(Re-)Formalization and Construction of Reusable and Robust Threshold Fuzzy Extractors

Keisuke Hara[1,2], Keitaro Hashimoto[2], Takahiro Matsuda[2], Wataru Nakamura[3], and Kenta Takahashi[3]

[1] The University of Osaka, Japan
hara-keisuke@ist.osaka-u.ac.jp
[2] National Institute of Advanced Industrial Science and Technology (AIST), Japan
{keitaro.hashimoto, t-matsuda}@aist.go.jp
[3] Hitachi, Ltd., Japan
{wataru.nakamura.va, kenta.takahashi.bw}@hitachi.com

Abstract. A fuzzy extractor is a cryptographic primitive that enables us to generate uniform randomness from fuzzy data w with high entropy and reproduce the same randomness from fuzzy data w' close to w. Conventional fuzzy extractors support only a single input, making them unsuitable for scenarios that require multi-factor authentication, distributed key management, or avoiding single points of failure. In this paper, we (re)formalize and construct k-out-of-n *threshold fuzzy extractors* which enable us to generate randomness from multiple fuzzy data $(w_1, \ldots, w_n)$ and recover the same string from $k \leq n$ fuzzy data $(w'_{i_1}, \ldots, w'_{i_k})$ such that each of w'_{i_j} is close to w_{i_j} $(1 \leq j \leq k)$. We formalize the syntax and security notions of threshold fuzzy extractors as a natural extension of ordinary fuzzy extractors. For security, we consider information-theoretic security and computational reusability and robustness as in ordinary fuzzy extractors. Then, we propose two generic constructions of threshold fuzzy extractors. The first one satisfies information-theoretic security, and the second one satisfies reusability and robustness.

Keywords: Fuzzy extractors, Threshold cryptography, Reusability, Robustness

1 Introduction

1.1 Background

Cryptographic primitives require uniformly random and reproducible keys to ensure their functionality and security. Yet, many real-world random sources (e.g., biometrics and physically unclonable functions (PUFs) on physical devices) are noisy; that is, their measurements differ due to small errors. Thus, they are only approximately reproducible. To reconcile this gap, fuzzy extractors

(FEs) [21, 22, 25, 26, 43, 44] have been widely studied: given fuzzy data $w \in \mathcal{M}$ drawn from a distribution W over a fuzzy data space $\mathcal{M}$, the generation algorithm Gen generates a string R and outputs a helper data P which allows error correction; the reproducing algorithm Rep correctly reproduces the same R from the helper data P and any $w' \in \mathcal{M}$ which is close to w. The minimal security requirement for FE is that the generated randomness R is indistinguishable from a uniformly random value, even knowing P, if W has a sufficiently large entropy and w is used only once for Gen. Beyond basic correctness and security, two strengthened properties called *reusability* [3, 6, 10, 46, 49] and *robustness* [7,14,20,30] have emerged as central to deal with the real-world usage of FE. Informally, reusability ensures that the same fuzzy data can be safely reused multiple times. Robustness prevents tampering with the helper data P, and such tampering can be detected in the reproduction phase. Recently, FE schemes that achieve both properties simultaneously have been proposed [47, 48]. To date, FE has become foundational for key derivations from fuzzy data and as components inside higher-level protocols [5, 19, 23, 24, 41].

While FE is already an attractive primitive for generating cryptographic keys from fuzzy data, it supports only a *single* input. This is a significant drawback when real-world usage of FEs is considered. In reality, systems supporting biometrics usually allow us to register multiple fuzzy data to improve security. For example, recent smartphones with fingerprint authentication support registering multiple fingerprints. Such a threshold setting is particularly attractive in cryptography for reducing the risk of a single point of failure and the risk of secret key compromise [1, 2, 9, 11, 15–17, 27, 32, 40, 50]. For example, a threshold signature scheme is used in Zcash [13, 15, 32, 51] to authenticate transactions. Since signatures are generated by using a part of the distributed signing key shares, even if some shares are compromised, the signature scheme remains unforgeable. Furthermore, threshold signatures offer reliability: transactions can still be signed even if part of the shares are lost, reducing the risk of losing assets. In response to the growing demand for threshold cryptography in the real world, NIST decided to standardize threshold cryptosystems [8].

To use FEs in threshold settings for multiple-fuzzy-data scenarios, Ma, Qi, and Lv [36] proposed *threshold fuzzy extractors* (TFEs) that enable us to generate and reproduce randomness from multiple fuzzy data. By using TFEs, we can upgrade systems that use single-input FEs into threshold systems that support multiple fuzzy data. For example, (single-input) FEs can be used to generate a signing key of signature schemes based on single biometric data. By replacing FEs with TFEs, a signing key can be generated from multiple biometric data, such as finger-veins and irises, in a threshold manner. More specifically, to generate a signing key, a user generates randomness from finger-veins and irises by using a TFE scheme, and then generates a signing key based on the randomness; once the user wants to sign a message, he/she recovers the same randomness from these biometrics and generates the signing key from it. This realizes a kind of threshold signature scheme that offers the same security and reliability properties as ordinary (2-out-of-2) threshold signature schemes. Other

cryptographic schemes that generate keys using FEs, or authentication protocols based on FEs, can also be thresholdized in the same manner as the above signature scheme example.

While Ma et al. initiated research on TFEs, as we will discuss in Section 3, their formalization and construction of TFEs have several issues. Especially, the syntax of their TFE schemes is redundant compared with that of ordinary single-input FE schemes. Because of this, a TFE scheme with their syntax cannot be used as a direct replacement for an FE scheme used inside existing secure systems. Besides, their security notions are problematic. Their definitions have some informal treatments and ambiguities, allowing room for multiple possible interpretations (which a formal definition of security must avoid). Notably, their definition of reusability fails to accurately capture real-world attack scenarios. Finally, their proposed TFE scheme is not provably secure because the underlying assumption is insufficient. (We will detail the issues in Section 3.) In this paper, we reconsider TFEs and aim to provide an appropriate formalization and provably secure constructions.

1.2 Our Contributions

In this paper, we reformalize threshold fuzzy extractors (TFEs) and provide generic constructions.

First, we redefine the syntax and security notions of TFE schemes. We consider a k-out-of-n setting, where randomness is generated from n fuzzy data and can be reproduced by using any k (out of n) "close" fuzzy data. As in ordinary FE schemes, we define two algorithms Gen and Rep for TFE: Gen takes as input n fuzzy data $(w_1, \cdots, w_n) \in \mathcal{M}^n$, which is sampled from some joint distribution $W = (W_1, \ldots, W_n)$, and outputs a string R and a helper data P. Rep takes k fuzzy data $(w'_{i_1}, \cdots, w'_{i_k})$ and a helper data P as input, and outputs a string R'.

The correctness requires that, for any set of fuzzy data $\{ w'_{i_j} \}_{j\in[k]}$[4], if each of w_{i_j} and w'_{i_j} is close enough, then $R = R'$ should hold. For security notions, we define information-theoretic security and computational reusability and robustness for TFE by naturally extending them for FEs [22,48]. Since we consider the threshold setting, an adversary against TFE schemes should be allowed to know $k-1$ fractions of n fuzzy data of its choice. Furthermore, we assume that each of w_i sampled from the joint distribution $W = (W_1, \ldots, W_n)$ has sufficiently large average conditional min-entropy $\tilde{\mathbf{H}}_\infty(W_i | \{ W_j \}_{j\in[n]\setminus\{i\}})$ for all $i \in [n]$. In this setting, information-theoretic security guarantees that R is statistically indistinguishable from a truly random value if the set of fuzzy data $\{ w_i \}_{i\in[n]}$ is used by Gen only once; Reusability ensures that, even when the same set of fuzzy data $\{ w_i \}_{i\in[n]}$ is used multiple times with different errors to generate $R_1, \ldots, R_m$, these values are computationally indistinguishable from truly random values $U_1, \ldots, U_m$; Robustness ensures that no computational adversary can forge a valid helper data even if the adversary sees multiple honestly-generated helper data.

[4] For an integer k, $[k]$ denotes the set $\{ 1, \ldots, k \}$.

Then, we propose two generic constructions of TFE schemes. First, we provide an information-theoretically secure TFE scheme based on an average-case secure sketch ($\mathsf{SS.Gen}, \mathsf{SS.Rec}$), a universal hash function H, a one-time pad encryption scheme OTP, and a secret sharing scheme. The construction idea is to split the random string R into k-out-of-n shares and encrypt each share with each fuzzy data. More specifically, $\mathsf{Gen}(w_1, \ldots, w_n)$ samples randomness $R \coloneqq m$ and generates its n secret shares $\{ m_i \}_{i \in [n]}$ with a k-out-of-n secret sharing scheme. Then, for each fuzzy data w_i, its sketch $s_i \coloneqq \mathsf{SS.Gen}(w_i)$ and secret key $\mathsf{sk}_i \coloneqq \mathsf{H}(w_i)$ are computed, respectively. Then, m_i is encrypted as $c_i \coloneqq \mathsf{OTP.Enc}(\mathsf{sk}_i, m_i)$. The final helper data is $P = \{ (s_i, c_i) \}_{i \in [n]}$. $\mathsf{Rep}(P, w_{i_1}, \ldots, w_{i_k})$ recomputes the secret key $\mathsf{sk}_{i_j} \coloneqq \mathsf{H}(w'_{i_j})$ from $w'_{i_j} \coloneqq \mathsf{SS.Gen}(s_{i_j}, w_{i_j})$ and decrypts the share $m_{i_j} \coloneqq \mathsf{OTP.Dec}(\mathsf{sk}_{i_j}, c_{i_j})$ for $j \in [k]$. Then, it reconstructs $R = m$ from k shares $\{ m_{i_j} \}_{j \in [k]}$. We prove that it meets our information-theoretic security notion due to the statistical security of the average-case secure sketch scheme, the perfect security of the secret sharing scheme, and the one-time pad encryption. More specifically, first, by using the (generalized) leftover hash lemma, we can prove that uncorrupted $n - k + 1$ secret keys sk_i are indistinguishable from truly random keys due to the assumption of min-entropy on the set of fuzzy data and the security of the average-case secure sketch scheme. Then, the security of the one-time pad encryption guarantees the confidentiality of $n - k + 1$ secret shares of the randomness R. As a result, no (computationally unbounded) adversary can obtain more than $k - 1$ shares. Thus, due to the security of the secret sharing scheme, it has no information about R.

Second, we construct a reusable and robust TFE scheme by modifying our information-theoretically secure TFE scheme. To achieve reusability, we need to tolerate adversarially-chosen errors in randomness generation. For this purpose, we observe that a homomorphism for the secure sketch scheme and a universal hash function, together with a related-key-secure symmetric-key encryption scheme, are needed. Roughly speaking, these properties guarantee the confidentiality of the secret-shared R in the $n - k + 1$ ciphertexts of uncorrupted keys, even if an adversary knows errors in the fuzzy data (and thus errors in the corresponding secret keys for encryption). To demonstrate robustness, we classify attacks against it and observe that an adversary must either modify an honestly-generated helper data or forge a new helper data for unknown fuzzy data (recall that Rep uses k fuzzy data, but the adversary is allowed to corrupt only $k - 1$) to break robustness. To prevent the former attack, we employ a one-time signature (OTS) scheme. We bind the set $\{ (s_i, c_i) \}_{i \in [n]}$ by generating an OTS signature for it as $\sigma \leftarrow \mathsf{OTS.Sign}(\mathsf{osk}, \{ (s_i, c_i) \}_{i \in [n]})$ and link the corresponding verification key ovk with the ciphertexts c_i. Since the adversary does not know the signing key osk, it cannot create new signatures for an honestly-generated helper data. To prevent the latter attack, we employ a related-key secure auxiliary-input authenticated encryption (AIAE) scheme [28,48], and we set the OTS's verification key ovk as an auxiliary input of it. Roughly, AIAE's integrity property guarantees that any adversary who does not know the secret key cannot forge a valid AIAE ciphertext. Since the adversary knows only $k - 1$

fuzzy data but Rep uses k data, the adversary must forge a valid ciphertext for unknown fuzzy data, which should be infeasible due to the integrity of AIAE.[5]

Finally, we provide concrete instantiations of our TFE schemes. Based on the known average-case secure sketch scheme and the universal hash family explained in [21, 22] and Shamir's secret sharing scheme [39], we obtain an information-theoretically secure TFE scheme with randomness space $\mathbb{Z}_p$. A reusable and robust TFE scheme in the standard model can be obtained from the average-case secure sketch scheme, the universal hash family, and the AIAE scheme based on the decisional Diffie-Hellman (DDH) assumption explained in [48], Shamir's secret sharing scheme [39], and the discrete-logarithm-based one-time signature scheme [4]. In the random oracle model, we can realize an AIAE scheme from any block cipher such as AES. (We discuss this in the full version of this paper.) We note that the size of P in our TFE schemes is linear in n; this is not a problem in use cases where n is small.

1.3 Applications and Use Cases of TFE

There are several applications of TFE, which help with distributed key management and improve fault tolerance. One use case of TFE is in the cryptocurrency ecosystem. TFE can be used to decentralize the authority to approve fund transfers on cryptocurrency exchanges. Some cryptocurrency exchanges employ the so-called Multi-Sig [12], where cryptocurrency is transferred to a recipient when both a sender and an exchange sign a transaction with a digital signature scheme. By using TFE at the exchange, the signing key can be delegated to specific individuals, distributing authority to authorize fund transfers among multiple administrators. Let us consider a scenario in which the signing key is distributed among n people working at the exchange, and transactions are signed when t of them gather. In this case, the n authorized individuals gather at the exchange's office, scan their biometric data using computer scanners, and execute TFE.Gen to generate the random string used for signature key generation. Next, when approving a fund transfer transaction, t people gather at the office and use TFE.Rep to recover the randomness, thereby obtaining the signature key. In this way, by utilizing TFE, the authority to approve transactions can be tied to individuals while being managed in a decentralized manner. Furthermore, reusable TFE enables us to securely generate multiple cryptographic keys from the same set of n biometric data.

