IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

DOI:10.1587/transfun.2024CIP0012

Publicized:2024/10/08

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Sample Recoverable Fuzzy Extractors*

Wataru NAKAMURA[†a)], *Nonmember and* **Kenta TAKAHASHI**[†b)], *Member*

SUMMARY To realize online biometric authentication systems with both of protection and utilization of biometric data, we propose a novel primitive called "Sample Recoverable Fuzzy Extractors (SRFEs)." Conventionally, Biometric Template Protection (BTP) is studied as an approach for preventing biometric data from leakage. An important requirement of BTP is that it is difficult to recover biometric data from the stored data, which is called irreversibility, and fuzzy extractors are known as one of promising primitives for realizing BTP. On the other hand, in some cases, it is desired that the system can utilize biometric samples such as images having captured during past enrollment and authentication processes. For example, when the authentication accuracy of a specific user is low, samples of past processes are helpful clue for investigation of a cause. Also, they can be used for multi-sample fusion to improve accuracy in a biometric template update, and for post verification of past processes. To enable utilization of past biometric samples for such various situations while protecting the biometric data, we define a SRFE as a primitive satisfying the following two properties: (i) It can recover the secret key along with samples of past enrollment and authentication processes from the stored data and a feature which can succeeds in the authentication. (ii) It is computationally difficult to recover the secret key from the stored data. We give a construction based on a fuzzy extractor and a symmetric encryption scheme satisfying a kind of key dependent message security. By using a SRFE, we realize a protocol of an online biometric authentication system which satisfies irreversibility while the past biometric samples can be recovered from the stored data with a help of the genuine user.

key words: *Biometric template protection, fuzzy extractors, utilization of past biometric samples, authenticated sample recoverability.*

1. Introduction

1.1 Background on Online User Authentication

In a current digital society with wide-spreading online services such as banking, shopping, file storage, and social media, online user authentication technology is essential to avoid unauthorized accesses of the system. User authentication methods are often classified into "what you know," "what you have," and "what you are," based on the type of secret information used for authentication [2][3]. For security, it is crucial to prevent the secret information leakage. Also, it is required to be secure against online attacks such as a replay attack and a real-time phishing. In addition to security, user convenience is important because inconvenience hinders widespread use.

Online user authentication methods based on "what you know" such as password-based schemes are widely used, but it is difficult for them to achieve both security and convenience. If users use a password easy to remember, it becomes easier for adversaries to guess. On the other hand, requiring users to remember complex knowledge reduces convenience.

Methods based on "what you have" can be realized by devices (e.g., smartphones, computers, IC cards, and hardware tokens) storing a secret key. In particular, digital signature-based schemes such as FIDO[†] [4] can protect online attacks such as a replay attack and a real-time phishing, and make it difficult for adversaries to guess the secret key even if they obtain the corresponding verification key stored in the server and signatures sent to the server. In addition, in order to lower the risk of unauthorized use, the device can perform additional local authentication, e.g., local biometric authentication as FIDO system does. However, schemes based on "what you have" requires users to bring a device storing the secret key to authenticate themselves, which reduces convenience. Furthermore, if users lose the device, they will not be able to be authenticated by the system.

A hopeful alternative is online user authentication schemes based on "what you are," i.e., online biometric authentication, used in commercial and governmental services [5][6]. Users can authenticate themselves using their own biometric traits such as face, fingerprint, and iris, without remembering complex passwords or bringing a device storing a secret key.

Online biometric authentication systems can simply be realized as follows: the server stores biometric data captured during an enrollment process, and compares it to biometric data captured during an authentication process. However, this scheme has the risk of biometric data leakage from the data stored in the server. For example, due to a vulnerability in a biometric ID system "Aadhaar," anybody had access to biometric data of more than 1 billion Indian citizens [7]. Biometric data leakage leads to privacy invasion and increases impersonation risk. Furthermore, unlike passwords, it is difficult for users to change their biometric data if compromised.

To protect biometric information from leakage, *Biometric Template Protection (BTP)* [8][9] has been studied and standardized. In BTP, biometric data is transformed into a protected biometric template so that it is difficult to recover

[†]The authors are with Hitachi, Ltd., Japan.

*This paper was presented in part at 2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2023) [1].

a) E-mail: wataru.nakamura.va@hitachi.com
b) E-mail: kenta.takahashi.bw@hitachi.com

[†]FIDO is a trademark of FIDO Alliance.

