random variable for a set V_j of indices for $j \in [1, d]$ and let $(y_{j+1}, \ldots, y_d)$ be a sample of $(f_{\mathsf{SP}}(V_{j+1}), \ldots, f_{\mathsf{SP}}(V_d))$. By a similar proof A1 (proof of Lemma 3), we obtain the following result:

$$\begin{aligned}
&\tilde{\mathbf{H}}_{\infty}[V_j | \mathcal{I}_j, f_{\mathsf{SP}}, \ldots, f_{\mathsf{SP}}, \{f_{\mathsf{SP}}(V_k) = y_k\}_{k=j+1}^{d}] \\
&\geq \tilde{\mathbf{H}}_{\infty}[V_j | \mathcal{I}_j, f_{\mathsf{SP}}, \ldots, f_{\mathsf{SP}}] - (d - j) \log |\mathbb{G}_2| - \log(1/\epsilon) \\
&= \tilde{\mathbf{H}}_{\infty}[V_j | \mathcal{I}_j] - (d - j) \log |\mathbb{G}_2| - \log(1/\epsilon) \\
&\geq \alpha - (d - j) \log |\mathbb{G}_2| - \log(1/\epsilon),
\end{aligned}$$

where ϵ shows negligible probability determined by a security parameter. As a result, if $\alpha \geq d \cdot \log |\mathbb{G}_2| + 3 \log(1/\epsilon) + \log(1/\xi) + \Theta(1)$, we have

$$\begin{aligned}
&\mathbf{H}_{\infty}[V_j | \mathcal{I}_j = I_j] \\
&\geq \alpha - (d - j) \log |\mathbb{G}_2| - \log(1/\epsilon) - \log(1/\xi) \\
&= j \log |\mathbb{G}_2| + 2 \log(1/\epsilon) + \Theta(1) \\
&\geq \log |\mathbb{G}_2| + 2 \log(1/\epsilon) + \Theta(1).
\end{aligned}$$

For $j \in [1,d]$, we can show that $\mathbf{H}_\infty[V_j|\mathcal{I}_j = I_j] \geq \log|\mathbb{G}_2| + 2\log(1/\epsilon) + \Theta(1)$, similarly as in the above process.

Let $EXP.real = (f_{\mathsf{SP}}(\overrightarrow{v_1}), \ldots, f_{\mathsf{SP}}(\overrightarrow{v_d}))$ for some samples $(V_1 = \overrightarrow{v_1}, \ldots, V_d = \overrightarrow{v_d})$, and let $EXP.rand = (u_1, \ldots, u_d)$ be random samples chosen from $\mathbb{G}_2^d$. Because of the above inequality, under the assumption that $\alpha \geq d \cdot \log|\mathbb{G}_2| + 3\log(1/\epsilon) + \log(1/\xi) + \Theta(1)$, we see via Lemma 2.3 that, for any $(V_1, \ldots, V_d)$ and $(f_{\mathsf{SP}}, \ldots, f_{\mathsf{SP}})$, the statistical distance between $EXP.real$ and $EXP.rand$ is less than $2\epsilon d$.

We can extend the above result of a single verification key to correlated random variables $W = (W_1, \ldots, W_q)$. Let $(V_{i1}, \ldots, V_{id})$ be a joint distribution of d subsets $(I_{i1}, \ldots, I_{id})$ for $i \in [1,q]$. If $\alpha \geq d \cdot q \cdot \log|\mathbb{G}_2| + 3\log(1/\epsilon) + \log(1/\xi) + \Theta(1)$, the statistical distance between $EXP.real = (\mathsf{VK}_{w_1}, \ldots, \mathsf{VK}_{w_k}, \ldots, \mathsf{VK}_{w_q})$ and $EXP.rand = (\mathsf{VK}_{w_1}, \ldots, R_k, \ldots, \mathsf{VK}_{w_q})$ is less than $2\epsilon(q \cdot d - (q \cdot d - d + 1) + 1) = 2\epsilon d$ for any $k \in [1,q]$ where $\mathsf{VK}_{w_i} = (f_{\mathsf{SP}_i}(\overrightarrow{v_{i1}}), \ldots, f_{\mathsf{SP}_i}(\overrightarrow{v_{id}}))$ and $R_k = (u_1, \ldots, u_d)$ for uniformly random $u_i \in \mathbb{G}_2^d$. □

Theorem 2. *If the FVS scheme is VK-private and the XDH assumption holds in $\mathbb{G}_1$, the FVS scheme is SIG-private in the random oracle model.*

In reality, if verification keys and signatures were to become revealed to an adversary, we would have to prove that it is infeasible for the adversary to glean any information on the fuzzy (biometric) data from the signatures. As mentioned above, however, we prove from Theorem 4.1 that the VK-privacy holds, meaning that it is difficult for the adversary to get some information on the fuzzy data from verification keys. In addition, the setup algorithm chooses a new signing parameter SP each time a verification key is generated for each fuzzy data. Thus, it is sufficient to show that a signature generated under a fixed SP does not reveal any information about a challenged fuzzy data.

To do this, a simulator chooses a challenge sample w^* along with other correlated samples. For the length n of w^*, we create the following sequence of games where w^* is used for generating a challenge signature:

$$\begin{aligned}
&\text{Game } 0 : w^* = (w^*[1], w^*[2], \ldots, w^*[n])\\
&\text{Game } 1 : w^* = (R_1, w^*[2], \ldots, w^*[n])\\
&\vdots\\
&\text{Game } \alpha : w^* = (R_1, \ldots, R_\alpha, w^*[\alpha+1], \ldots, w^*[n])\\
&\vdots\\
&\text{Game } n : w^* = (R_1, \ldots, R_n).
\end{aligned}$$

In Game 0, the challenge signature is generated for the original w^* as in a real game, whereas in Game n the challenge signature is generated for a random vector and thus does not have any information on the original w^*. For proving the SIG-privacy, it is sufficient to show that it is infeasible for the adversary to distinguish between Game $(\alpha - 1)$ and Game α under the DDH problem.

Lemma 4. *Under the XDH assumption in $\mathbb{G}_1$, it is infeasible to distinguish between Game (α-1) and Game α.*

Proof. Given an XDH instance $(g, h, g^a, g^b, T) \in \mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{G}_1^3$, a challenger $\mathcal{C}$ interacts with an adversary $\mathcal{A}$ who tries to break the the SIG-privacy of our FVS scheme.