Another use case of TFE is verifying product authenticity using artifact-metrics. Artifact-metrics [29,34,37] is unique physical characteristics of artifacts, and they can be used for product identification, very much like biometrics is used to identify persons. TFE enables us to create a product certificate based on a key generated from multiple artifact-metrics (e.g., optical and electrical properties). Using TFE in the n-out-of-n setting makes it more difficult to counterfeit the product. When we use TFE in the k-out-of-n setting, even if some metrics cannot

[5] In the formal proof for robustness, we also employ the linearity of the secure sketch scheme.

be obtained due to product damage, the product can still be authenticated using the remaining metrics, thereby enhancing fault tolerance. In this use case, a helper data is attached to the certificate to enable the recipient to recover the secret key from the received product. Robustness of TFE prevents an adversary from tampering with the helper data during product delivery to customers.

1.4 Paper Organization

In Section 2, we review the basic notation and the definitions for cryptographic primitives treated in this paper. In Section 3, we discuss the issues in Ma et al.'s formalization and construction of TFE. In Section 4, we provide our formalization of TFE and security definitions: information-theoretic security, (computational) reusability, and robustness. In Section 5, we give our generic construction of information-theoretically secure TFE. In Section 6, we give our generic construction of reusable and robust TFE. Section 7 explains concrete instantiations of TFE obtained from our generic constructions, and shows their efficiency.

2 Preliminaries

In this section, we review some basic notations and cryptographic primitives.

2.1 Basic Notation

For $n, m \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$ and $[n \setminus m] := \{1, \ldots, m-1, m+1, \ldots, n\} = [n] \setminus \{m\}$. If $\mathcal{X}$ is a finite set, then $x \xleftarrow{\$} \mathcal{X}$ denotes that $x \in \mathcal{X}$ is sampled uniformly at random. If D is a distribution over some set, then $x \xleftarrow{\$} D$ denotes that x is sampled according to D. For two values z and z', $[\![z = z']\!]$ denotes the operation that returns 1 if and only if $z = z'$ holds. ($[\![z \neq z']\!]$ is defined analogously.) A function $f : \mathbb{N} \to [0, 1]$ is *negligible* if $f(\lambda) < \lambda^{-c}$ holds for all $c \in \mathbb{N}$ and all sufficiently large $\lambda \in \mathbb{N}$. $\mathsf{negl}(\lambda)$ denotes an unspecified negligible function. "PPT" stands for *probabilistic polynomial time*. For two random variables X and Y, $\mathbf{H}_\infty(X) := -\log(\max_x \Pr[X = x])$ denotes the min-entropy of X, $\tilde{\mathbf{H}}_\infty(X|Y) := -\log(\mathbb{E}_{y \xleftarrow{\$} Y}[\max_x \Pr[X = x|Y = y]]) = -\log(\mathbb{E}_{y \leftarrow Y}[2^{-\mathbf{H}_\infty(X|Y=y)}])$ denotes the average conditional min-entropy of X given Y, and $\mathsf{SD}(X, Y) := \frac{1}{2}\sum_x |\Pr[X = x] - \Pr[Y = x]|$ denotes the statistical distance between X and Y.

We will use a one-time pad encryption scheme $\mathsf{OTP} = (\mathsf{OTP.Enc}, \mathsf{OTP.Dec})$, which is a perfectly secure deterministic symmetric-key encryption scheme. We will provide its formal definition in the full version of this paper.

2.2 Universal Hash Function Family

In this section, we recall the definitions of a universal hash function family and the generalized leftover hash lemma.

Definition 1 (Universal Hash Function Family). *Let $\mathcal{X}$ and $\mathcal{Y}$ be sets. A family of functions $\mathcal{H} \coloneqq \{\mathsf{H} : \mathcal{X} \to \mathcal{Y}\}$ is universal if for any $x_1, x_2 \in \mathcal{X}$ such that $x_1 \neq x_2$, we have*

$$\Pr_{\mathsf{H} \xleftarrow{\$} \mathcal{H}}[\mathsf{H}(x_1) = \mathsf{H}(x_2)] \leq \frac{1}{|\mathcal{Y}|}.$$

Definition 2 (Homomorphic Universal Hash Functions). *We say that a family of universal hash functions $\mathcal{H}$ is homomorphic, if for all $\mathsf{H} \in \mathcal{H}$, it holds that*

$$\mathsf{H}(x + x') = \mathsf{H}(x) + \mathsf{H}(x').$$

We recall the generalized leftover hash lemma [21] in the form we use later.

Lemma 1 (Generalized Leftover Hash Lemma [21]). *Let $\mathcal{H} = \{\mathsf{H} : \mathcal{X} \to \mathcal{Y}\}$ be a universal hash family. Let us consider the following game (which we call the GLHL game) between the challenger and an adversary $\mathcal{A}$:*

1. *$\mathcal{A}$ sends the description of a joint distribution (X, Z) such that X is a distribution over $\mathcal{X}$ and Z is a distribution over a set $\mathcal{Z}$ to the challenger.*
2. *The challenger chooses a bit $b \xleftarrow{\$} \{0,1\}$ and samples $\mathsf{H} \xleftarrow{\$} \mathcal{H}$, $(x, z) \xleftarrow{\$} (X, Z)$, and $y \xleftarrow{\$} \mathcal{Y}$. If $b = 1$, the challenger returns $(\mathsf{H}, \mathsf{H}(x), z)$ to $\mathcal{A}$; Otherwise, the challenger returns (H, y, z) to $\mathcal{A}$.*
3. *$\mathcal{A}$ outputs $b' \in \{0,1\}$.*

Then, for any computationally unbounded adversary $\mathcal{A}$, it holds that

$$\mathsf{Adv}^{\mathsf{LHL}}_{\mathcal{H},\mathcal{A}}(\lambda) \coloneqq |\Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]| \leq \frac{1}{2}\sqrt{|\mathcal{Y}| \cdot 2^{-\tilde{\mathbf{H}}_\infty(X|Z)}}.$$

2.3 Secret Sharing

In this section, we review the definition of k-out-of-n secret sharing.

Definition 3 (Secret Sharing Scheme). *A k-out-of-n secret sharing scheme* SecretShare *over $\mathcal{M}$ consists of the following two algorithms:*

$\mathsf{Share}(k, n, s)$**:** *The share generation algorithm takes two integers k and n and a secret $s \in \mathcal{M}$ as input, and outputs a set of n shares $(s_i)_{i \in [n]}$.*

$\mathsf{Reconst}(J, (s_i)_{i \in J})$**:** *The (deterministic) reconstruction algorithm takes a set $J \subseteq [n]$ and shares $(s_i)_{i \in J}$ as input, and outputs a secret $s \in \mathcal{M}$.*

We require that a k-out-of-n secret sharing scheme SecretShare *satisfy the following properties.*

Correctness: *We say that* SecretShare *satisfies correctness, if for any $k, n \in \mathbb{N}$ such that $0 < k \leq n$, any $s \in \mathcal{M}$, any possible shares $(s_i)_{i \in [n]} \leftarrow \mathsf{Share}(k, n, s)$ and any subset $J \subseteq [n]$ with $k \leq |J|$, we have $\mathsf{Reconst}(J, (s_i)_{i \in J}) = s$.*

Security: *Let us consider the following game between the challenger and an adversary $\mathcal{A}$:*

1. *$\mathcal{A}$ sends a set $J \subset [n]$ such that $|J| \leq k-1$ and two secrets $s^0, s^1 \in \mathcal{M}$ to the challenger.*
2. *The challenger chooses a bit $b \xleftarrow{\$} \{0,1\}$ and generates $(s_i)_{i\in[n]} \leftarrow \mathsf{Share}(k, n, s^b)$. Then, it returns $(s_i)_{i\in J}$ to $\mathcal{A}$.*
3. *$\mathcal{A}$ outputs $b' \in \{0,1\}$.*

We say that SecretShare *is secure, if for any computationally unbounded adversary $\mathcal{A}$, it holds that*

$$\Pr[b'=1|b=1] = \Pr[b'=1|b=0].$$

2.4 One-Time Signature

In this section, we recall the definition of one-time signature (OTS) schemes.

Definition 4 (One-Time Signature). *A one-time signature (OTS) scheme* OTS *consists of the following three algorithms:*

$\mathsf{OTS.Gen}(1^\lambda)$**:** *The key generation algorithm takes a security parameter 1^λ as input, and outputs a pair of verification and signing keys $(\mathsf{ovk}, \mathsf{osk})$.*

$\mathsf{OTS.Sign}(\mathsf{osk}, m)$**:** *The signature generation algorithm takes a signing key* osk *and a message m as input, and outputs a signature σ.*

$\mathsf{OTS.Ver}(\mathsf{ovk}, m, \sigma)$**:** *The (deterministic) verification algorithm takes a verification key* ovk*, a message m, and a signature σ as input, and outputs 1 or 0.*

We require that an OTS scheme OTS *satisfy the following properties.*

Correctness: *We say that* OTS *satisfies correctness, if for any $\lambda \in \mathbb{N}$, any key $(\mathsf{ovk}, \mathsf{osk}) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$, and any message m, it holds that* $\mathsf{OTS.Ver}(\mathsf{ovk}, m, \mathsf{OTS.Sign}(\mathsf{osk}, m)) = 1$.

Security: *Consider the following game between the challenger and an adversary $\mathcal{A}$:*

1. *The challenger generates $(\mathsf{ovk}^*, \mathsf{osk}^*) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$ and sends ovk^* to $\mathcal{A}$.*
2. *$\mathcal{A}$ sends a message m to the challenger. Then, the challenger generates $\sigma \leftarrow \mathsf{OTS.Sign}(\mathsf{osk}^*, m)$ and returns σ to $\mathcal{A}$.*
3. *$\mathcal{A}$ outputs (m^*, σ^*). If $(m,\sigma) \neq (m,\sigma^*) \wedge \mathsf{OTS.Ver}(\mathsf{ovk}^*, m^*, \sigma^*) = 1$ holds, the challenger sets $b := 1$. Otherwise, the challenger sets $b := 0$.*

We say that OTS *is* sEUF-CMA *secure, if for any PPT adversary $\mathcal{A}$, it holds that $\mathsf{Adv}^{\mathsf{unf}}_{\mathsf{OTS},\mathcal{A}}(\lambda) := \Pr[b=1] = \mathsf{negl}(\lambda)$.*

2.5 Secure Sketch

In this section, we review the definition of an average-case secure sketch (SS) scheme with linearity and the homomorphic property.

Definition 5 (Average-Case Secure Sketch [21, Definition 4]). *A secure sketch (SS) scheme* SS *is defined by the following (deterministic) algorithms:*

$\mathsf{SS.Gen}(w)$**:** *The sketch generation algorithm takes fuzzy data* w *as input, and outputs a sketch* s*.*

$\mathsf{SS.Rec}(w', s)$**:** *The reconstruction algorithm takes fuzzy data* w' *and a sketch* s *as input, and outputs* $\tilde{w}$*.*

Let $\mathcal{M}$ *be a metric space with a distance function* dis*, to which fuzzy data belongs, and* $\mathcal{S}$ *be a space of sketches* s*. We say that* SS *is an* $(\mathcal{M}, \ell, \tilde{\ell}, d)$*-average-case secure sketch scheme (or simply, an* $(\mathcal{M}, \ell, \tilde{\ell}, d)$*-secure sketch scheme), if it satisfies the following properties:*

Correctness: *For any* $w, w' \in \mathcal{M}$ *such that* $\mathsf{dis}(w, w') \leq d$*, it holds that* $\mathsf{SS.Rec}(w', \mathsf{SS.Gen}(w)) = w$*.*

Privacy: *For any random variable* W *over* $\mathcal{M}$ *and* I *over* $\{0,1\}^*$ *such that* $\tilde{\mathbf{H}}_\infty(W|I) \geq \ell$*, it holds that* $\tilde{\mathbf{H}}_\infty(W|\mathsf{SS.Gen}(W), I) \geq \tilde{\ell}$*.*

Definition 6 (Linearity). *Let* $\mathsf{SS} = (\mathsf{SS.Gen}, \mathsf{SS.Rec})$ *be an* $(\mathcal{M}, \ell, \tilde{\ell}, d)$*-secure sketch scheme. We say that* SS *satisfies linearity if there exists an efficiently computable (deterministic) function* g *with the following property: Let* $w \in \mathcal{M}$*,* $\tilde{s} \in \mathcal{S}$*, and* δ *such that* $\mathsf{dis}(0, \delta) \leq d$*. Furthermore, let* $s := \mathsf{SS.Gen}(w)$ *and* $\tilde{w} := \mathsf{SS.Rec}(w + \delta, \tilde{s})$*. Then, it holds that* $g(\delta, s, \tilde{s}) = \tilde{w} - w$*.*

Definition 7 (Homomorphism). *Let* $\mathsf{SS} = (\mathsf{SS.Gen}, \mathsf{SS.Rec})$ *be an* $(\mathcal{M}, \ell, \tilde{\ell}, d)$*-secure sketch scheme. We say that* SS *satisfies homomorphism, if it holds that* $\mathsf{SS.Gen}(w + w') = \mathsf{SS.Gen}(w) + \mathsf{SS.Gen}(w')$ *for all* $w, w' \in \mathcal{M}$*.*

2.6 Auxiliary-Input Authenticated Encryption

In this section, we recall the definition of auxiliary-input authenticated encryption (AIAE) introduced by Han et al. [28].