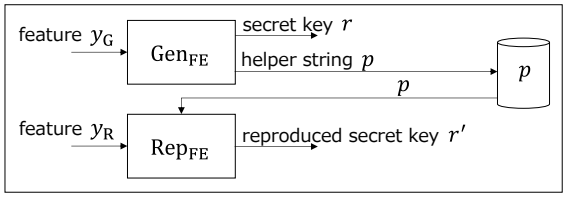


Fig. 1 Processes and Data Flow of FEs

the original biometric data from the template. This requirement is called *irreversibility*[†]. One of promising primitives for realizing BTP is *Fuzzy Extractors (FEs)* [10], which derive a secret key from biometric data. As Fig. 1, the generation algorithm $\mathrm{Gen_{FE}}$ of a FE generates a pair (r, p) of a secret key r and a helper string p from a fuzzy biometric feature y_G. The reproduction process $\mathrm{Rep_{FE}}$ reproduces secret key r' from the helper string p and a feature y_R. The correct secret key r is reproduced if features y_G and y_R are sufficiently close, and it is difficult to guess y_G or r from the helper string p alone. By using the secret key r for generating a digital signature, secure online authentication can be realized [11].

1.2 Motivation for Irreversibility with Authenticated Sample Recoverability (IASR)

While irreversibility is required to lower the risk of biometric data leakage, it is desired in some cases that the system can utilize biometric samples such as images of biometric traits having captured during past enrollment and authentication processes. Examples of such cases include the following (Case A)–(Case C).

(Case A) *Investigation of a cause of low authentication accuracy.* The biometric sample acquired from an individual is susceptible to changes due to improper interaction with the sensor, modifications in sensor characteristics, variations in environmental factors, and temporary alterations in the biometric trait itself [12]. Because of these changes of the obtained samples, it is possible that some uses are frequently rejected from the system. When a user is relatively frequently rejected, samples having captured during past enrollment and authentication processes can be a powerful clue for investigating the cause in more detail, which will lead to fewer false rejects. For example, if the user notices by checking the samples that they tend to be acquired under too bright environment (or in tilting postures), the user can be careful about the lighting conditions (or postures). In addition, once the common causes for such users are identified, the system can be modified to reduce them. For example, if it turns out that users tend to input their biometric traits into the scanning device in inappropriate postures, the system can be modified so that the correct input posture is displayed on the screen.

(Case B) *Multi-sample fusion in biometric template update.* Enrolled biometric templates have to be updated for various reasons. For example, when a biometric authentication system upgrades algorithms such as feature extraction, it has to update enrolled template to reflect the upgrade. Also, a user's template has to be updated when the sample captured for enrollment is unrepresentative [12] or when his/her biometric trait changes due to aging [13]. To improve authentication accuracy, an effective technique is multi-sample fusion [12][14][15][16], which generates a template using multiple samples. If this technique is used during template update process, the system can enroll new templates that achieve higher accuracy. However, scanning samples many times during the update process is a burden for users. If samples during past processes can be utilized for multi-sample fusion, the system can improve authentication accuracy without placing the burden on users.

(Case C) *Post verification of past processes.* When some problem is found on a past process, the sample having captured during the process can be helpful for post verification of the process. For example, when a user notices a past authentication history that he/she does not remember[††], it is possible that a *False Acceptance (FA)* occurred, i.e., the system accepted another user's sample, or a *Presentation Attack (PA)* [17] occurred, i.e., the process was done by an attacker inputting a fake biometric sample. If FA or PA occurred, the user will want to show that he/she did not perform the authentication because the unauthorized authentication usually leads to some damage such as unauthorized money transfer or payment. The system administrator will also want to know whether FA or PA occurred or not because some measures need to be taken if it actually occurred. On the other hand, it is possible that the process was performed by the user, but he/she simply forgot it or makes a malicious claim. If the sample having captured during the process can be utilized, it can help to investigate in detail after the fact whether the sample is from another user, fake, or from the genuine user. For example, to investigate whether a FA occurred, one can match the sample during the process and the sample during the enrollment using matching schemes different from the one employed by the system[†††]. Also, to

[†]Although BTP has other requirements, we focus on irreversibility because it is the most important when we consider lowering the risk of biometric data leakage.

[††]In services such as payment at store or bank transfer using biometric authentication, the service provider may not notice an unauthorized authentication at the time of the process and notices it when a user notices and points it out later.