- **Setup.** $\mathcal{C}$ chooses samples $(w^*, w_1, \ldots, w_q) \leftarrow (W^*, W_1, \ldots, W_q)$, where those $q+1$ correlated distributions are from $\mathcal{A}$. In particular, for a sample $w^* = (w^*[1], \ldots, w^*[n])$ corresponding to W^*, $\mathcal{C}$ sets $w^*[1] = R_1, \ldots, w^*[\alpha - 1] = R_{\alpha-1}$ for randomly chosen values $R_1, \ldots, R_{\alpha-1} \in \mathbb{Z}_p$. For each w_j, $\mathcal{C}$ generates $(\mathsf{VK}_{w_j}, \mathsf{SP}_j)$ for $j = 1, \ldots, q$ as follows:

- Choose random values ϕ and $\{x_{ji}, y_{ji}\}_{i=1}^{n}$ in $\mathbb{Z}_q$.
- Set $\mathsf{SP}_j = \{g, g^{\phi}, g^{x_{ji}}, g^{y_{ji}}\}_{i=1}^{n}$.
- Select d sets of random indices I_m such that $|I_m| = \ell$.
- Set a subset $\overrightarrow{v_{jm}} = \{w_j[i] | i \in I_m\}$ for $m \in [1, d]$.
- Choose $\{r_{jm}\}_{m=1}^{d}$ at random in $\mathbb{Z}_p$.
- Set $vk_{jm} = (h^{r_{jm}\left(\sum_{i \in I_m} x_{ji} + y_{ji} w_j[i]\right)}, h^{r_{jm}})$.

For the target w^*, $\mathcal{C}$ chooses $\{x_i^*, y_i^*\}_{i=1}^{n}$ in $\mathbb{Z}_p$, and sets $\{X_i^* = g^{x_i^*}, Y_i^* = g^{y_i^*}\}_{i=1, i \neq \alpha}^{n}$ and $X_\alpha^* = g^a g^{x_\alpha^*}$, $Y_\alpha^* = g^{y_\alpha^*}$, under the implicit setting of $\tilde{x}_\alpha^* = a + x_\alpha^*$. $\mathcal{C}$ then sets $\mathsf{SP}^* = (g, g^{\phi}, \{X_i^*, Y_i^*\}_{i=1}^{n})$ and gives SP^* along with $(\mathsf{VK}_{w_j}, \mathsf{SP}_j)$ for $j = 1, \ldots, q$ to $\mathcal{A}$.

- **Signing queries.** $\mathcal{A}$ issues a pair of a correlated distribution W' with $(W^*, W_1, \ldots, W_q)$, an index $j \in [1, q]$ of signing parameter SP_j, and a message m. Then, $\mathcal{C}$ chooses a sample $w' = (w'[1], \ldots, w'[n]) \leftarrow W'$ correlated with $(w^*, w_1, ..., w_q)$. $\mathcal{C}$ performs the ordinary signature generation algorithm by taking (SP_j, w', m) as inputs.
- **Challenge.** $\mathcal{A}$ sends a message m^* to $\mathcal{C}$. In particular, we use a non-interactive zero-knowledge (NIZK) simulator to generate two elements (σ_4^*, σ_5^*) without knowing the witness. $\mathcal{C}$ generates a challenge signature as follows:

  - Set $\sigma_{(1,i)}^* = (g^b)^{(x_i^* + y_i^* \cdot w^*[i])}$ for $i \in [1, n]$ and $i \neq \alpha$.
  - Set $\sigma_{(1,\alpha)}^* = T(g^b)^{(x_\alpha^* + y_\alpha^* \cdot w^*[\alpha])}$.
  - Set $\sigma_2^* = g^b$.
  - Set $\sigma_3^* = (g^b)^{\phi}$.
  - Obtain (σ_4^*, σ_5^*) using the NIZK simulator.

  $\mathcal{C}$ sends $\sigma_{w^*}^* = (\{\sigma_{(1,i)}^*\}_{i=1}^{n}, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*)$ to $\mathcal{A}$.
- **Guess.** $\mathcal{A}$ outputs a guess $b \in 0, 1$ in response to the challenge signature.

If $T = g^{ab}$, then $\mathcal{A}$ interacts with $\mathcal{B}$ in **Game** $(\alpha - 1)$, because $\sigma_{(1,\alpha)}^* = T(g^b)^{(x_\alpha^* + y_\alpha^* \cdot w^*[\alpha])} = \left(g^{a + x_\alpha^*} (g^{y_\alpha^*})^{w^*[\alpha]}\right)^b = \left(X_\alpha^* (Y_\alpha^*)^{w^*[\alpha]}\right)^b$. Otherwise, $\sigma_{(1,\alpha)}^*$ becomes a random element, in which case $\mathcal{A}$ is in **Game** α. Therefore, depending on the ability of $\mathcal{A}$, $\mathcal{C}$ is able to solve the given XDH problem. □

Theorem 3. *If the FVS is VK-private and SIG-private and the DL assumption holds in $\mathbb{G}_1$, the FVS scheme is reusable in random oracle model.*

The proof for Theorem 4.3 is almost the same as the proof for unforgeability in Reference [20], but the difference resides in the fact that an adversary in our reusability proof is given verification keys and signatures that correspond to correlated fuzzy (biometric) data. In other words, even if such correlated fuzzy data is reused, our proof shows that it is difficult for the adversary to generate a valid signature with (unknown) target fuzzy data. To prove reusability, we need the VK-privacy and SIG-privacy to guarantee that the verification keys and signatures exposed to the adversary do not reveal any information about the fuzzy data. At this point, there are two strategies we anticipate that the adversary might take in their forgery. The first is that the adversary would guess w' from a certain distribution W and then generate a signature on the input (SP, w', m). However, as long as W is assumed to be a distribution with α-entropy ℓ-samples and α is sufficiently large with respect to the security parameter, such a strategy can be successful with negligible probability only.