Definition 8 (Auxiliary-Input Authenticated Encryption). *An auxiliary-input authenticated encryption (AIAE) scheme* AIAE *consists of the following three algorithms:*

$\mathsf{AIAE.Setup}(1^\lambda)$**:** *The setup algorithm takes a security parameter* 1^λ *as input, and outputs a public parameter* pp*. The public parameter* pp *implicitly defines the key space* $\mathcal{K}$*, the plaintext space* $\mathcal{M}$*, and the auxiliary input space* $\mathcal{AUX}$*.*

$\mathsf{AIAE.Enc}(\mathsf{pp}, k, m, \mathsf{aux})$**:** *The encryption algorithm takes a public parameter* pp*, a key* k*, a plaintext* m*, and an auxiliary input* aux *as input, and outputs a ciphertext* c*.*

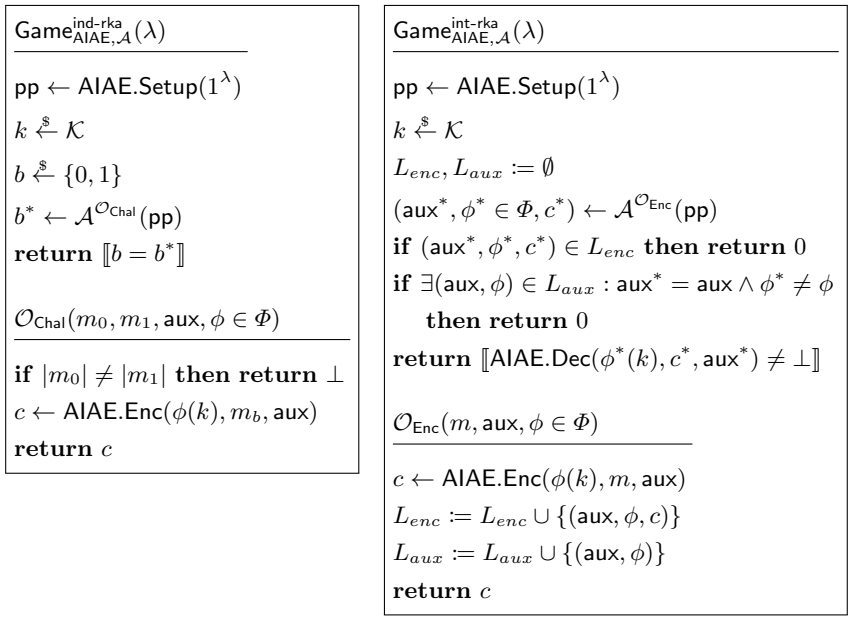

Fig. 1: Security game for IND-Φ-RKA security (left) and that for weak INT-Φ-RKA security (right) of an AIAE scheme AIAE.

$\mathsf{AIAE.Dec}(\mathsf{pp}, k, c, \mathsf{aux})$**:** *The (deterministic) decryption algorithm takes a public parameter* pp*, a key* k*, a ciphertext* c*, and an auxiliary input* aux *as input, and outputs a plaintext* $m \in \mathcal{M}$ *or an error symbol* $\bot \notin \mathcal{M}$*.*

We say that AIAE *satisfies correctness, if for any* $\lambda \in \mathbb{N}$, $\mathsf{pp} \leftarrow \mathsf{AIAE.Setup}(1^\lambda)$, $k \in \mathcal{K}$, $m \in \mathcal{M}$, *and* $\mathsf{aux} \in \mathcal{AUX}$*, it holds that* $\mathsf{AIAE.Dec}(\mathsf{pp}, k, \mathsf{AIAE.Enc}(\mathsf{pp}, k, m, \mathsf{aux}), \mathsf{aux}) = m$*.*

Definition 9 (IND-Φ-RKA Security for AIAE). *Let* $\mathsf{AIAE} = (\mathsf{AIAE.Setup}, \mathsf{AIAE.Enc}, \mathsf{AIAE.Dec})$ *be an AIAE scheme, and let* $\Phi = \{\phi : \mathcal{K} \to \mathcal{K}\}$ *be a class of related-key deriving functions. We say that* AIAE *satisfies* IND-Φ-RKA *security, if for any PPT adversary* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\mathsf{ind\text{-}rka}}_{\mathsf{AIAE},\mathcal{A}}(\lambda) := 2 \cdot \left| \Pr[\mathsf{Game}^{\mathsf{ind\text{-}rka}}_{\mathsf{AIAE},\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right| = \mathsf{negl}(\lambda),$$

where the game $\mathsf{Game}^{\mathsf{ind\text{-}rka}}_{\mathsf{AIAE},\mathcal{A}}(\lambda)$ *is defined as in Fig. 1 (left).*

Definition 10 (Weak INT-Φ-RKA Security for AIAE). *Let* $\mathsf{AIAE} = (\mathsf{AIAE.Setup}, \mathsf{AIAE.Enc}, \mathsf{AIAE.Dec})$ *be an AIAE scheme, and let* $\Phi = \{\phi : \mathcal{K} \to \mathcal{K}\}$ *be a class of related-key deriving functions. We say that* AIAE *satisfies weak* INT-Φ-RKA *security, if for any PPT adversary* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\mathsf{int\text{-}rka}}_{\mathsf{AIAE},\mathcal{A}}(\lambda) := \Pr[\mathsf{Game}^{\mathsf{int\text{-}rka}}_{\mathsf{AIAE},\mathcal{A}}(\lambda) = 1] = \mathsf{negl}(\lambda),$$

where the game $\mathsf{Game}^{\mathsf{int\text{-}rka}}_{\mathsf{AIAE},\mathcal{A}}(\lambda)$ *is defined as in Fig. 1 (right).*

Related-key secure AIAE in the standard model can be constructed from the DDH assumption [28]. Also, related-key secure AIAE exists in an idealized model without any number-theoretic assumption [35]. (We discuss in detail in the full version of this paper.)

3 Issues in Ma et al.'s Formalization and Construction

In this section, we discuss issues in the formalization and construction of TFE by Ma et al. [36].

3.1 Issues in Formalization

We analyze their syntax and security notion, and observe the following issues. (Please refer to [36, Section 4.1] for their actual syntax and security notions for TFE.)

Incompatible syntax with ordinary FE: The syntax of TFE by Ma et al. is different from what we expect from ordinary FEs. In contrast to ordinary FEs that consist of three algorithms Setup, Gen, and Rep, a TFE scheme with their syntax consists of the four algorithms Setup, $\mathsf{Register}$, Gen, and Rep with the following syntax:

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$: It generates a public parameters pp.
- $\mathsf{Register}(\mathsf{pp}, \{ w_i \}_{i\in[n]}, k) \to \{ iP_i \}_{i\in[n]}$: It generates a set of individual helper data $\{ iP_i \}_{i\in[n]}$ from n fuzzy data $\{ w_i \}_{i\in[n]}$ so that k-out-of-n users can generate/recover random strings.
- $\mathsf{Gen}(\mathsf{pp}, \{ w_i \}_{i\in S}, \{ iP_i \}_{i\in S}) \to (cP, cR)$: It generates a random string cR and a related helper data cP from the (sub)set of fuzzy data $\{ w_i \}_{i\in S}$ and their corresponding helper data $\{ iP_i \}_{i\in S}$ such that $S \subseteq [n]$ and $k \leq |S|$.
- $\mathsf{Rep}(\mathsf{pp}, cP, \{ w_i \}_{i\in S}, \{ iP_i \}_{i\in S}) \to cR$: It reproduces a random string cR from cP and the set of fuzzy data $\{ w_i \}_{i\in S}$ and their corresponding helper data iP_i such that $S \subseteq [n]$ and $k \leq |S|$.

We think that this TFE syntax is designed with their intended applications in mind. Thus, it is incompatible with ordinary FEs, and their TFE scheme cannot be used directly as a drop-in replacement of ordinary FE schemes.

Ambiguous definitions: There are some informal treatments and ambiguities in their security definitions. For example, the definition of *inconsistency detection* is as follows:

> ***Inconsistency detection.*** *Inconsistent inputs will always be detected, including:*
>
> - *inconsistent fingerprint, a member might use w in the* $\mathsf{Register}$ *phase, but input w' satisfying* $\mathsf{dis}(w, w') > t$ *in* Gen *and/or* Rep *phase.*

– *inconsistent iP and cP, a member might input wrong or modified helper data in the* Gen *and/or* Rep *phase.*

If the number of consistent messages is less than k, Gen *and* Rep *output $\perp$.*

The above explanation lacks a clear definition of adversaries (e.g., capabilities and goals), despite being a security notion. Reusability is formally defined by a security game between an adversary and the challenger, but it also has some ambiguity. To capture scenarios where fuzzy data is reused, an oracle query has been formalized that allows an adversary to execute Register multiple times with their chosen error values. However, it is not explicitly stated whether these queries can be executed adaptively or not. The presence or absence of adaptive queries affects security levels, and thus, the ambiguity regarding it is obviously problematic.

Insufficiency of the reusability definition: To capture the pseudorandomness of cR, they define reusability in a similar way to that for ordinary FEs. We observe that their reusability definition does not properly capture actual situations in the real world. Specifically, during the reusability game, an adversary should be allowed to register the same set of fuzzy data with different error values, capturing the case where the same data is reused/registered multiple times. However, in their definition, it is assumed that the n fuzzy data input to the Register algorithm have the same error value. This assumption is unrealistic because each fuzzy data has a distinct error value in general. Second, errors are not considered in the challenge phase. The challenge randomness cR is generated from the set of fuzzy data, which is identical to the set used to generate the individual helper data. This means that at best, cR can appear random only when it is generated without errors. However, in the real world, multiple random strings are generated under various error values. Therefore, this definition does not properly capture the real-world situation in which a TFE scheme is used, leading us to conclude that their reusability definition is insufficient.

3.2 Issues in Construction

We discuss here the TFE scheme proposed by Ma et al. [36, Section 4.2]. Their construction uses the following building blocks:

– A homomorphic average-case strong extractor Ext with seed space $\mathcal{K}$.
– A homomorphic secure sketch (SS.Gen, SS.Rec).
– A group generation algorithm $(p, g, \mathbb{G}) \leftarrow \mathcal{G}(1^\lambda)$ that outputs a description of a multiplicative cyclic group.
– The symmetric-key version of the ElGamal encryption scheme (ElGamal.Enc, ElGamal.Dec).
– Shamir's secret sharing scheme (Shamir.Share, Shamir.Reconst).
– A collision-resistant hash function H.

Using the above building blocks, Ma et al.'s TFE scheme is constructed as in Fig. 2. We observe that their TFE scheme has the following issues.

The scheme is not provably secure: Ma et al. claim that their TFE scheme satisfies reusability under some assumptions on the building blocks and the distribution of fuzzy data. Among them, the assumption on the hash function H is that it is collision-resistant. However, (aside from the ambiguity of their definition of reusability,) we point out that their security claim cannot be true. Specifically, reusability (at least its intuitive definition) necessarily implies that the common randomness cR looks random against an adversary who only knows the helper data iP_i and cP. However, for their TFE scheme to satisfy such a least security requirement, assuming H is a collision-resistant hash function is insufficient. For example, the identity function is perfectly collision-resistant, and thus it can be used as H. Then, the hash value is $h = \mathsf{H}(csk, X) = (csk, X)$. Since the adversary knows $cP = (X, h)$, it can compute $cR = X^{csk}$ without knowing the secret fuzzy data. We note that if the hash function H is a random oracle, this attack is prevented. However, we need an unambiguous security definition before we can start discussing and constructing provably secure schemes.

Misuse of the ElGamal encryption scheme: The message space of the ElGamal encryption scheme is $\mathbb{G}$, while their TFE scheme encrypts $csk \in \mathbb{Z}_p$ with it, which is clearly incompatible.

4 Reformalization of Threshold Fuzzy Extractor

In the previous section, we discussed the problems in Ma et al.'s treatment of threshold fuzzy extractors (TFEs). To solve the problems regarding formalization, in this work, we reformalize k-out-of-n TFEs as a natural extension of ordinary FEs. First, we define TFE as being composed of the randomness generation algorithm Gen and the reproduction algorithm Rep (together with the setup algorithm Setup), similarly to FE. These algorithms work in the same manner as those in FE, except that Gen takes as input a set of n fuzzy data $\{ w_i \}_{i\in[n]}$ sampled from a joint distribution $W = (W_1, \ldots, W_n)$, and Rep takes k fuzzy data $\{ w_{i_j} \}_{j\in[k]}$. Correctness guarantees that if w_{i_j} and w'_{i_j} are close for each index $j \in [k]$, then the same R can be reproduced with the corresponding P. The formal syntax and correctness are as follows.