[†††]The matching scheme employed by the system is constrained by the system requirements. For example, if irreversibility is required, only schemes satisfying it can be used. On the other hand, in post verification, one can use any schemes including any feature extraction schemes. Therefore, even if the system accepted another user's sample during the authentication process, there is a chance to notice that the input sample was from another user.

investigate whether a PA occurred, one can check the sample using up-to-date Presentation Attack Detection (PAD) schemes[†].

A trivial scheme to enable the utilization of past samples is storing them itself, but this scheme cannot protect biometric data. To protect biometric data, one might come up with the scheme of storing them in encrypted form, and the key for decrypting them. However, this scheme also does not satisfy irreversibility because samples can be recovered from the stored data. To realize both of irreversibility and utilization of biometric data, we aim at constructing online biometric authentication which satisfies irreversibility while past samples can be recovered *with a help of the genuine user*, i.e., with a sample captured from the genuine user again. To make this concrete, we introduce *Irreversibility with Authenticated Sample Recoverability (IASR)*[††] for online biometric authentication systems as the property of satisfying the following two:

(a) *Irreversibility* : It is difficult to recover a sample or a feature having used for past enrollment or authentication processes from the stored data.
(b) *Authenticated Sample Recoverability (ASR)* : From the stored data and a biometric sample which can succeed in the authentication[†††], the system can recover samples having captured during past enrollment and authentication processes.

The goal of this paper is to realize an online biometric authentication system with IASR.

1.3 Limitation of Conventional FEs on IASR

Some conventional FEs can realize online biometric authentication systems satisfying *Irreversibility with Authenticated Feature Recoverability (IAFR)*, a weaker property than IASR. We define IAFR as the property of satisfying irreversibility and the following AFR:

(b') *Authenticated Feature Recoverability (AFR)* : From the stored data and a biometric sample which can succeed in the authentication, the system can recover *features having extracted* during past enrollment and authentication processes.

Indeed, in FEs constructed with a Secure Sketch (SS), e.g., Dodis et al.'s [10], the helper string includes a sketch of the feature y_G for generation. The feature y_G can be recovered from the sketch and a feature y' sufficiently close to y_G, while y_G has sufficient min-entropy given the sketch. Therefore, authentication systems based on FEs with a SS can recover the feature y_G having extracted during the enrollment from the stored data and a feature y' close to y_G. Features during past authentication processes can be also recovered if the system runs the generation algorithm additionally during each authentication process. Some FEs have a property called reusability, which ensures security even when multiple helper strings are stored. Therefore, by using a reusable FE with a SS, the system satisfies IAFR.

However, for (Case A)–(Case C) described above, IASR is needed rather than IAFR. Features input for FEs are elements of a metric space such as a Hamming space, but in situations such as (Case A)–(Case C), raw biometric samples such as images of biometric traits are needed. The samples are a powerful clue for the investigation in (Case A), required for multi-sample fusion in (Case B), and helpful information for the post verification in (Case C).

1.4 Our Contribution

To realize online biometric authentication systems with IASR, we propose *Sample Recoverable Fuzzy Extractors (SRFEs)* as a primitive satisfying the following three properties:

(Property 1) From the stored data and a sample x_R close to one x_G for the generation process, the secret key can be correctly reproduced, where the closeness is measured by the closeness of the extracted features.
(Property 2) From the stored data and a feature y_{SR} extracted to a sample x_{SR} close to one x_G for the generation process, the sample recovery process can correctly recovers the samples (x_G, x_R) for generation and successful reproduction processes.
(Property 3) It is difficult for any Probabilistic Polynomial Time (PPT) adversary who obtains the stored data to distinguish the secret key r and a uniform random number.

We explain this definition in more detail using Fig. 2. We define SRFEs as a tuple $(\mathrm{Gen}_{\mathrm{SRFE}}, \mathrm{Rep}_{\mathrm{SRFE}}, \mathrm{SRec}_{\mathrm{SRFE}})$ of generation, reproduction, and sample recovery algorithms. Although generation $\mathrm{Gen}_{\mathrm{FE}}$ and reproduction $\mathrm{Rep}_{\mathrm{FE}}$ of conventional FEs takes a feature as an input, we input a sample, not a feature, to $\mathrm{Gen}_{\mathrm{SRFE}}$ and $\mathrm{Rep}_{\mathrm{SRFE}}$ in order to treat recovery of the samples having been input to them. $\mathrm{Gen}_{\mathrm{SRFE}}$

[†]Even if the system employs a PAD scheme, it is not always up-to-date because PAD schemes have been actively studied. Therefore, even if the fake sample did not detect the PA, there is a chance to notice it by up-to-date PAD schemes.