The other strategy is to reuse a previous signature without changing the fuzzy data w' embedded into it. More specifically, there are two prongs to this strategy. One is to re-randomize the previous signature by raising a new random exponent s' into the elements $\{\sigma_{(1,i}\}_{i=1}^{n}$, σ_2, and σ_3. Thus, the discrete logarithm of such elements becomes $s \cdot s'$, where s was chosen by a signer and s' is

now selected by an adversary. The important point is that the adversary still cannot know the exact discrete logarithm $s \cdot s'$, which is then the witness necessary for generating the other signature elements (σ_4, σ_5) as an NIZK proof as it relates to $s \cdot s'$. However, generating such an NIZK proof equates to breaking the statistical soundness of proving the equality of discrete logarithms [28] in regard to the unknown witness. Thus, the probability that the adversary would succeed is at most q_h/p, where q_h is the number of H-oracle queries, which is also negligible. Now, the remaining case is to reuse the previous signature as it is and simply reconstruct a new proof (σ_4, σ_5). Fortunately, this case can be reduced to the forgery of a one-time multi-user Schnorr signature [29], which is provably secure under the DL assumption. In our proof, a slight variant of the one-time multi-user Schnorr signature proves the equality of discrete logarithms rather than proving knowledge. In line with this, the variant is also provably unforgeable [30,31] against chosen-message attacks in a multi-user setting (MU-SUF-CMA). Based on the variant we use, we show that, under the DL assumption, it is difficult for the adversary to succeed in the remaining case.

Proof. A simulator $\mathcal{B}$ uses an adversary $\mathcal{A}$ (which breaks the reusability of the FVS scheme) as a subroutine to forge a signature in the one-time multi-user Schnorr signature scheme. Let $q_s \leq \rho$ for the number q_s of signature queries. Given ρ independent public keys $\left(g, g_1, \{(g^{s_1}, g_1^{s_1}), \ldots, (g^{s_\rho}, g_1^{s_\rho})\}\right)$ of the one-time multi-user Schnorr signature scheme, $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

- **Setup.** $\mathcal{A}$ gives correlated random variables $(W_1, \ldots, W_q) \in \mathcal{W}^q$, of which each is in $\mathcal{Z}^n$. $\mathcal{B}$ samples $(w_1, \ldots, w_q) \leftarrow (W_1, \ldots, W_q)$ and runs **Setup**$(1^\lambda, n, d, \ell, w_j, t)$ to obtain a signing parameter SP_j and a verification key VK_{w_j} for $j = 1, \ldots, q$. In the setup algorithm, $(x_{j1}, \ldots, x_{jn})$ and $(y_{j1}, \ldots, y_{jn})$ are selected uniformly at random in $\mathbb{Z}_p$. $\mathcal{B}$ gives $\{(\mathsf{SP}_j, \mathsf{VK}_{w_j})\}_{j=1}^q$ to $\mathcal{A}$.
- **Signing queries.** For $j \in [1, q]$, $\mathcal{A}$ issues signing queries with input, namely a random variable W' correlated with $(W_1, \ldots, W_q)$, an index of signing parameter SP_j, and a message m_k. $\mathcal{B}$ responds to the query as follows:
  - Choose a sample $w' = (w'[1], \ldots, w'[n])$ correlated with $(w_1, \ldots, w_q)$ from W'.
  - Generate $\sigma_{1,i} = (g^{s_k})^{x_{ji} + y_{ji} w'[i]}$ for $i \in [1, n]$, and set $M_k = \left(g^{s_k}, g_1^{s_k}, \{\sigma_{1,i}\}_{i \in [1,n]}, m_k\right)$.
  - Query (j, M_k) to the signing oracle of the one-time multi-user Schnorr signature scheme, meaning a signing query on a message M_k under the j-th public key, and receive (h_k, c_k).
  - Set $\sigma_k = \left(\{\sigma_{1,i}\}_{i \in [1,n]}, g^{s_k}, g_1^{s_k}, h_k, c_k\right)$ and give σ_k to $\mathcal{A}$.
- **Output.** $\mathcal{A}$ outputs $(m^*, \sigma^*) = \left(m^*, \left(\{\sigma_{1,i}^*\}_{i \in [1,n]}, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*\right)\right)$. $\mathcal{B}$ checks if $(m^*, \sigma^*) \neq (m_k, \sigma_k)$ for any $k \in [1, q_s]$, and finds k^* such that $\sigma_2^* = g^{s_{k^*}}, \sigma_3^* = g_1^{s_{k^*}}$. After finding the index j corresponding with the k^*-th query, $\mathcal{B}$ checks if **Verify**$(\mathsf{VK}_{w_j}, \sigma^*, m^*)$ outputs 1. If it does, $\mathcal{B}$ outputs a forgery of the one-time multi-user Schnorr signature scheme as follows:
  - Set $M^* = \left(\sigma_2^*, \sigma_3^*, \{\sigma_{1,i}^*\}_{i \in [1,n]}, m^*\right)$.
  - Output $\left(k^*, M^*, (\sigma_4^*, \sigma_5^*)\right)$.

It follows that, as longs as $\mathcal{A}$ breaks the reusability of the FVS scheme, then $\mathcal{B}$ can break the MU-SUF-CMA security of the Schnorr signature scheme. Indeed, if **Verify**$(\mathsf{VK}_{w_j}, \sigma^*, m^*) = 1$, this means that (σ_4^*, σ_5^*) is the valid signature of the message $\left(\sigma_2^*, \sigma_3^*, \{\sigma_{1,i}^*\}_{i \in [1,n]}, m^*\right)$ under the k^*-th public key (σ_2^*, σ_3^*).