Definition 11 (Threshold Fuzzy Extractor). *Let $0 < k \leq n$, and let $\mathcal{M}$ be a metric space with a distance function dis. A k-out-of-n TFE scheme TFE consists of the following three algorithms:*

$\mathsf{TFE.Setup}(1^\lambda)$**:** *The setup algorithm takes a security parameter 1^λ as input, and outputs a common public parameter pp. The public parameter pp implicitly defines the randomness space $\mathcal{R}$.*

$\mathsf{TFE.Gen}(\mathsf{pp}, k, n, (w_1, \ldots, w_n))$**:** *The generation algorithm takes pp, two integers k and n, and a set of n fuzzy data $(w_1, \ldots, w_n) \in \mathcal{M}^n$ as input, and outputs a helper data P and a string $R \in \mathcal{R}$.*

$\mathsf{Setup}(1^\lambda)$

Selects a hash function H
$seed \xleftarrow{\$} \mathcal{K}$
$(p, g, \mathbb{G}) \leftarrow \mathcal{G}(1^\lambda)$
return $\mathsf{pp} := (seed, p, g, \mathbb{G}, \mathsf{H})$

$\mathsf{Gen}(\mathsf{pp}, \{ w_i \}_{i \in S}, \{ iP_i \}_{i \in S})$

if $|S| < k$ **then return** $\perp$
foreach $i \in S$ **do**
  parse $(s_i, ct_i, h_i) := iP_i$
  $w'_i := \mathsf{SS.Rec}(s_i, w_i)$
  $sk'_i := \mathsf{Ext}(seed, w'_i)$
  if $h_i \neq \mathsf{H}(sk'_i, s_i, ct_i)$ **then return** $\perp$
  $csk_i := \mathsf{ElGamal.Dec}(sk'_i, ct_i)$
$csk := \mathsf{Shamir.Reconst}(S, (csk_i)_{i \in S})$
$x \xleftarrow{\$} \mathbb{Z}_p; X := g^x$
$h := \mathsf{H}(csk, X)$
return $(cP, cR) := ((X, h), X^{csk})$

$\mathsf{Register}(\mathsf{pp}, \{ w_i \}_{i \in [n]}, k)$

$csk \xleftarrow{\$} \mathbb{Z}_p$
$(csk_i)_{i \in [n]} \leftarrow \mathsf{Shamir.Share}(k, n, csk)$
foreach $i \in [n]$ **do**
  $s_i := \mathsf{SS.Gen}(w_i)$
  $sk_i := \mathsf{Ext}(seed, w_i)$
  $ct_i \leftarrow \mathsf{ElGamal.Enc}(sk_i, csk_i)$
  $h_i := \mathsf{H}(sk_i, s_i, ct_i)$
return $\{ iP_i = (s_i, ct_i, h_i) \}_{i \in [n]}$

$\mathsf{Rep}(\mathsf{pp}, \{ w_i \}_{i \in S}, \{ iP_i \}_{i \in S}, cP)$

if $|S| < k$ **then return** $\perp$
foreach $i \in S$ **do**
  parse $(s_i, ct_i, h_i) := iP_i$
  $w'_i := \mathsf{SS.Rec}(s_i, w_i)$
  $sk'_i := \mathsf{Ext}(seed, w'_i)$
  if $h_i \neq \mathsf{H}(sk'_i, s_i, ct_i)$ **then return** $\perp$
  $csk_i := \mathsf{ElGamal.Dec}(sk'_i, ct_i)$
$csk := \mathsf{Shamir.Reconst}(S, (csk_i)_{i \in S})$
if $h \neq \mathsf{H}(csk, X)$ **then return** $\perp$
return $cR := X^{csk}$

Fig. 2: Ma et al's TFE scheme [36, Fig. 2]. We note that Ma et al. described $\mathsf{Shamir.Reconst}$ explicitly in their paper.

$\mathsf{TFE.Rep}(\mathsf{pp}, P, ((i_1, w_{i_1}), \ldots, (i_k, w_{i_k})))$**:** *The (deterministic) reproduction algorithm takes* pp*, a helper data* P*, and* k *tuples of an index and fuzzy data* $((i_1, w_{i_1}), \ldots, (i_k, w_{i_k}))$ *as input, and outputs a string* $R \in \mathcal{R}$ *or an error symbol* $\perp$*.*

We say that a TFE scheme TFE *satisfies correctness if the following holds: Let* $(w_1, \ldots, w_n), (w'_1, \ldots, w'_n) \in \mathcal{M}^n$ *be two sets of fuzzy data such that* $\mathsf{dis}(w_i, w'_i) \leq d$ *holds for each* $i \in [n]$*. Furthermore, let* $\mathsf{pp} \leftarrow \mathsf{TFE.Setup}(1^\lambda)$ *and* $(P, R) \leftarrow \mathsf{TFE.Gen}(\mathsf{pp}, k, n, (w_1, \ldots, w_n))$*. Then, for any mutually distinct* $i_1, \ldots, i_{k'} \in [n]$ *such that* $k \leq k' \leq n$*, it holds that*

$$\mathsf{TFE.Rep}(\mathsf{pp}, P, ((i_1, w'_{i_1}), \ldots, (i'_k, w'_{i_{k'}}))) = R.$$

Next, we give the definitions of the security notions for TFE: *information-theoretic security, (computational) reusability and (computational) robustness.* We define them by extending the existing security definitions for FEs [22, 48] to

the TFE setting. In all of our security notions, we consider an adversary who can know $k-1$ fuzzy data among the n fuzzy data under attack. Furthermore, we assume that each of W_i has sufficiently large average conditional min-entropy, i.e., $\tilde{\mathbf{H}}_\infty(W_i|\{W_j\}_{j\in[n\setminus i]}) \geq \ell_i$ for some ℓ_i. In this setting, when the set of fuzzy data is used only once, information-theoretic security guarantees that (R, P) is statistically indistinguishable from (U, P), where U is a truly random value; reusability guarantees that pairs $(R_i, P_i) \leftarrow \mathsf{Gen}(w_1 + \delta_{1,i}, \ldots, w_n + \delta_{n,i})$ generated multiple times with adversarially-chosen errors $\delta_{1,i}, \ldots, \delta_{n,i}$ are indistinguishable from (U_i, P_i) where U_i denotes a truly random value; Robustness guarantees that no adversary can forge a valid helper data P.

The formal definitions of these security notions are as follows.

Definition 12 (Information-Theoretic Security). *We say that a k-out-of-n TFE scheme $\mathsf{TFE} = (\mathsf{TFE.Setup}, \mathsf{TFE.Gen}, \mathsf{TFE.Rep})$ satisfies $(\mathcal{M}, (\ell_1, \ldots, \ell_n), \mathcal{R}, d, \epsilon)$-information-theoretic security, if for any joint distribution $W = (W_1, \ldots, W_n)$ over $\mathcal{M}^n$ such that $\tilde{\mathbf{H}}_\infty(W_i|\{W_j\}_{j\in[n\setminus i]}) \geq \ell_i$ holds for any $i \in [n]$, and for any computationally unbounded adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathsf{itsec}}_{\mathsf{TFE},\mathcal{A}}(\lambda) \coloneqq 2 \cdot \left| \Pr\left[b' = b : \begin{array}{l} b \xleftarrow{\$} \{0,1\}, \\ \mathsf{pp} \leftarrow \mathsf{TFE.Setup}(1^\lambda), \\ (w_1, \ldots, w_n) \xleftarrow{\$} W, \\ (j_1, \ldots, j_{k-1}) \leftarrow \mathcal{A}(\mathsf{pp}), \\ (P, R_0) \leftarrow \mathsf{TFE.Gen}(\mathsf{pp}, k, n, (w_1, \ldots, w_n)), \\ R_1 \xleftarrow{\$} \mathcal{R}, \\ b' \leftarrow \mathcal{A}(P, w_{j_1}, \ldots, w_{j_{k-1}}, R_b) \end{array} \right] - \frac{1}{2} \right| \leq \epsilon.$$

Definition 13 ((Computational) Reusability and Robustness). *We say that a k-out-of-n TFE scheme $\mathsf{TFE} = (\mathsf{TFE.Setup}, \mathsf{TFE.Gen}, \mathsf{TFE.Rep})$ satisfies $(\mathcal{M}, (\ell_1, \ldots, \ell_n), \mathcal{R}, d, \epsilon_1, \epsilon_2)$ robustness and reusability if for any (efficiently samplable[6]) joint distribution $W = (W_1, \ldots, W_n)$ over $\mathcal{M}^n$ such that $\tilde{\mathbf{H}}_\infty(W_i|\{W_j\}_{j\in[n\setminus i]}) \geq \ell_i$ holds for any $i \in [n]$, the following properties hold:*

Reusability: *For any PPT adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathsf{reu}}_{\mathsf{TFE},\mathcal{A}}(\lambda) \coloneqq 2 \cdot \left| \Pr[\mathsf{Game}^{\mathsf{reu}}_{\mathsf{TFE},\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right| \leq \epsilon_1,$$

where the game $\mathsf{Game}^{\mathsf{reu}}_{\mathsf{TFE},\mathcal{A}}(\lambda)$ is defined as in Fig. 3 (top).

Robustness: *For any PPT adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathsf{rob}}_{\mathsf{TFE},\mathcal{A}}(\lambda) \coloneqq \Pr[\mathsf{Game}^{\mathsf{rob}}_{\mathsf{TFE},\mathcal{A}}(\lambda) = 1] \leq \epsilon_2,$$

where the game $\mathsf{Game}^{\mathsf{rob}}_{\mathsf{TFE},\mathcal{A}}(\lambda)$ is defined as in Fig. 3 (bottom).

[6] For robustness and reusability of a TFE scheme, we only treat efficiently samplable fuzzy data distributions.

$\mathsf{Game}^{\mathsf{reu}}_{\mathsf{TFE},\mathcal{A}}(\lambda)$

$b \xleftarrow{\$} \{0,1\}$
$\mathsf{pp} \leftarrow \mathsf{TFE.Setup}(1^\lambda)$
$(w_1, \ldots, w_n) \xleftarrow{\$} W$
$(j_1, \ldots, j_{k-1}) \leftarrow \mathcal{A}(\mathsf{pp})$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Chal}}}(w_{j_1}, \ldots, w_{j_{k-1}})$
return $[\![b' = b]\!]$

$\mathcal{O}_{\mathsf{Chal}}(\delta_1, \ldots, \delta_n)$

if $\exists i \in [n] : \mathsf{dis}(0, \delta_i) > d$ **then return** $\perp$
$(P, R_0) \leftarrow \mathsf{TFE.Gen}(\mathsf{pp}, k, n, (w_1 + \delta_1, \ldots, w_n + \delta_n))$
$R_1 \xleftarrow{\$} \mathcal{R}$
return (P, R_b)

$\mathsf{Game}^{\mathsf{rob}}_{\mathsf{TFE},\mathcal{A}}(\lambda)$

$\mathsf{pp} \leftarrow \mathsf{TFE.Setup}(1^\lambda)$
$(w_1, \ldots, w_n) \xleftarrow{\$} W$
$L_{Gen} \coloneqq \emptyset$
$(j_1, \ldots, j_{k-1}) \leftarrow \mathcal{A}(\mathsf{pp})$
$(P^*, \delta^*_{i_1}, \ldots, \delta^*_{i_k}) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Gen}}}(w_{j_1}, \ldots, w_{j_{k-1}})$
if $\exists i_j \in \{i_1, \ldots, i_k\} : \mathsf{dis}(0, \delta^*_{i_j}) > d$ **then return** 0
if $P^* \in L_{Gen}$ **then return** 0
if $\mathsf{TFE.Rep}(\mathsf{pp}, P^*, (i_1, w_{i_1} + \delta^*_{i_1}), \ldots, (i_k, w_{i_k} + \delta^*_{i_k})) \neq \perp$ **then**
  return 1
else return 0

$\mathcal{O}_{\mathsf{Gen}}(\delta_1, \ldots, \delta_n)$

if $\exists i \in [n] : \mathsf{dis}(0, \delta_i) > d$ **then return** $\perp$
$(P, R) \leftarrow \mathsf{TFE.Gen}(\mathsf{pp}, k, n, (w_1 + \delta_1, \ldots, w_n + \delta_n))$
$L_{Gen} \coloneqq L_{Gen} \cup \{P\}$
return (P, R)

Fig. 3: Security game for reusability (top) and that for robustness (bottom) of a k-out-of-n TFE scheme TFE.

Remark 1 (How to Use TFE in Practice). Here, we give an example of how a TFE scheme defined above is used, using the use case scenario of thresholdizing the signing functionality described in Section 1. First of all, a public parameter pp is generated with $\mathsf{TFE.Setup}$. We assume pp is honestly generated. Next, to generate a signing key, key generation randomness R is generated from n fuzzy data with $\mathsf{TFE.Gen}$. Here, $\mathsf{TFE.Gen}$ is a single algorithm that runs on a single device. Therefore, we need to gather multiple fuzzy data onto a single device. When generating the signing key based on the biometric data of n different users, it is assumed that the n users will physically gather and have their biometrics scanned using the scanner provided by the device. The helper data P, output along with the string R, is stored, e.g., in the device or a cloud storage. We note that the security of TFE, as defined above, holds even if an adversary knows the helper data. To sign a message, the signing key is regenerated by reconstructing the randomness R from the helper data P and k fuzzy data via the $\mathsf{TFE.Rep}$ algorithm. If the helper data is stored on a cloud storage, the devices used to execute $\mathsf{TFE.Gen}$ and $\mathsf{TFE.Rep}$ can be different.

Remark 2 (On the Entropy Requirement on Fuzzy Data). In order for TFEs (or any other cryptographic primitives dealing with "fuzzy data" that needs to be kept private) to be useful for use cases in which biometric data is used as fuzzy data, in the first place, it is necessary that practical biometric data can actually have high enough entropy. This is highly dependent on what biometric feature we use and how we extract data from it, and investigating/evaluating particular biometric features/extraction methods effective for TFE is beyond our scope. Still, several existing works, e.g., [18, 31, 33, 42, 45], have shown evaluations/estimations on min-entropy (or related statistics) of some particular biometric features/extraction methods using real-world datasets of biometrics. More specifically, Tong et al. [45], Shukla et al. [42], and Kuznetsov et al. [33] used images of fingerprints, irises, and faces, respectively, as the source of fuzzy data in FE; Katsumata et al. [31] used finger-vein images in a primitive called fuzzy signature, and De Oliveira Nunes et al. [18] used fingerprint images in a primitive called oblivious extractor. We expect that these biometrics can be used for TFE as well.