Since it is clearly proven that the variant of the one-time multi-user Schnorr signature scheme is MU-SUF-CMA secure under the DL assumption in the random oracle model [29], the reusability of our FVS scheme can also be proven in the random oracle model under the DL problem. □

5. Performance Analysis

Our FVS scheme is constructed using the subset-based sampling method of the re-usable fuzzy extractor [17], which does not require fuzzy (biometric) input data to have sufficiently high min-entropy and can still tolerate a sub-linear fraction of errors. Generally, biometric data sources are non-uniformly and uniquely distributed from person to person; thus, it is not easy to expect that such biometric data will have high min-entropy. Nevertheless, most reusable fuzzy extractors [8,13–15] based on secure sketches require source data with high min-entropy because a secure sketch is known to cause an entropy loss of a biometric input data. For comparison, we focus on fuzzy extractors [17,18] that do not rely on secure sketches, and therefore not require a high min-entropy source. A reusable fuzzy extractor suggested by Reference [12] is not based on a secure sketch and tolerates a linear fraction of errors, but it uses a so-called pseudoentropic isometry that requires each biometric input data component to have high min-entropy. This requirement is also far from realistic biometric information. A reusable fuzzy extractor suggested by Apon et al. [16] is constructed based on the hardness of Learning with Errors (LWE) problems, where biometric data is injected into the part with LWE errors. In that case, such biometric data must follow a certain error distribution (e.g., Gaussian) in order to ensure the security of LWE problems. This reveals a limitation in its real-world application potential. Furthermore, the LWE-based fuzzy extractor [16] requires a time-consuming reproducing algorithm, where another sample of biometric data is subtracted from an LWE instance that was previously created per each component, and then a randomly chosen linear system is expected to be solved. A problem arises if any error components in the chosen linear system are not 0, at which point a new linear system must be randomly reselected until achieving success. The same problem is found in the reusable fuzzy signature [19] that follows the LWE-based fuzzy extractor technique. To mitigate the reproducing problem, [16,19] should be limited to dealing with only a logarithmic size of errors.

Eventually, we compare our FVS scheme with previous fuzzy extractors [17,18] and the original FVS scheme [20], which all follow the subset-based sampling technique [17]. We specifically consider authentication protocols where a fuzzy extractor or an FVS scheme is instantiated to authenticate a user using biometric data. During protocol executions, we compare our proposed scheme with existing schemes in terms of storage or transmission costs and computational costs on the part of the user. In regards to the fuzzy extractor, we assume that a digital signature scheme $\mathcal{S} = (\mathbf{KeyGen}, \mathbf{Sign}, \mathbf{Verify})$ is additionally provided. As usual, an authentication protocol consists of two phases; enrollment and authentication. In the enrollment phase, a user is registered with an authentication server by sending an identity ID, a verification key vk_{ID}, and helper data P_{ID}. In each authentication phase, a user receives helper data P_{ID} from the server, recovers a secret key for signature generation using their biometric data, and returns a signature σ_{ID} in response to a challenge message R sent by the server. Figures 1 and 2 present two authentication protocols in more detail.

5.1. Storage or Transmission Costs

In the authentication phase, helper data P_{ID} is needed to generate a signing key from user biometric data. There are two options through which the user obtains an P_{ID}. The first way is to store it in their own personal device, and the other is to receive it from the server for each instance of authentication. The first method can reduce the amount of transmission, while carrying a personal storage device. Conversely, the second one does not require a personal device, while the server sends out a huge amount of transmission, and it has the advantage in that the user authentication can work on secure devices that are shared by multiple users. For comparison purposes, we present the size of P_{ID} as the storage or transmission cost in Table 2.

Table 2. Comparison of storage or transmission costs with helper data.

	n	Error Tolerance Rate (t/n)	# of Components In Each Subset (ℓ)	# of Subsets (d)	Helper Data (P_{ID}, SP)
[17]	512	20%	80	61.6×10^5	0.83 GB
	1024		80	61.6×10^5	0.90 GB
	2048		80	61.6×10^5	0.96 GB
[18]	512	20%	16 ($\times 5$)	53.6×10^3	3.76 MB
	1024		20 ($\times 4$)	26.3×10^3	2.01 MB
	2048		27 ($\times 3$)	11.6×10^4	10.16 MB
[20]	512	20%	80	61.6×10^5	0.05 MB
	1024		80	61.6×10^5	0.09 MB
	2048		80	61.6×10^5	0.19 MB
Ours	512	20%	80	61.6×10^5	0.03 MB
	1024		80	61.6×10^5	0.06 MB
	2048		80	61.6×10^5	0.13 MB

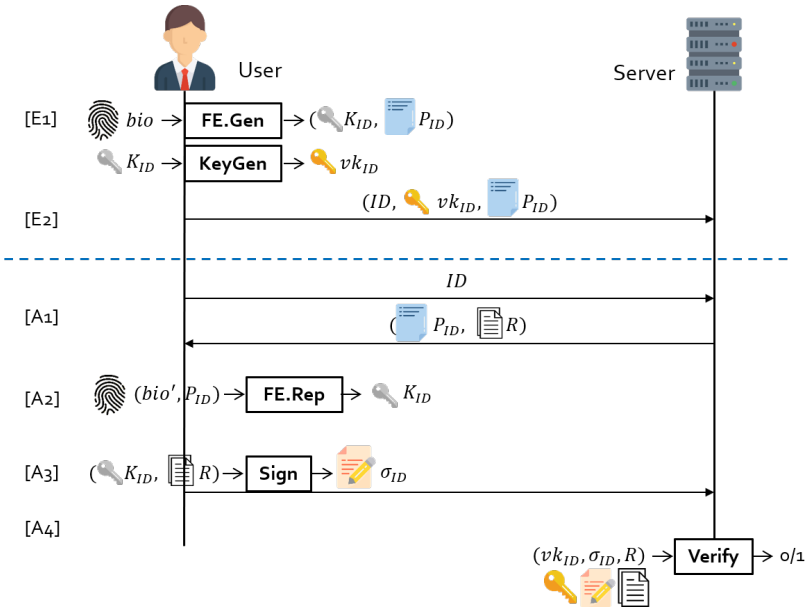

Figure 1. Authentication with a fuzzy extractor.

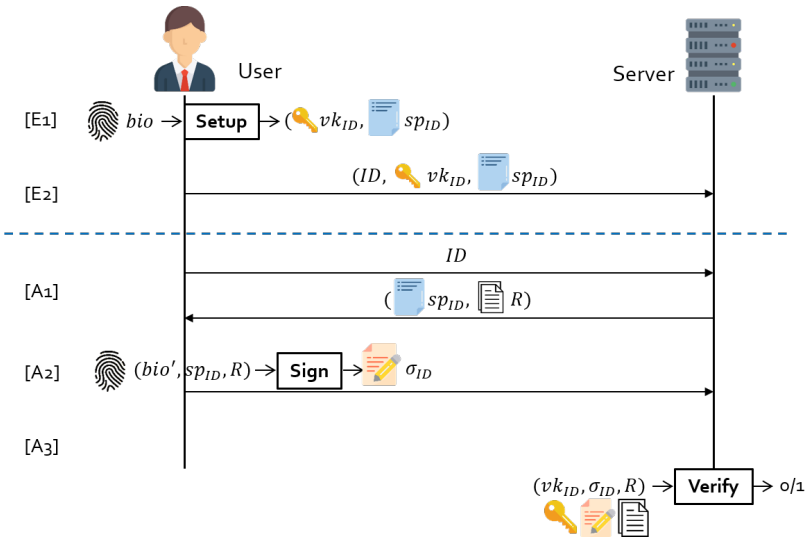

Figure 2. Authentication with our fuzzy vector signature (FVS).

Let n be the dimension of biometric data, d be the number of subsets, ℓ be the number of elements in each subset, and t be the maximum number of errors among n elements. With fuzzy extractors [17,18], helper data P_{ID} consists of an information set, a nonce, and an output of a hash function per subset. To begin, the number d of subsets is obtained via $d \approx \ln(1/\delta) \cdot e^{\frac{t\ell}{n}}$ in Reference [17], whereas d is computed as $d \approx m/2q$ in Reference [18], where $q = \frac{\tau\binom{m}{\tau}}{\binom{n}{t}} \sum_{\eta=\tau}^{m} (-1)^{\eta-\tau} \cdot \frac{\binom{m-\tau}{\eta-\tau} \times \binom{n-\tau\ell}{\tau}}{t}$ for (τ, m)-threshold scheme [32]. In this case, the information set per subset has ℓ indices which are represented by $\ell \log n$ bits. When using SHA-256 as the hash function, we set the size of a nonce to be sufficient at $176 (= 256 - 80)$ bits. As a result, the whole size of P_{ID} is about $d \cdot (\ell \log n + 256 + 176)$ for n =512, 1024, and 2048 cases, which, as shown in Table 2, becomes huge when setting $t/n = 0.20$ as an error tolerance rate and $\ell = 80$. On the other hand, with FVS schemes, the number of subsets is determined by $d \approx \ln(1/\delta) \cdot e^{\frac{t\ell}{n}}$ following [17], but helper data P_{ID} consists of only a signing parameter SP, regardless of the number of subsets. Indeed, in Figure 3, SP in Reference [20] is $3n + 1$ elements in $\mathbb{G}_1$, whereas SP in ours is $2n + 2$ elements in $\mathbb{G}_1$. When taking the Type-3 pairing [33] at the 100-bit security level, the size of elements in $\mathbb{G}_1$ and $\mathbb{Z}_p$ is 256 bits. For $n = 512$, 1024, and 2048 cases, Table 2 shows that the P_{ID} size of the FVS schemes are overwhelmingly smaller than that of the fuzzy extractors. Compared to Reference [20], our FVS scheme obtains a slightly shorter size of SP with the same parametrization. Regarding the signature size, the σ_{ID} in our scheme consists of $n + 2$ elements in $\mathbb{G}_1$ plus 2 elements in $\mathbb{Z}_p$ that are transmitted to the server. When taking the Type-3 paring and $n = 512$ again, the amount of σ_{ID} transmission is about $(512 + 4) \times 256/2^{13} = 16$ KB. $\delta = 1/2$ is the the probability that verification fails, and we expect the user to run step [A2] of the authentication protocols twice, during which the transmission cost of σ_{ID} becomes about 32 KB.

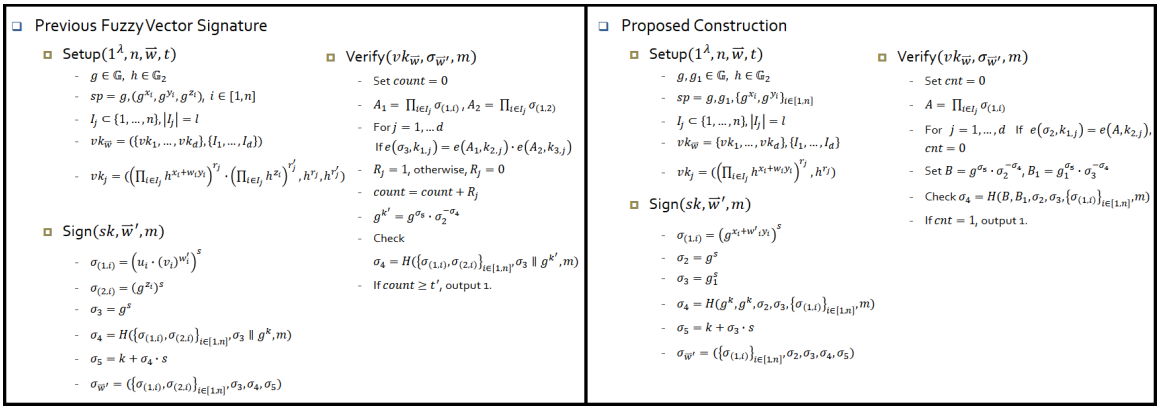

Figure 3. Comparison between previous and our construction.

5.2. Computation Cost

For the fuzzy extractor, we considered the computational cost required for obtaining a signing key K_{ID} by running the **FE.Rep** algorithm that takes as input helper data P_{ID} and a biometric data bio'. This is the step for [A2] in Figure 1. We assume that a reproduced value from **FE.Rep** is straightforwardly used as the singing key corresponding with the verification key vk_{ID}. If a hash function H, such as SHA-256, is used as a digital locker [17], K_{ID} is locked with $(\text{nonce}_i, H(\text{nonce}_i, \{bio\}_i) \bigoplus (K_{ID} || 0^s))$ for a positive integer s, where $\{bio\}_i$ is the set of biometric data corresponding to a subset i among n components. Therefore, with a new bio', the unlocking algorithm needs to perform a hash computation $|H|$ plus an XOR operation $|X|$ per each subset i until K_{ID} is obtained. Consequently, the **FE.Rep** algorithm in Reference [17] must compute $d \cdot (|\mathrm{H}| + |X|)$ operations as a worst case scenario. Since [18] also requires solving a (τ, m)-secret sharing scheme, the **FE.Rep** algorithm chooses a set of τ shares among m unlocked values and then solves a secret-sharing scheme, leading to performing additional $\frac{d}{m}\left(\binom{m}{\tau} \cdot \tau m(m-1)\right)|X|$ operations. In contrast, the FVS scheme needs to run the **FVS.Sign** algorithm to generate a signature σ_{ID} by taking

(bio', SP, R). This is the step for [A2] in Figure 2. In our FVS scheme, the **FVS.Sign** algorithm needs $(n+4)$ exponentiations in $\mathbb{G}_1$ for the dimension n of biometric data. Compared to Reference [20], the signing operation is reduced by about half, which is shown in Table 3.