Here we introduce one example from Katsumata et al. [31]. They experimentally evaluated a method for converting real-world finger-vein images into fuzzy data, using a dataset of finger-vein images from more than 500 people. According to [31, Table 2], in one setting of their method, extracting fuzzy data (represented as a real vector of 300 dimensions) from 4 finger-veins, they estimated $(\mathtt{FNMR}, \mathtt{FMR}) = (0.076, 2^{-128})$ where $\mathtt{FNMR}$ stands for False Non-Matching Rate and denotes the probability that two fuzzy data extracted from the same user is identified as coming from different users, while $\mathtt{FMR}$ stands for False Matching Rate and denotes the probability that two fuzzy data extracted from different users is accidentally identified as coming from the same user. Here, note that $\mathtt{FNMR}$ corresponds to the correctness error (which can be easily reduced by simply repeating the extraction method multiple times), and $-\log_2 \mathtt{FMR}$ lower-bounds

the entropy of the fuzzy data. Thus, finger-veins (extracted using the method of [31]) should also be usable in the context of TFE.

In each of our security definitions for TFE, we require each coordinate W_i of fuzzy data distribution $W = (W_1, \ldots, W_n)$ to have high (average conditional) min-entropy conditioned on the remaining coordinates $\{W_j\}_{j \in [n \setminus i]}$. We think that this assumption is reasonable in use cases in which we think of biometric features (extracted from the human body in some way) as the source of fuzzy data, and each coordinate W_i is taken from different users.

5 Our Information-Theoretic Secure TFE Scheme

In this section, we present a construction of an information-theoretically secure k-out-of-n TFE scheme. Let $\mathcal{R}$ be an arbitrary group. Let $\mathsf{SS} = (\mathsf{SS.Gen}, \mathsf{SS.Rec})$ be an $(\mathcal{M}, \ell, \tilde{\ell}, d)$-secure sketch scheme, $\mathsf{SecretShare} = (\mathsf{Share}, \mathsf{Reconst})$ be a k-out-of-n secret sharing scheme over $\mathcal{R}$, $\mathsf{OTP} = (\mathsf{OTP.Enc}, \mathsf{OTP.Dec})$ be a one-time pad (OTP) encryption scheme over $\mathcal{R}$, and let $\mathcal{H} = \{\mathsf{H} : \mathcal{M} \to \mathcal{R}\}$ be a universal hash family. Using these building blocks, our information-theoretically secure k-out-of-n TFE scheme InfTFE with a randomness space $\mathcal{R}$ is constructed as in Fig. 4.

At a high level, $\mathsf{InfTFE.Gen}$ generates a random string $R \xleftarrow{\$} \mathcal{R}$ and its n share of the k-out-of-n secret sharing scheme $\mathsf{SecretShare}$. Then, each share is encrypted using the one-time pad encryption scheme with a secret key generated from the corresponding fuzzy data. Therefore, $\mathsf{InfTFE.Rep}$ can recover the randomness R by using k fuzzy data w'_{i_j} such that each of them is close to the original data.

The correctness of InfTFE follows from the correctness of the underlying building blocks. Specifically, suppose $(P = (s_i, c_i)_{i \in [n]}, R = m)$ is generated from $\mathsf{InfTFE.Gen}$ using fuzzy data $(w_1, \ldots, w_n)$ as input, and suppose $\mathsf{InfTFE.Rep}$ is executed with input P and $((i_1, w_i), \ldots, (i_k, w_k))$. Suppose further that $\mathsf{dis}(w_j, w'_j) \leq d$ holds for each $j \in \{i_1, \ldots, i_k\}$. Then, by the correctness of the underlying secure sketch scheme SS, for each $j \in \{i_1, \ldots, i_k\}$, w_j can be correctly recovered in $\mathsf{InfTFE.Rep}$, and thus the secret key $\mathsf{sk}_j = \mathsf{H}(w_j)$ can be recovered. Then, due to the correctness of the OTP scheme, the k shares $\{m_{i_j}\}_{j \in [k]}$ (out of n shares of m) are correctly recovered. Finally, due to the correctness of the underlying secret sharing scheme $\mathsf{SecretShare}$, the randomness $R = m$ can be precisely reproduced.

We now show that InfTFE satisfies information-theoretic security. An intuition behind the security proof is as follows. By definition, an adversary can know only $k-1$ fractions of n fuzzy data used for randomness generation. Therefore, to recover the string R, it needs to obtain additional shares encrypted with the remaining $n-k+1$ uncorrupted fuzzy data. However, since we assume the conditional min-entropy $\tilde{\mathbf{H}}_\infty(W_i | \{W_j\}_{j \in [n \setminus i]})$ is sufficiently large for each $i \in [n]$, the security of the secure sketch scheme and the generalized leftover hash lemma guarantee that the uncorrupted OTP secret keys look random even if the adversary knows sketches $\{s_i\}_{i \in [n]}$ and $k-1$ fuzzy data. Thus, the OTP encryption

$\underline{\mathsf{InfTFE.Setup}(1^\lambda)}$

$\mathsf{H} \xleftarrow{\$} \mathcal{H}$
return $\mathsf{pp} := \mathsf{H}$

$\underline{\mathsf{InfTFE.Gen}(\mathsf{pp}, k, n, (w_1, \ldots, w_n))}$

$m \xleftarrow{\$} \mathcal{R}$
$(m_1, \ldots, m_n) \leftarrow \mathsf{Share}(k, n, m)$
foreach $i \in [n]$ **do**
  $s_i := \mathsf{SS.Gen}(w_i)$
  $\mathsf{sk}_i := \mathsf{H}(w_i)$
  $c_i := \mathsf{OTP.Enc}(\mathsf{sk}_i, m_i)$
return $(P, R) := ((s_i, c_i)_{i\in[n]}, m)$

$\underline{\mathsf{InfTFE.Rep}(\mathsf{pp}, P, ((i_1, w_{i_1}), \ldots, (i_k, w_{i_k})))}$

parse $(s_i, c_i)_{i\in[n]} := P$
$J := \{i_1, \ldots, i_k\}$
foreach $i \in J$ **do**
  $w'_i := \mathsf{SS.Rec}(s_i, w_i)$
  $\mathsf{sk}'_i := \mathsf{H}(w'_i)$
  $m_i := \mathsf{OTP.Dec}(\mathsf{sk}'_i, c_i)$
return $R := \mathsf{Reconst}(J, \{m_i\}_{i\in J})$

Fig. 4: Our information-theoretically secure k-out-of-n TFE scheme InfTFE.

scheme perfectly hides the information on the $n - k + 1$ uncorrupted shares. Therefore, the adversary cannot obtain more than $k - 1$ shares of R, which in turn implies that R is hidden from the adversary's view by the security of the secret sharing scheme.

We now give the formal statement for the security of InfTFE.

Theorem 1. *Let $\ell, \tilde{\ell}, \ell_1, \ldots, \ell_n$ be positive real numbers such that $\ell_i \geq \ell$ for all $i \in [n]$. Assume that* SS *is an $(\mathcal{M}, \ell, \tilde{\ell}, d)$-secure sketch scheme, and the k-out-of-n secret sharing scheme* SecretShare *and the OTP encryption scheme* OTP *are perfectly secure. Then,* InfTFE *satisfies $(\mathcal{M}, (\ell_1, \ldots, \ell_n), \mathcal{R}, d, \epsilon)$-information-theoretic security with $\epsilon = n\sqrt{|\mathcal{R}| \cdot 2^{-\tilde{\ell}}}$. In particular, if $\tilde{\ell} = \log_2 |\mathcal{R}| + \omega(\log_2 \lambda)$, then ϵ is negligible in λ.*

Proof. Let $W = (W_1, \ldots, W_n)$ be a joint distribution over $\mathcal{M}^n$ such that $\tilde{\mathbf{H}}_\infty(W_i | \{W_j\}_{j\in[n\setminus i]}) \geq \ell_i \geq \ell$ holds for all $i \in [n]$. Let $\mathcal{A}$ be an arbitrary computationally unbounded adversary that attacks the information-theoretic security of InfTFE. We will prove the security of InfTFE using the following sequence of games $\mathsf{Game}_0, \ldots, \mathsf{Game}_3$. (In the following, we overload the notation and let Game_t for $t \in \{0, \ldots, 3\}$ denote the event that $\mathcal{A}$ succeeds in guessing the challenge bit in Game_t.)

Game_0. This is the original game for the information-theoretic security of InfTFE.
By definition, we have $\mathsf{Adv}^{\mathsf{itsec}}_{\mathsf{InfTFE},\mathcal{A}}(\lambda) = 2 \cdot \left|\Pr[\mathsf{Game}_0] - \frac{1}{2}\right|$.

Game_1. In this game, after $\mathcal{A}$ outputs $(j_1, \ldots, j_{k-1})$, sk_i for each $i \in [n] \setminus \{j_1, \ldots, j_{k-1}\}$ is generated as $\mathsf{sk}_i \xleftarrow{\$} \mathcal{R}$, instead of being computed as $\mathsf{sk}_i := \mathsf{H}(w_i)$.

We will show the following lemma.

Lemma 2. *It holds that* $|\Pr[\mathsf{Game}_0] - \Pr[\mathsf{Game}_1]| \leq \frac{n}{2}\sqrt{|\mathcal{R}| \cdot 2^{-\tilde{\ell}}}$.

Proof. (of Lemma 2) For $\beta \in \{0, \ldots, n\}$, let $\mathsf{Game}_{0.\beta}$ be the game defined in the same manner as Game_0, except that sk_i for $i \leq \beta$ with $i \notin \{j_1, \ldots, j_{k-1}\}$ is generated as $\mathsf{sk}_i \xleftarrow{\$} \mathcal{R}$, instead of being computed as $\mathsf{sk}_i := \mathsf{H}(w_i)$. Then, by definition, $\mathsf{Game}_{0.0}$ (resp. $\mathsf{Game}_{0.n}$) is identical to Game_0 (resp. Game_1). Hence, we have

$$\begin{aligned} |\Pr[\mathsf{Game}_0] - \Pr[\mathsf{Game}_1]| &= |\Pr[\mathsf{Game}_{0.0}] - \Pr[\mathsf{Game}_{0.n}]| \\ &= \left| \sum_{\beta \in [n]} \left(\Pr[\mathsf{Game}_{0.(\beta-1)}] - \Pr[\mathsf{Game}_{0.\beta}] \right) \right|. \end{aligned} \tag{1}$$

Furthermore, note that for all $\beta \in [n]$, if $\beta \in \{j_1, \ldots, j_{k-1}\}$, then the distribution of $\mathcal{A}$'s view in $\mathsf{Game}_{0.(\beta-1)}$ and that in $\mathsf{Game}_{0.\beta}$ are identical. (This observation will be used in the analysis of the reduction $\mathcal{B}$ below.)

We prove the lemma by using the generalized leftover hash lemma (Lemma 1). As a preparation, consider the values x and z generated by the following procedure parameterized by $i \in [n]$:

$$\boxed{\begin{array}{l} (w_1, \ldots, w_n) \xleftarrow{\$} W \\ x := w_i \\ z := (\mathsf{SS.Gen}(w_i), w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_n) \end{array}}$$

Let Z denote the distribution of z generated by the above procedure, and for $i \in [n]$, let $W_{n \setminus i}$ denote the distribution of $(w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_n)$. Then, (W_i, Z) forms a joint distribution such that $\tilde{\mathbf{H}}_\infty(W_i|Z) \geq \tilde{\ell}$ holds, due to the security of SS and the given condition that $\tilde{\mathbf{H}}_\infty(W_i|W_{n \setminus i}) \geq \ell$ holds for all $i \in [n]$. Using this joint distribution (W_i, Z), we consider the following computationally unbounded adversary $\mathcal{B}$ against the GLHL game, using $\mathcal{A}$.

1. $\mathcal{B}$ chooses $\alpha \in [n]$ uniformly at random, sends the description of the joint distribution (W_α, Z) defined above to the challenger.
2. Upon receiving $(\mathsf{H}, y, z = (s = \mathsf{SS.Gen}(w_\alpha), \{ w_j \}_{j \in [n \setminus \alpha]}))$ from the challenger, $\mathcal{B}$ randomly samples the challenge bit $d \in \{0, 1\}$ for $\mathcal{A}$, sets $\mathsf{pp} := \mathsf{H}$, and sends pp to $\mathcal{A}$.
3. When $\mathcal{A}$ sends $(j_1, \ldots, j_{k-1})$, $\mathcal{B}$ computes (P, R_0) as follows:
   (a) $\mathcal{B}$ samples $m \xleftarrow{\$} \mathcal{R}$ and computes $(m_1, \ldots, m_n) \leftarrow \mathsf{Share}(k, n, m)$.
   (b) If $\alpha \in \{j_i, \ldots, j_{k-1}\}$, then $\mathcal{B}$ samples $\widehat{w}_\alpha$ according to the distribution W_α conditioned on $Z = z$, namely, $W_j = w_j$ for all $j \in [n \setminus \alpha]$ and $\mathsf{SS.Gen}(\widehat{w}_\alpha) = s$ $(= \mathsf{SS.Gen}(w_\alpha))$. (Note that the conditional resampling considered here cannot be done efficiently in general. However, $\mathcal{B}$ can perform this since it is computationally unbounded. Note

also that the distribution of $(w_1, \ldots, w_{\alpha-1}, \widehat{w}_\alpha, w_{\alpha+1}, \ldots, w_n)$ is identical to that of the original fuzzy data $(w_1, \ldots, w_n)$ sampled according to $W = (W_1, \ldots, W_n)$ by the challenger in the GLHL game.)

(c) For each $i \in [n]$, $\mathcal{B}$ proceeds as follows:
- If $i \in \{j_1, \ldots, j_{k-1}\}$, then $\mathcal{B}$ computes $s_i := \mathsf{SS.Gen}(w_i)$, $\mathsf{sk}_i := \mathsf{H}(w_i)$, and $c_i := \mathsf{OTP.Enc}(\mathsf{sk}_i, m_i)$, where if $i = \alpha$ and thus $\mathcal{B}$ needs w_α for computing sk_α, then $\mathcal{B}$ uses $\widehat{w}_\alpha$ chosen above as w_α.
- If $i \in [n] \setminus \{j_1, \ldots, j_{k-1}\}$, then $\mathcal{B}$ proceeds as follows:
  - If $i < \alpha$, then $\mathcal{B}$ computes $s_i := \mathsf{SS.Gen}(w_i)$, samples $\mathsf{sk}_i \xleftarrow{\$} \mathcal{R}$, and computes $c_i := \mathsf{OTP.Enc}(\mathsf{sk}_i, m_i)$.
  - If $i = \alpha$, then $\mathcal{B}$ sets $s_\alpha := s$, $\mathsf{sk}_\alpha := y$, and $c_\alpha := \mathsf{OTP.Enc}(\mathsf{sk}_\alpha, m_\alpha)$.[7]
  - If $i > \alpha$, then $\mathcal{B}$ computes $s_i := \mathsf{SS.Gen}(w_i)$, $\mathsf{sk}_i := \mathsf{H}(w_i)$, and $c_i := \mathsf{OTP.Enc}(\mathsf{sk}_i, m_i)$.

(d) $\mathcal{B}$ sets $P := (s_i, c_i)_{i \in [n]}$ and $R_0 := m$.

4. $\mathcal{B}$ samples $R_1 \xleftarrow{\$} \mathcal{R}$, and sends $(P, w_{j_1}, \ldots, w_{j_{k-1}}, R_d)$ to $\mathcal{A}$, where if $\alpha \in \{j_1, \ldots, j_{k-1}\}$ and thus $\mathcal{B}$ needs w_α, then $\mathcal{B}$ uses $\widehat{w}_\alpha$ as w_α.
5. When $\mathcal{A}$ outputs d' and terminates, $\mathcal{B}$ sets $b' := [\![d' = d]\!]$, and terminates with output b'.

Let b be the challenge bit for $\mathcal{B}$ in the GLHL game. Note that $\mathcal{A}$ chooses the indices $\{j_1, \ldots, j_{k-1}\}$ after seeing only H, and thus for each $\beta \in [n]$, whether $\beta \in \{j_1, \ldots, j_{k-1}\}$ occurs is independent of b. Furthermore, if $\alpha = \beta$ and $\beta \in \{j_1, \ldots, j_{k-1}\}$, then $\mathcal{B}$ does not use the challenge instance y at all, and thus $\mathcal{B}$'s behavior (and consequently, $\mathcal{A}$'s entire view) is totally independent of b. Note also that in this case, $\mathcal{B}$ simulates $\mathsf{Game}_{0.(\beta-1)}$ perfectly for $\mathcal{A}$ (and its entire view is distributed identically to that in $\mathsf{Game}_{0.\beta}$ as well, as observed earlier). This is due to the conditional resampling of $\widehat{w}_\alpha$ performed by $\mathcal{B}$, which ensures that the distribution of $(w_1, \ldots, w_{\alpha-1}, \widehat{w}_\alpha, w_{\alpha+1}, \ldots, w_n)$ is identical to that of the original fuzzy data $(w_1, \ldots, w_n)$ sampled according to $W = (W_1, \ldots, W_n)$. Using them, $\mathcal{B}$ computes $(P, w_{j_1}, \ldots, w_{j_{k-1}}, R_d)$ in exactly the same way as done in $\mathsf{Game}_{0.(\beta-1)}$ (and $\mathsf{Game}_{0.\beta}$).

We now argue that for each $\beta \in [n]$, if $\alpha = \beta$ and $b = 1$, then $\mathcal{B}$ simulates $\mathsf{Game}_{0.(\beta-1)}$ perfectly for $\mathcal{A}$, regardless of whether $\beta \in \{j_1, \ldots, j_{k-1}\}$ or not. Specifically, in this case, y is computed as $y := \mathsf{H}(w_\beta)$. Now, note that if $\beta \notin \{j_1, \ldots, j_{k-1}\}$, then y is used as sk_β. Thus, sk_i for $i \leq \beta - 1$ and $i \notin \{j_1, \ldots, j_{k-1}\}$ is a truly random value, and sk_i for $i \geq \beta$ or $i \in \{j_1, \ldots, j_{k-1}\}$ is computed as $\mathsf{sk}_i := \mathsf{H}(w_i)$, which is exactly how sk_i's in $\mathsf{Game}_{0.(\beta-1)}$ are generated. On the other hand, if $\beta \in \{j_1, \ldots, j_{k-1}\}$, then, as mentioned earlier, $\mathcal{B}$ simulates $\mathsf{Game}_{0.(\beta-1)}$ perfectly for $\mathcal{A}$. Hence, $\mathcal{B}$ simulates $\mathsf{Game}_{0,(\beta-1)}$ perfectly for $\mathcal{A}$, regardless of whether $\beta \in \{j_1, \ldots, j_{k-1}\}$ or not. Since $\mathcal{B}$ outputs 1 only when $d' = d$ occurs, for all $\beta \in [n]$, we have

$$\Pr[b' = 1 | \alpha = \beta \wedge b = 1] = \Pr[\mathsf{Game}_{0,(\beta-1)}].$$

[7] Note that this step is performed only if $\alpha \notin \{j_1, \ldots, j_{k-1}\}$.

We next argue that if $\alpha = \beta$ and $b = 0$, then $\mathcal{B}$ simulates $\mathsf{Game}_{0,\beta}$ perfectly for $\mathcal{A}$. Specifically, in this case, y is a truly random value, and if $\beta \notin \{j_1, \ldots, j_{k-1}\}$, then y is used as sk_β. Then, sk_i for $i \leq \beta$ and $i \notin \{j_1, \ldots, j_{k-1}\}$ is a truly random value, and sk_i for $i \geq \beta + 1$ or $i \in \{j_1, \ldots, j_{k-1}\}$ is computed as $\mathsf{sk}_i \coloneqq \mathsf{H}(w_i)$, which is exactly how sk_i's in $\mathsf{Game}_{0.\beta}$ are generated. On the other hand, if $\beta \in \{j_1, \ldots, j_{k-1}\}$, then $\mathcal{B}$ simulates $\mathsf{Game}_{0,\beta}$ perfectly for $\mathcal{A}$, as mentioned earlier. Hence, with a similar argument to the above, for all $\beta \in [n]$, we have

$$\Pr[b' = 1 | \alpha = \beta \wedge b = 0] = \Pr[\mathsf{Game}_{0,\beta}].$$

Since α is chosen uniformly at random from $[n]$, independently of b, for all $\beta \in [n]$, we have

$$\Pr[\alpha = \beta | b = 1] = \Pr[\alpha = \beta | b = 0] = \frac{1}{n}.$$

We can now calculate $\mathsf{Adv}^{\mathsf{LHL}}_{\mathcal{H},\mathcal{B}}(\lambda)$ as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{LHL}}_{\mathcal{H},\mathcal{B}}(\lambda) &= |\Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0]| \\
&= \left| \sum_{\beta \in [n]} \Pr[b' = 1 \wedge \alpha = \beta | b = 1] - \sum_{\beta \in [n]} \Pr[b' = 1 \wedge \alpha = \beta | b = 0] \right| \\
&= \left| \sum_{\beta \in [n]} \Pr[\alpha = \beta | b = 1] \cdot \Pr[b' = 1 | \alpha = \beta \wedge b = 1] \right. \\
&\qquad \left. - \sum_{\beta \in [n]} \Pr[\alpha = \beta | b = 0] \cdot \Pr[b' = 1 | \alpha = \beta \wedge b = 0] \right| \\
&= \frac{1}{n} \left| \sum_{\beta \in [n]} \left(\Pr[\mathsf{Game}_{0.(\beta-1)}] - \Pr[\mathsf{Game}_{0.\beta}] \right) \right| \\
&\leq \frac{1}{2} \sqrt{|\mathcal{R}| \cdot 2^{-\tilde{\ell}}},
\end{aligned}$$

where the final inequality is due to $\tilde{\mathbf{H}}_\infty(W_\alpha | Z) \geq \tilde{\ell}$ and the generalized leftover hash lemma (Lemma 1). The combination of the final inequality here and Eq. (1) imply Lemma 2. □

Game_2. In this game, for $i \in [n] \setminus \{j_1, \ldots, j_{k-1}\}$, c_i is computed as $c_i \coloneqq \mathsf{OTP.Enc}(\mathsf{sk}_i, 0)$. Since sk_i for $i \in [n] \setminus \{j_1, \ldots, j_{k-1}\}$ is chosen uniformly at random, the perfect security of OTP guarantees that $\mathcal{A}$'s views in the two games are distributed identically. Thus, we have $\Pr[\mathsf{Game}_1] = \Pr[\mathsf{Game}_2]$.

Game_3. In this game, an independent randomness $m' \xleftarrow{\$} \mathcal{M}$ is sampled and $\{m'_i\}_{i \in [n]} \leftarrow \mathsf{Share}(k, n, m')$ is generated. Then, c_i for each $i \in \{j_1, \ldots, j_{k-1}\}$ is computed as $c_i \coloneqq \mathsf{OTP.Enc}(\mathsf{sk}_i, m'_i)$.

To prove the indistinguishability of Game_2 and Game_3, we consider the following adversary $\mathcal{B}'$ against the security of $\mathsf{SecretShare}$, using $\mathcal{A}$.

1. $\mathcal{B}'$ randomly samples the challenge bit b for $\mathcal{A}$, samples $\mathsf{H} \xleftarrow{\$} \mathcal{H}$ and $(w_1, \ldots, w_n) \xleftarrow{\$} W$, and sends $\mathsf{pp} \coloneqq \mathsf{H}$ to $\mathcal{A}$.
2. When $\mathcal{A}$ sends $(j_1, \ldots, j_{k-1})$, $\mathcal{B}'$ computes (P, R_0) as follows:
   (a) $\mathcal{B}'$ samples $m, m' \xleftarrow{\$} \mathcal{R}$, sends $(J \coloneqq \{ j_1, \ldots, j_{k-1} \}, m', m)$ to its challenger, and receives $(m_i)_{i \in J}$.
   (b) For each $i \in [n]$, $\mathcal{B}'$ proceeds as follows.
      i. $\mathcal{B}'$ computes $s_i \coloneqq \mathsf{SS.Gen}(w_i)$ and $\mathsf{sk}_i \coloneqq \mathsf{H}(w_i)$.
      ii. If $i \in \{ j_1, \ldots, j_{k-1} \}$ holds, then $\mathcal{B}'$ computes $c_i \coloneqq \mathsf{OTP.Enc}(\mathsf{sk}_i, m_i)$. Otherwise, $\mathcal{B}'$ computes $c_i \coloneqq \mathsf{OTP.Enc}(\mathsf{sk}_i, 0)$.
   (c) $\mathcal{B}'$ sets $P \coloneqq (s_i, c_i)_{i \in [n]}$ and $R_0 \coloneqq m$.
3. $\mathcal{B}'$ samples $R_1 \xleftarrow{\$} \mathcal{R}$ and sends $(P, w_{j_1}, \ldots, w_{j_{k-1}}, R_b)$ to $\mathcal{A}$.
4. When $\mathcal{A}$ outputs b' and terminates, $\mathcal{B}'$ outputs $[\![b' = b]\!]$ and terminates.

If $\mathcal{B}'$ receives the secret shares of m from the challenger, then it perfectly simulates Game_2 for $\mathcal{A}$, while if $\mathcal{B}$ receives the secret shares of m', it perfectly simulates Game_3 for $\mathcal{A}$. Since $\mathcal{B}$ outputs 1 only when $b' = b$ occurs, due to the perfect security of $\mathsf{SecretShare}$, we have $\Pr[\mathsf{Game}_2] = \Pr[\mathsf{Game}_3]$.

In Game_3, both R_0 and R_1 are generated uniformly at random, and both of them are independent of P. Therefore, $\mathcal{A}$'s view in Game_3 is distributed identically, independently of the challenge bit. Thus, we have $\Pr[\mathsf{Game}_3] = 1/2$.