Table 3. Comparison of computational costs necessary for signature generation.

	Operation
[17]	$d \cdot (\lvert H\rvert + \lvert X\rvert)$ + **Sign**
[18]	$\frac{d}{m}\left(\binom{m}{\tau} \cdot \tau m(m-1)\right)\lvert X\rvert + d \cdot (\lvert H\rvert + \lvert X\rvert)$ + **Sign**
[20]	$(2n+2)\lvert E\rvert$
Ours	$(n+4)\lvert E\rvert$

∘ $|H|$: Hash, $|X|$: XOR, $|E|$: Exponentiation.

In order to measure the actual amount of calculation, we considered how much computation the user should perform by substituting the numbers in Table 2 directly. For instance, let $n = 512$ and $\ell = 80$. The fuzzy extractor by Canetti et al. [17] must complete $(61.6 \times 10^5)(|H| + |X|)$ operations, and when setting $(\tau, m) = (5, 32)$, the fuzzy extractor by Cheon et al. [18] needs to compute approximately $(53.6 \times 10^3)|H| + (16.7 \times 10^{11})|X|$ operations—which still seems to be a burdensome amount of calculation to perform on a personal device. To compare, when considering 0.3 ms for an exponentiation in $\mathbb{G}_1$ [34], the **FVS.Sign** algorithm in ours takes about $(512+4) \cdot 0.3 \approx 155$ ms. Since $\delta = 1/2$, the user is required to run the step [A2] of the authentication protocols twice, such that the **FVS.Sign** algorithm, which is computed by the user, takes about 310 ms.

5.3. Implementation

We implemented our fuzzy vector signature as a C program to measure actual time consumption. All implementations are performed on an Intel Core i7-8700k with 8GB RAM running Ubuntu 18.04 LTS, and GNU GCC version 7.5.0 is used for the compilation. For implementation, we selected the BLS12381 curve that offer around a 128-bit security level and a SHA-256 for a hash function. In a BLS12381 curve, the size of a element of $\mathbb{G}_1$ and $\mathbb{G}_2$ are 192 bit and 384 bit, respectively. Our implementation codes are available to https://github.com/Ilhwan123/FVS.

For each error rate 5% and 10%, we measured the time consumption required for running our scheme several times. Table 4 presents parameter setting, storage size, and time required for each algorithm, which is the results of our implementation. Compared with Table 2, the size of the $\mathbb{G}_1$ group element is 192 bits, which is less than 256 bits. Therefore, the size of a signing parameter is smaller. However, this depends on the curve type, so it can be different depending on which curve is selected.

Note that, if signature verification fails in the parameter setting in Table 4, the total verification time takes about 56 s.

Table 4. Implementation results of FVS using BLS12381 curve in case of error tolerance rate (t/n) : 12.5%, the number of components in each subset (ℓ) : 80, and the number of subsets (d) : 15,268 with $\delta = 1/2$.

n	Signing Parameter	Verification Key	Signature	Setup (s)	Sign (ms)	Verify (s)	Error Rate *bio'*
512	24.04 KB	1.95 MB	12.10 KB	23.97	133.86	0.302 17.954	5% 10%
1024	48.04 KB	1.95 MB	24.11 KB	24.47	264.88	0.254 21.830	5% 10%
2048	96.04 KB	1.95 MB	48.10 KB	25.92	532.01	0.377 21.364	5% 10%

• Signing Parameter, Verification Key, and Signature means the size of them, respectively, for each fuzzy (biometric) data length n. • **Setup**, **Sign**, and **Verify** means the average time required for each algorithm. • Error rate *bio'* means a percent of difference between a input data of **Setup** *bio* and a input data of **Sign** *bio'*.

6. Conclusions

In this paper, we presented an FVS scheme improved upon across all aspects of efficiency and security that is, more strictly speaking, based on the subset-based sampling method [17]. Compared to the original FVS scheme [20], we reduced the size of the signing parameter and the verification key to approximately two-thirds their original sizes and cut the signature size by about half. In addition, we reduced the number of pairings necessary for signature verification to about two-thirds the original number.

We prove that our FVS scheme is VK-private and SIG-private, meaning that the verification key and signatures generated using user correlated fuzzy (biometric) data do not reveal any information about the fuzzy input data. Additionally, instead of the unforgeability of Reference [20], we define the reusuability property which guarantees that a user is able to reuse their fuzzy correlated (biometric) data to generate polynomially-many verification keys, all while still making it infeasible for an adversary to forge a signature without any fuzzy (biometric) data. Under the reusability notion, we can prove that our FVS scheme is reusable, assuming that our FVS scheme is $\{\mathrm{VK}, \mathrm{SIG}\}$-private and the DL assumption holds.

In the remote authentication protocol of our FVS scheme, a user must receive the signing parameter and transmit a signature in response to a random challenge message. The primary advantage of FVS-based (biometric) authentication is that the transmission cost, including the signing parameter and signature, is determined only by the number of dimensions with respect to the fuzzy (biometric) data, not by the number of entire subsets. Thus, unlike the authentication protocol with a fuzzy extractor, the transmission cost between the user and the authentication server becomes remarkably smaller. However, the disadvantage of our FVS-based authentication scheme is that the server is required to perform pairing operations by the number of entire subsets, which is the worst case scenario. Such a burden may be somewhat alleviated by utilizing the computing power of the server in parallel, but it could be more desirable to build a new FVS scheme that supports efficient batch verification operations in the future.

Author Contributions: Conceptualization, J.H.P.; Formal analysis, I.L., and D.H.L.; methodology, I.L., and M.S.; validation, M.S., and J.H.P.; writing—original draft preparation, I.L.; writing—review and editing, M.S., and J.H.P.; supervision, D.H.L.; project administration, D.H.L.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2016-6-00600, A Study on Functional Encryption: Construction, Security Analysis, and Implementation).