Combining everything together, we have

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{itsec}}_{\mathsf{InfTFE},\mathcal{A}}(\lambda) &= 2 \cdot \left| \Pr[\mathsf{Game}_0] - \frac{1}{2} \right| \\
&\leq 2 \cdot \left| \Pr[\mathsf{Game}_1] - \frac{1}{2} \right| + n\sqrt{|\mathcal{R}| \cdot 2^{-\tilde{\ell}}} \\
&= 2 \cdot \left| \Pr[\mathsf{Game}_2] - \frac{1}{2} \right| + n\sqrt{|\mathcal{R}| \cdot 2^{-\tilde{\ell}}} \\
&= 2 \cdot \left| \Pr[\mathsf{Game}_3] - \frac{1}{2} \right| + n\sqrt{|\mathcal{R}| \cdot 2^{-\tilde{\ell}}} \\
&= n\sqrt{|\mathcal{R}| \cdot 2^{-\tilde{\ell}}}.
\end{aligned}$$

This completes the proof of Theorem 1. □

6 Our Reusable and Robust TFE Scheme

In this section, we present our reusable and robust TFE scheme. The construction framework is identical to our information-theoretically secure scheme. To achieve reusability and robustness, we modify it as follows.

- We require that the secure sketch scheme SS be homomorphic and linear, and the universal hash function H be homomorphic. Roughly speaking, these properties ensure reusability and robustness even if an adversary chooses errors on fuzzy data.

– We replace the OTP encryption scheme with an AIAE scheme. To prevent tampering of ciphertexts, an integrity property is required. Furthermore, we assume the AIAE scheme is related-key secure. This prevents leakage of encrypted secret shares even when (nearly) identical secret keys are reused, and an adversary knows errors in those keys.
– We authenticate the set of sketches and ciphertexts, namely $(s_i, c_i)_{i\in[n]}$, using an OTS scheme. More specifically, Gen generates a signature $\sigma \leftarrow \mathsf{OTS.Sign}(\mathsf{osk}, (s_i, c_i)_{i\in[n]})$ and the corresponding verification key ovk is linked with each ciphertext c_i as the auxiliary input of AIAE. Thanks to this, no adversary can create a valid signature for honestly-generated ciphertexts, since the adversary does not know the one-time signing key, thereby preventing the reuse of helper data P to create a malformed one.

Formally, our proposed construction uses the following building blocks:

– An OTS scheme $\mathsf{OTS} = (\mathsf{OTS.Gen}, \mathsf{OTS.Sign}, \mathsf{OTS.Ver})$.
– An AIAE scheme $\mathsf{AIAE} = (\mathsf{AIAE.Setup}, \mathsf{AIAE.Enc}, \mathsf{AIAE.Dec})$ with a key space $\mathcal{K}$, a plaintext space $\mathcal{M}_{\mathsf{AIAE}}$, and an auxiliary input space $\{0,1\}^*$.
– A secure sketch scheme $\mathsf{SS} = (\mathsf{SS.Gen}, \mathsf{SS.Rec})$.
– A secret sharing scheme $\mathsf{SecretShare} = (\mathsf{Share}, \mathsf{Reconst})$ over $\mathcal{M}_{\mathsf{AIAE}}$.
– A universal hash family $\mathcal{H} = \{\mathsf{H} : \mathcal{M} \to \mathcal{K}\}$.

Using them, our proposed reusable and robust TFE scheme $\mathsf{rrTFE} = (\mathsf{rrTFE.Setup}, \mathsf{rrTFE.Gen}, \mathsf{rrTFE.Rep})$ is constructed as in Fig. 5.

We can see that rrTFE is correct if OTS, AIAE, $\mathsf{SecretShare}$, and SS are correct. More precisely, due to the correctness of OTS, the OTS signature σ is correctly verified. Then, since $\mathsf{dis}(w_i, w'_i) \le d$ holds for any $i \in [n]$, by the correctness of SS, w'_j can be correctly recovered for any $j \in \{i_1, \ldots, i_k\}$. Thus, the secret key satisfies $\mathsf{sk}_j = \mathsf{H}(w'_j)$. Then, by the correctness of AIAE, the k shares $\{m_j\}_{j\in\{i_1,\ldots,i_k\}}$ are correctly recovered. Finally, due to the correctness of $\mathsf{SecretShare}$, the randomness $R = m$ can be precisely reproduced.

We now give the formal statement for the security of rrTFE.

Theorem 2. *Let $\ell, \tilde{\ell}, \ell_1, \ldots, \ell_n$ be positive real numbers such that $\ell_i \ge \ell$ for all $i \in [n]$. Assume AIAE satisfies IND-Φ-RKA security and weak INT-Φ-RKA security for key-shift function family $\Phi = \{\phi_\Delta : \mathcal{K} \to \mathcal{K} \mid \phi_\Delta(x) = x + \Delta\}_{\Delta\in\mathcal{K}}$, OTS satisfies $\mathsf{sEUF\text{-}CMA}$ security, $\mathsf{SecretShare}$ is perfectly secure, $\mathcal{H}$ is homomorphic, and SS is an $(\mathcal{M}, \ell, \tilde{\ell}, d)$-secure sketch scheme with linearity and homomorphism. Assume further that $\tilde{\ell} = \log_2 |\mathcal{K}| + \omega(\log_2 \lambda)$. Then, rrTFE satisfies $(\mathcal{M}, (\ell_1, \ldots, \ell_n), \mathcal{M}_{\mathsf{AIAE}}, d, \mathsf{negl}(\lambda), \mathsf{negl}(\lambda))$ reusability and robustness.*

More specifically, for any PPT adversaries $\mathcal{A}$ and $\mathcal{A}'$, there exist PPT adversaries $\mathcal{B}$, $\mathcal{B}'_1$, and $\mathcal{B}'_2$ such that

$$\mathsf{Adv}^{\mathsf{reu}}_{\mathsf{rrTFE},\mathcal{A}}(\lambda) \le 2(n-k+1)\mathsf{Adv}^{\mathsf{ind\text{-}rka}}_{\mathsf{AIAE},\mathcal{B}}(\lambda) + n\sqrt{|\mathcal{K}| \cdot 2^{-\tilde{\ell}}},$$

$$\mathsf{Adv}^{\mathsf{rob}}_{\mathsf{rrTFE},\mathcal{A}'}(\lambda) \le q_T \mathsf{Adv}^{\mathsf{unf}}_{\mathsf{OTS},\mathcal{B}'_1}(\lambda) + (n-k+1)\left(\mathsf{Adv}^{\mathsf{int\text{-}rka}}_{\mathsf{AIAE},\mathcal{B}'_2}(\lambda) + \frac{1}{2}\sqrt{|\mathcal{K}| \cdot 2^{-\tilde{\ell}}}\right),$$

where q_T denotes the maximum number of queries to $\mathcal{O}_{\mathsf{Gen}}$ made by $\mathcal{A}'$.

$\mathsf{rrTFE.Setup}(1^\lambda)$

$\mathsf{H} \xleftarrow{\$} \mathcal{H}$

$\mathsf{pp}_{\mathsf{AIAE}} \xleftarrow{\$} \mathsf{AIAE.Setup}(1^\lambda)$

return $\mathsf{pp} := (\mathsf{H}, \mathsf{pp}_{\mathsf{AIAE}})$

$\mathsf{rrTFE.Gen}(\mathsf{pp}, k, n, (w_1, \ldots, w_n))$

$m \xleftarrow{\$} \mathcal{M}_{\mathsf{AIAE}}$

$(m_1, \ldots, m_n) \leftarrow \mathsf{Share}(k, n, m)$

$(\mathsf{ovk}, \mathsf{osk}) \leftarrow \mathsf{OTS.Gen}(1^\lambda)$

foreach $i \in [n]$ **do**

  $s_i := \mathsf{SS.Gen}(w_i)$

  $\mathsf{sk}_i := \mathsf{H}(w_i)$

  $c_i \leftarrow \mathsf{AIAE.Enc}(\mathsf{sk}_i, m_i, \mathsf{ovk})$

$\sigma \leftarrow \mathsf{OTS.Sign}(\mathsf{osk}, (s_i, c_i)_{i\in[n]})$

$P := ((s_i, c_i)_{i\in[n]}, \mathsf{ovk}, \sigma)$

$R := m$

return (P, R)

$\mathsf{rrTFE.Rep}(\mathsf{pp}, P, ((i_1, w_{i_1}), \ldots, (i_k, w_{i_k})))$

parse $((s_i, c_i)_{i\in[n]}, \mathsf{ovk}, \sigma) := P$

if $\mathsf{OTS.Ver}(\mathsf{ovk}, ((s_i, c_i)_{i\in[n]}, \sigma) = 0$ **then**

  return $\perp$

$J := \{i_1, \ldots, i_k\}$

foreach $i \in J$ **do**

  $w'_i := \mathsf{SS.Rec}(s_i, w_i)$

  $\mathsf{sk}'_i := \mathsf{H}(w'_i)$

  $m_i \leftarrow \mathsf{AIAE.Dec}(\mathsf{sk}'_i, c_i, \mathsf{ovk})$

if $\forall i \in J : m_i \neq \perp$ **then**

  return $R := \mathsf{Reconst}(J, \{m_i\}_{i\in J})$

else return $\perp$

Fig. 5: Our reusable and robust k-out-of-n TFE scheme rrTFE.

Due to the page limitation, the formal proof is given in the full version of this paper. Here, we provide an intuitive explanation of the proof.

To prove reusability, we show that the answers from the $\mathcal{O}_{\mathsf{Chal}}$ oracle leak no information about the generated randomness R, thereby an adversary cannot distinguish R from a random value. More precisely, we show that the AIAE ciphertexts corresponding to $n-k+1$ uncorrupted fuzzy data leak no information about encrypted secret shares, and thus the adversary cannot recover secret-shared randomness R. First, using an information-theoretic argument, we can prove that uncorrupted $n - k + 1$ AIAE secret keys are indistinguishable from random, provided the secure sketch scheme SS is secure, and the min-entropy assumption about the joint distribution holds. Next, we employ the homomorphic properties of the secure sketch scheme SS and the hash function H, and the confidentiality of AIAE against related-key attacks with respect to key-shift functions. Roughly speaking, the homomorphic property ensures that the adversary, who chooses errors δ on fuzzy data, must know the errors on the AIAE secret key sk; the confidentiality against related-key attacks guarantees that ciphertexts leak no information about their plaintexts even if the adversary knows errors on the secret key. As a result, we can conclude that the helper data returned by the $\mathcal{O}_{\mathsf{Chal}}$ oracle computationally hides information about R. This leads to the reusability of rrTFE.

To prove robustness, we show that an adversary can neither reuse honestly-generated helper data as a forgery P^*, nor create a new valid AIAE ciphertext c^* without knowing the AIAE secret key. The former follows from the strong unforgeability of OTS. Since the set $(s_i, c_i)_{i\in[n]}$ is authenticated with an OTS signature, and the corresponding verification key is linked with each AIAE ciphertext c_i as its auxiliary information, the adversary cannot create a new valid OTS signature for an honestly-generated helper data. The latter follows from the integrity of AIAE against related-key attacks. Since the adversary knows only $k-1$ fuzzy data while k fuzzy data is needed for reproduction, the adversary needs to forge a valid (s_i^*, c_i^*) for an uncorrupted fuzzy data w_i^*. However, it is infeasible due to the linearity of SS and the integrity of AIAE: linearity ensures that the adversary knows the error on the secret key sk_{i^*}, which corresponds to s_{i^*} and $\delta_{i^*}^*$, and the integrity of AIAE guarantees that the adversary cannot forge a valid ciphertext even if it knows errors on the secret key. This leads to the robustness of rrTFE.

7 Instantiation of Our TFE Schemes

This section explains how to instantiate our generic constructions of TFE schemes.

Instantiations. For the information-theoretic secure scheme, we require a secure sketch scheme, a universal hash family, and a secret sharing scheme. For them, we can use the average-case secure sketch scheme and the universal hash function given in [21, 22], and Shamir's secret sharing scheme [39], respectively. As a result, the randomness space of the TFE is an additive group $\mathbb{Z}_p$, and the OTP encryption scheme works with addition over $\mathbb{Z}_p$. For a reusable and robust scheme, we require a secure sketch scheme with homomorphism and linearity, a universal hash family, an AIAE scheme, and an OTS scheme. For such a secure sketch scheme and a homomorphic universal hash function family, we can use the secure sketch scheme in [48] based on the linear error-correcting code, the universal hash family provided in [48, Appendix B], respectively. Additionally, we require an AIAE scheme and an OTS scheme. If we can assume the random oracle model, both schemes can be obtained from any authenticated encryption with associated data (AEAD) [38] (e.g., AES-GCM) and any hash function (e.g., SHA-256). (We discuss in detail in the full version of this paper.) In the standard model, we have the DDH-based AIAE scheme [28] and the DL-based OTS scheme [4].

Efficiency. Table 1 shows the efficiency of the instantiations of our TFE schemes. The bit-length of helper data P is proportional to the number of fuzzy data n; this is not a problem in use cases where n is small.

References

1. Abdalla, M., Chevassut, O., Fouque, P.A., Pointcheval, D.: A simple threshold authenticated key exchange from short secrets. In: Roy, B.K. (ed.) ASI-

Table 1: Efficiency comparison of the instantiations of InfTFE and rrTFE. $|\mathsf{H}|$ denotes the bit-lengths of the description of the universal hash function. $|s|$ denotes the bit-length of the secure sketch. $|\mathbb{G}|$ denotes the bit-length of an element in a group $\mathbb{G}$ with order p. $|\mathbb{Z}_p|$ denotes the bit-length of an element in $\mathbb{Z}_p$. StdM and ROM stand for standard model and random oracle model, respectively.