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

Appendix A. Proof of Lemma 3

Proof. Let $(H_1, \ldots, H_q) \leftarrow \mathcal{H}^q$ and $(U_1, \ldots, U_q)$ be q independent and uniform distributions over V. Let $\mathcal{D} = (H_1, H_1(X_1), \ldots, H_q, H_q(X_q))$ and $\mathcal{D}_i = (H_1, U_1, \ldots, H_i, U_i, H_{i+1}, H_{i+1}(X_{i+1}), \ldots, H_q, H_q(X_q))$ be two distributions for $i \in [1, q]$. By mathematical induction, we prove that D and D_i are $2\epsilon i$-close for $i \in [1, q]$.

For $i = 1$, Lemma 2 shows that with a probability of at least $1 - \epsilon$ over the sample of $(y_2, \ldots, y_q) \leftarrow (H_2(X_2), \ldots, H_q(X_q))$, it holds that

$$\begin{aligned}
&\mathbf{H}_\infty[X_1 | H_2, \ldots, H_q, \{H_i(X_i) = y_i\}_{i=2}^q] \\
&\geq \mathbf{H}_\infty[X_1 | H_2, \ldots, H_q] - (q-1)\log|V| - \log(1/\epsilon) \\
&= \mathbf{H}_\infty[X_1] - (q-1)\log|V| - \log(1/\epsilon) \\
&\geq \alpha - (q-1)\log|V| - \log(1/\epsilon) \\
&= \log|V| + 2\log(1/\epsilon) + \Theta(1).
\end{aligned}$$

In this case, the leftover hash lemma [35] implies that the two distributions $\mathcal{D}$ and $\mathcal{D}_1$ are 2ϵ-close.

Next, assuming that the above lemma holds for $i-1 < q$, we show that the case for i also holds. Lemma 2 shows that, with a probability of at least $1-\epsilon$ over the sample of $(y_{i+1}, \ldots, y_q) \leftarrow (H_{i+1}(X_{i+1}), \ldots, H_q(X_q))$, it holds that

$$\begin{aligned}
&\mathbf{H}_\infty[X_i | H_{i+1}, \ldots, H_q, \{H_j(X_j) = y_j\}_{j=i+1}^{q}] \\
&\geq \mathbf{H}_\infty[X_i | H_{i+1}, \ldots, H_q] - (q-i)\log|V| - \log(1/\epsilon) \\
&= \mathbf{H}_\infty[X_i] - (q-i)\log|V| - \log(1/\epsilon) \\
&\geq \alpha - (q-i)\log|V| - \log(1/\epsilon) \\
&= i\log|V| + 2\log(1/\epsilon) + \Theta(1) \\
&\geq \log|V| + 2\log(1/\epsilon) + \Theta(1).
\end{aligned}$$

Similarly, the leftover hash lemma [35] shows that the two distributions $\mathcal{D}_{i-1}$ and $\mathcal{D}_i$ are 2ϵ-close. It follows that $\Delta(\mathcal{D}, \mathcal{D}_i) \leq \Delta(\mathcal{D}, \mathcal{D}_{i-1}) + \Delta(\mathcal{D}_{i-1}, \mathcal{D}_i) \leq 2\epsilon(i-1) + 2\epsilon = 2\epsilon i$, which concludes the proof of Lemma 3. □

References

1. Bruce, V.; Young, A. Understanding face recognition. *Br. J. Psychol.* **1986**, *77*, 305–327. [CrossRef] [PubMed]
2. Daugman, J. How iris recognition works. *IEEE Trans. Circuits Syst. Video Technol.* **2004**, *14*, 21–30. [CrossRef]
3. Ding, Y.; Zhuang, D.; Wang, K. A study of hand vein recognition method. In Proceedings of the IEEE International Conference Mechatronics and Automation, 2005; Volume 4, pp. 2106–2110. [CrossRef]
4. Jain, A.; Ross, A.; Prabhakar, S. An Introduction to Biometric Recognition. *IEEE Trans. Circuits Syst. Video Technol.* **2004**, *14* 4–20. [CrossRef]
5. Maltoni, D.; Maio, D.; Jain, A.K.; Prabhakar, S. *Handbook of Fingerprint Recognition*; Springer: London, UK, 2009. ISBN:978-1-84882-254-2.
6. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Proceedings of the Advances in Cryptology (EUROCRYPT 2004), Interlaken, Switzerland, 2–6 May 2004; pp. 523–540.[CrossRef]
7. Takahashi, K.; Matsuda, T.; Murakami, T.; Hanaoka, G.; Nishigaki, M. A signature scheme with a fuzzy private key. In Proceedings of the 13th International Conference on Applied Cryptography and Network Security, New York, NY, USA, 2–5 June 2015; pp. 105–126. [CrossRef]
8. Boyen, X. Reusable Cryptographic Fuzzy Extractors. In Proceedings of the 11th ACM Conference on Computer and Communications Security, Washington, DC, USA, 25–29 October 2004; pp. 82–91. [CrossRef]
9. Daugman, J. Probing the Uniqueness and Randomness of IrisCodes: Results From 200 Billion Iris Pair Comparisons. *Proc. IEEE* **2006**, *94*, 1927–1935. [CrossRef]
10. Desoky, A.I.; Ali, H.A.; Abdel-Hamid, N.B. Enhancing iris recognition system performance using templates fusion. *Ain Shams Eng. J.* **2012**, *3*, 133–140. [CrossRef]
11. Hollingsworth, K.P.; Bowyer, K.W.; Flynn, P.J. Improved Iris Recognition through Fusion of Hamming Distance and Fragile Bit Distance. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2465–2476. [CrossRef] [PubMed]
12. Alamélou, Q.; Berthier, P.E.; Cachet, C.; Cauchie, S.; Fuller, B.; Gaborit, P.; Simhadri, S. Pseudoentropic Isometries: A New Framework for Fuzzy Extractor Reusability. In Proceedings of the 2018 on Asia Conference on Computer and Communications Security, Incheon, Korea, 4–8 June 2018; pp. 673–684. [CrossRef]
13. Wen, Y.; Liu, S. Reusable Fuzzy Extractor from LWE. In Proceedings of the Australasian Conference on Information Security and Privacy 2018, Wollongong, Australia , 11–13 July 2018; pp. 13–27.[CrossRef]
14. Wen, Y.; Liu, S.; Han, S. Reusable Fuzzy Extractor from the Decisional Diffie—Hellman Assumption. *Des. Codes Cryptogr.* **2018**, *86*, 2495–2512. [CrossRef]

15. Wen, Y.; Liu, S. Robustly Reusable Fuzzy Extractor from Standard Assumptions. In Proceedings of the Advances in Cryptology (ASIACRYPT 20180, Brisbane, Australia, 2–6 December 2018; pp. 459–489. [CrossRef]
16. Apon, D.; Cho, C.; Eldefrawy, K.; Katz, J. Efficient, Reusable Fuzzy Extractors from LWE. In Proceedings of the International Conference on Cyber Security Cryptography and Machine Learning, Be'er Sheva, Israel, 29–30 June 2017; pp. 1–18. [CrossRef]
17. Canetti, R.; Fuller, B.; Paneth, O.; Reyzin, L.; Smith, A. Reusable fuzzy extractors for low-entropy distributions. In Proceedings of the Advances in Cryptology (EUROCRYPT 2016), Vienna, Austria, 8–12 May 2016; pp. 117–146. [CrossRef]
18. Cheon, J.; Jeong, J.; Kim, D.; Lee, J. A Reusable Fuzzy Extractor with Practical Storage Size: Modifying Canetti et al.'s Construction. In Proceedings of the Australasian Conference on Information Security and Privacy 2018, Wollongong, Australia, 11–13 July 2018; pp. 28–44. [CrossRef]
19. Tian, Y.; Li, Y.; Deng, R.H.; Sengupta, B.; Yang, G. Lattice-Based Remote User Authentication from Reusable Fuzzy Signature. Cryptology ePrint Archive: Report 2019/743. Available online: https://eprint.iacr.org/2019/743 (accessed on 24 June 2019).
20. Seo, M.; Hwang, J.Y.; Lee, D.H.; Kim, S.; Kim, S.; Park, J.H. Fuzzy Vector Signature and Its Application to Privacy-Preserving Authentication. *IEEE Access* **2019**, *7* 69892–69906. [CrossRef]
21. Boneh, D.; Raghunathan, A.; Segev, G. Function-Private Identity-Based Encryption: Hiding the Function in Functional Encryption. In Proceedings of the Advances in Cryptology (CRYPTO 2013), Santa Barbara, CA, USA, 18–22 August 2013; pp. 461–478. [CrossRef]
22. Regev, O. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing (STOC 2005), Baltimore, MA, USA, 22–24 May 2005; pp. 84–93. [CrossRef]
23. Fuller, B.; Meng, X.; Reyzin, L. Computational Fuzzy Extractors. In Proceedings of the advances in Cryptology (ASIACRYPT 2013), Bangalore, India, 1–5 December 2013; pp. 174–193. [CrossRef]
24. Matsuda, T.; Takahashi, K.; Murakami, T.; Hanaoka, G. Fuzzy Signatures: Relaxing Requirements and a New Construction. In Proceedings of the Applied Cryptography and Network Security (ACNS 2016), London, UK, 19–22 June 2016; pp. 97–116. [CrossRef]
25. Yasuda, M.; Shimoyama, T.; Takenaka, M.; Abe, N.; Yamada, S.; Yamaguchi, J. Recovering Attacks Against Linear Sketch in Fuzzy Signature Schemes of ACNS 2015 and 2016. In Proceedings of the Information Security Practice and Experience, Melbourne, Australia, 13–15 December 2017; pp. 409–421. [CrossRef]
26. Takahashi, K.; Matsuda, T.; Murakami, T.; Hanaoka, G.; Nishigaki, M. Signature schemes with a fuzzy private key. In *Int. J. Inf. Secur.* **2019**, *18*, 581–617. [CrossRef]
27. Vadhan, S.P. Pseudorandomness. *Found. Trends Theor. Comput. Sci.* **2012**, *7* 1–336. [CrossRef]
28. Bernhard, D.; Pereira, O.; Warinschi, B. How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios. In Proceedings of the Advances in Cryptology (ASIACRYPT 2012), Beijing, China, 2–6 December 2012; pp. 626–643. [CrossRef]
29. Kiltz, E.; Masny, D.; Pan, J. Optimal security proofs for signatures from identification schemes. In Proceedings of the Advances in Cryptology (CRYPTO 2016), Santa Barbara, CA, USA, 14–18 August 2016; pp. 33–61. [CrossRef]
30. Boneh, D.; Boyen, X. Short Signatures Without Random Oracles. In Proceedings of the Advances in Cryptology (EUROCRYPT 2004), Interlaken, Switzerland, 2–6 May 2004; pp. 56–73. [CrossRef]
31. Boneh, D.; Shen, E.; Waters, B. Strongly Unforgeable Signatures Based on Computational Diffie-Hellman. In Proceedings of the Public Key Cryptography (PKC 2006), New York, NY, USA, 24–26 April 2006; pp. 229–240. [CrossRef]
32. Kurihara, J.; Kiyomoto, S.; Fukushima, K.; Tanaka, T. A New (k,n)-Threshold Secret Sharing Scheme and Its Extension. In Proceedings of the Information Security 2008, Taipei, Taiwan, 15–18 September 2008; pp. 455–470. [CrossRef]
33. ISO/IEC 15946-5:2017. *Information Technology-Security Techniques-Cryptographic Techniques Based on Elliptic Curves—Part 5: Elliptic Curve Generation*; International Organization for Standardization: Geneva, Switzerland , 2017.
34. Bos, J.W.; Costello, C.; Naehrig, M. Exponentiating in Pairing Groups. In Proceedings of the Selected Areas in Cryptography (SAC 2013), Coimbra, Portugal, 18–22 March 2013; pp. 438–455. [CrossRef]

35. HÅstad, J.; Impagliazzo, R.; Levin, L.A.; Luby, M. A Pseudorandom Generator from Any One-Way Function. *SIAM J. Comput.* **1999**, *28*, 1364–1396. [CrossRef]