Scheme	$\lvert\mathsf{pp}\rvert$	$\lvert P\rvert$	Assumptions	Model
InfTFE	$\lvert\mathsf{H}\rvert$	$n(\lvert s\rvert + \lvert\mathbb{Z}_p\rvert)$	None	Std
rrTFE (StdM)	$O(\lambda)$	$n(\lvert s\rvert + 10\lambda + 2) + \lambda + 3\lvert\mathbb{G}\rvert + \lvert\mathbb{Z}_p\rvert$	DDH (over $\mathbb{QR}$), DL	Std
rrTFE (ROM)	$\lvert\mathsf{H}\rvert$	$n(\lvert s\rvert + 2\lambda) + \lambda + 3\lvert\mathbb{G}\rvert + \lvert\mathbb{Z}_p\rvert$	DL	RO

ACRYPT 2005. LNCS, vol. 3788, pp. 566–584. Springer, Berlin, Heidelberg (Dec 2005). https://doi.org/10.1007/11593447_31

2. Agrawal, S., Miao, P., Mohassel, P., Mukherjee, P.: PASTA: PASsword-based threshold authentication. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018. pp. 2042–2059. ACM Press (Oct 2018). https://doi.org/10.1145/3243734.3243839
3. 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: Kim, J., Ahn, G.J., Kim, S., Kim, Y., López, J., Kim, T. (eds.) ASIACCS 18. pp. 673–684. ACM Press (Apr 2018). https://doi.org/10.1145/3196494.3196530
4. Bellare, M., Shoup, S.: Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 201–216. Springer, Berlin, Heidelberg (Apr 2007). https://doi.org/10.1007/978-3-540-71677-8_14
5. Blanton, M., Gasti, P.: Secure and efficient protocols for iris and fingerprint identification. In: Atluri, V., Díaz, C. (eds.) ESORICS 2011. LNCS, vol. 6879, pp. 190–209. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23822-2_11
6. Boyen, X.: Reusable cryptographic fuzzy extractors. In: Atluri, V., Pfitzmann, B., McDaniel, P. (eds.) ACM CCS 2004. pp. 82–91. ACM Press (Oct 2004). https://doi.org/10.1145/1030083.1030096
7. Boyen, X., Dodis, Y., Katz, J., Ostrovsky, R., Smith, A.: Secure remote authentication using biometric data. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 147–163. Springer, Berlin, Heidelberg (May 2005). https://doi.org/10.1007/11426639_9
8. Brandão, L.T.A.N., Peralta, R.: Nist first call for multi-party threshold schemes (2025). https://doi.org/https://doi.org/10.6028/NIST.IR.8214C.2pd
9. Brost, J., Egger, C., Lai, R.W.F., Schmid, F., Schröder, D., Zoppelt, M.: Threshold password-hardened encryption services. In: Ligatti, J., Ou, X., Katz, J., Vigna, G. (eds.) ACM CCS 2020. pp. 409–424. ACM Press (Nov 2020). https://doi.org/10.1145/3372297.3417266
10. Canetti, R., Fuller, B., Paneth, O., Reyzin, L., Smith, A.D.: Reusable fuzzy extractors for low-entropy distributions. In: Fischlin, M., Coron, J.S. (eds.) EURO-

CRYPT 2016, Part I. LNCS, vol. 9665, pp. 117–146. Springer, Berlin, Heidelberg (May 2016). https://doi.org/10.1007/978-3-662-49890-3_5
11. Canetti, R., Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Adaptive security for threshold cryptosystems. In: Wiener, M.J. (ed.) CRYPTO'99. LNCS, vol. 1666, pp. 98–115. Springer, Berlin, Heidelberg (Aug 1999). https://doi.org/10.1007/3-540-48405-1_7
12. Coinbase: What is Multi-Signature (Multi-Sig)? https://www.coinbase.com/learn/wallet/what-is-a-multi-signature-multi-sig-wallet (2026), accessed: April 13, 2026
13. Connolly, D., Komlo, C., Goldberg, I., Wood, C.A.: The Flexible Round-Optimized Schnorr Threshold (FROST) Protocol for Two-Round Schnorr Signatures. RFC 9591 (Jun 2024). https://doi.org/10.17487/RFC9591, https://www.rfc-editor.org/info/rfc9591
14. Cramer, R., Dodis, Y., Fehr, S., Padró, C., Wichs, D.: Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 471–488. Springer, Berlin, Heidelberg (Apr 2008). https://doi.org/10.1007/978-3-540-78967-3_27
15. Crites, E.C., Katz, J., Komlo, C., Tessaro, S., Zhu, C.: On the adaptive security of FROST. In: Kalai, Y.T., Kamara, S.F. (eds.) CRYPTO 2025, Part VI. LNCS, vol. 16005, pp. 480–511. Springer, Cham (Aug 2025). https://doi.org/10.1007/978-3-032-01887-8_16
16. Crites, E.C., Komlo, C., Maller, M.: Fully adaptive Schnorr threshold signatures. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part I. LNCS, vol. 14081, pp. 678–709. Springer, Cham (Aug 2023). https://doi.org/10.1007/978-3-031-38557-5_22
17. Damgård, I., Dupont, K.: Efficient threshold RSA signatures with general moduli and no extra assumptions. In: Vaudenay, S. (ed.) PKC 2005. LNCS, vol. 3386, pp. 346–361. Springer, Berlin, Heidelberg (Jan 2005). https://doi.org/10.1007/978-3-540-30580-4_24
18. De Oliveira Nunes, I., Rindal, P., Shirvanian, M.: Oblivious extractors and improved security in biometric-based authentication systems. In: Tsudik, G., Conti, M., Liang, K., Smaragdakis, G. (eds.) ESORICS 2023, Part I. LNCS, vol. 14344, pp. 290–312. Springer, Cham (Sep 2023). https://doi.org/10.1007/978-3-031-50594-2_15
19. Dodis, Y., Kanukurthi, B., Katz, J., Reyzin, L., Smith, A.: Robust fuzzy extractors and authenticated key agreement from close secrets. IEEE Transactions on Information Theory **58**(9), 6207–6222 (2012)
20. Dodis, Y., Katz, J., Reyzin, L., Smith, A.: Robust fuzzy extractors and authenticated key agreement from close secrets. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 232–250. Springer, Berlin, Heidelberg (Aug 2006). https://doi.org/10.1007/11818175_14
21. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM journal on computing **38**(1), 97–139 (2008)
22. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Berlin, Heidelberg (May 2004). https://doi.org/10.1007/978-3-540-24676-3_31
23. Dupont, P.A., Hesse, J., Pointcheval, D., Reyzin, L., Yakoubov, S.: Fuzzy password-authenticated key exchange. In: Nielsen, J.B., Rijmen, V. (eds.) EURO-

CRYPT 2018, Part III. LNCS, vol. 10822, pp. 393–424. Springer, Cham (Apr / May 2018). https://doi.org/10.1007/978-3-319-78372-7_13
24. Evans, D., Huang, Y., Katz, J., Malka, L.: Efficient privacy-preserving biometric identification. In: Network and Distributed System Security Symposium (2011)
25. Fuller, B., Meng, X., Reyzin, L.: Computational fuzzy extractors. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 174–193. Springer, Berlin, Heidelberg (Dec 2013). https://doi.org/10.1007/978-3-642-42033-7_10
26. Fuller, B., Reyzin, L., Smith, A.D.: When are fuzzy extractors possible? In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 277–306. Springer, Berlin, Heidelberg (Dec 2016). https://doi.org/10.1007/978-3-662-53887-6_10
27. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Robust threshold DSS signatures. In: Maurer, U.M. (ed.) EUROCRYPT'96. LNCS, vol. 1070, pp. 354–371. Springer, Berlin, Heidelberg (May 1996). https://doi.org/10.1007/3-540-68339-9_31
28. Han, S., Liu, S., Lyu, L.: Efficient KDM-CCA secure public-key encryption for polynomial functions. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 307–338. Springer, Berlin, Heidelberg (Dec 2016). https://doi.org/10.1007/978-3-662-53890-6_11
29. International Organization for Standardization: ISO 22387:2022 security and resilience — authenticity, integrity and trust for products and documents — validation procedures for the application of artefact metrics. International Standard (Dec 2022), https://www.iso.org/standard/80717.html, edition 1
30. Kanukurthi, B., Reyzin, L.: An improved robust fuzzy extractor. In: Ostrovsky, R., De Prisco, R., Visconti, I. (eds.) SCN 08. LNCS, vol. 5229, pp. 156–171. Springer, Berlin, Heidelberg (Sep 2008). https://doi.org/10.1007/978-3-540-85855-3_11
31. 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: Vigna, G., Shi, E. (eds.) ACM CCS 2021. pp. 2046–2065. ACM Press (Nov 2021). https://doi.org/10.1145/3460120.3484586
32. Komlo, C., Goldberg, I.: FROST: Flexible round-optimized Schnorr threshold signatures. In: Dunkelman, O., Jacobson, Jr., M.J., O'Flynn, C. (eds.) SAC 2020. LNCS, vol. 12804, pp. 34–65. Springer, Cham (Oct 2020). https://doi.org/10.1007/978-3-030-81652-0_2
33. Kuznetsov, A., Zakharov, D., Frontoni, E., Romeo, L., Rosati, R.: Deep learning based fuzzy extractor for generating strong keys from biometric face images. In: 2022 IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T). pp. 421–426 (2022). https://doi.org/10.1109/PICST57299.2022.10238643
34. Lim, D., Lee, J., Gassend, B., Suh, G., van Dijk, M., Devadas, S.: Extracting secret keys from integrated circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems **13**(10), 1200–1205 (2005). https://doi.org/10.1109/TVLSI.2005.859470
35. Lucks, S.: Ciphers secure against related-key attacks. In: Roy, B.K., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 359–370. Springer, Berlin, Heidelberg (Feb 2004). https://doi.org/10.1007/978-3-540-25937-4_23
36. Ma, J., Qi, B., Lv, K.: Threshold reusable fuzzy extractor and an application to joint access control via biometric information. Information Science **579**, 525–540 (2021)

37. Matsumoto, T., Hoga, M., Ohyagi, Y., Ishikawa, M., Naruse, M., Hanaki, K., Suzuki, R., Sekiguchi, D., Tate, N., Ohtsu, M.: Nano-artifact metrics based on random collapse of resist. Scientific Reports **4**(1), 6142 (Aug 2014). https://doi.org/10.1038/srep06142, https://doi.org/10.1038/srep06142
38. Rogaway, P.: Authenticated-encryption with associated-data. In: Atluri, V. (ed.) ACM CCS 2002. pp. 98–107. ACM Press (Nov 2002). https://doi.org/10.1145/586110.586125
39. Shamir, A.: How to share a secret. Communications of the Association for Computing Machinery **22**(11), 612–613 (Nov 1979). https://doi.org/10.1145/359168.359176
40. Shoup, V., Gennaro, R.: Securing threshold cryptosystems against chosen ciphertext attack. In: Nyberg, K. (ed.) EUROCRYPT'98. LNCS, vol. 1403, pp. 1–16. Springer, Berlin, Heidelberg (May / Jun 1998). https://doi.org/10.1007/BFb0054113
41. Shukla, A., Demarest, L., Fuller, B., Ahmad, S., Manicke, C., Russell, A., Chen, S.: Fuzzy extractors are practical: Cryptographic strength key derivation from the iris. In: ACM CCS 2025 (2025). https://doi.org/10.1145/3719027.3765098
42. Shukla, A., Demarest, L., Fuller, B., Ahmad, S., Manicke, C., Russell, A., Chen, S.: Fuzzy extractors are practical: Cryptographic strength key derivation from the iris. In: Huang, C.Y., Chen, J.C., Shieh, S.P., Lie, D., Cortier, V. (eds.) ACM CCS 2025. pp. 3605–3619. ACM Press (Oct 2025). https://doi.org/10.1145/3719027.3765098
43. Skoric, B., Tuyls, P.: An efficient fuzzy extractor for limited noise. Cryptology ePrint Archive, Report 2009/030 (2009), https://eprint.iacr.org/2009/030
44. Skoric, B., Tuyls, P., Ophey, W.: Robust key extraction from physical uncloneable functions. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 407–422. Springer, Berlin, Heidelberg (Jun 2005). https://doi.org/10.1007/11496137_28
45. Tong, V.V.T., Sibert, H., Lecœur, J., Girault, M.: Biometric fuzzy extractors made practical: A proposal based on fingercodes. In: Lee, S.W., Li, S.Z. (eds.) Advances in Biometrics. pp. 604–613. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
46. Wen, Y., Liu, S.: Reusable fuzzy extractor from LWE. In: Susilo, W., Yang, G. (eds.) ACISP 18. LNCS, vol. 10946, pp. 13–27. Springer, Cham (Jul 2018). https://doi.org/10.1007/978-3-319-93638-3_2
47. Wen, Y., Liu, S.: Robustly reusable fuzzy extractor from standard assumptions. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part III. LNCS, vol. 11274, pp. 459–489. Springer, Cham (Dec 2018). https://doi.org/10.1007/978-3-030-03332-3_17
48. Wen, Y., Liu, S., Gu, D.: Generic constructions of robustly reusable fuzzy extractor. In: Lin, D., Sako, K. (eds.) PKC 2019, Part II. LNCS, vol. 11443, pp. 349–378. Springer, Cham (Apr 2019). https://doi.org/10.1007/978-3-030-17259-6_12
49. Wen, Y., Liu, S., Han, S.: Reusable fuzzy extractor from the decisional Diffie-Hellman assumption. DCC **86**(11), 2495–2512 (2018). https://doi.org/10.1007/s10623-018-0459-4
50. Wong, H.W.H., Ma, J.P.K., Yin, H.H.F., Chow, S.S.M.: Real threshold ECDSA. In: NDSS 2023. The Internet Society (Feb 2023)
51. Zcash Foundation: Frost (2026). https://doi.org/https://zfnd.org/frost/