[††]In the conference version [1], the property of satisfying (a) and (b) is called ASR. In this paper, we define IASR and ASR as above because we think that this definition of the terms more appropriately describes the properties.

[†††]In (Case B), the system is required to check whether the user trying to update a template is the genuine user. By performing the check through the biometric authentication, a biometric sample succeeding in the authentication can be obtained and used for ASR. On the other hand, in (Case A) and (Case C), user's cooperation is needed to obtain such a sample. We believe that users will tend to cooperate in these cases for the following reasons. In (Case A), the investigation leads to improvement of their convenience in authentication. We note that for users relatively frequently rejected, e.g., with probability 10%, the system can obtain a sample succeeding in authentication by simply repeating the scanning from them a few times. Also, if at least some users cooperate, the identified causes will enable system modifications, which will also reduce false rejects for other users. In (Case C), the user will want to show that he/she did not perform the authentication as described above.

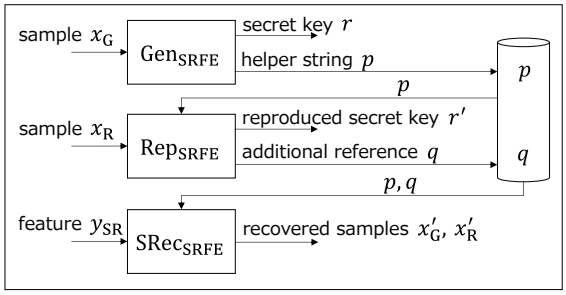


Fig. 2 Processes and Data Flow of SRFEs

generates a secret key r and a helper string p from a sample x_G, and p is stored. (Property 1) means that if x_R and x_G are close, then $\mathrm{Rep}_{\mathrm{SRFE}}$ correctly reproduce r from x_R and the stored p. Unlike the reproduction of conventional FEs, $\mathrm{Rep}_{\mathrm{SRFE}}$ additionally outputs data q, which we call *an additional reference*, and q is also stored to be used for the sample recovery. (Property 2) means that if samples x_{SR} and x_G are sufficiently close, then $\mathrm{SRec}_{\mathrm{SRFE}}$ correctly recovers (x_G, x_R) from a feature y_{SR}[†] extracted from x_{SR} and the stored (p, q), provided that $\mathrm{Rep}_{\mathrm{SRFE}}$ using x_R is successful. (Property 3) means that it is difficult for any PPT adversary who obtains (p, q) to distinguish the secret key r and a uniform random number[††].

Our technical contributions are threefold:

(i) *Formal definition for SRFEs*. We give a formal definition of SRFEs so that they satisfy the above (Property 1)–(Property 3), along with a biometric data setting, which specifies how fuzzy biometric data are generated.

(ii) *Construction of SRFEs*. We propose a generic construction of a SRFE based on a FE and a symmetric encryption scheme satisfying a kind of key dependent message security. Also, we give a specific construction of a SRFE.

(iii) *Online Biometric Authentication with IASR*. We construct a protocol of an online biometric authentication system using a SRFE, and show that the authentication system satisfies IASR. Also, we analyze the performance of the proposed system.

1.5 Organization

The rest of this paper is organized as follows. In Sect. 1.6, we explain the relation of this paper and an earlier version. In Sect. 2, we review basic notation and standard definitions. In Sect. 3, we formalize a biometric data setting and SRFEs. In Sect. 4, we propose a construction of SRFEs. In Sect. 5, we apply SRFEs to online user authentication. In Sect. 6, we describe conclusion.

1.6 Relation to the Conference Version

This work is an extended version of a conference paper [1]. In addition to editorial changes, this paper mainly differs from [1] in the following aspects. In [1], we introduced Sample Recoverable Biometric Signatures (SRBSs) (which we describe in Sect. 5.1 in this paper), and gave a generic construction and an instantiation. We can observe that the construction in [1] includes a scheme satisfying the above (Property 1)–(Property 3). Based on this, in this paper we introduce a new primitive SRFEs. We give a generic construction and an instantiation of a SRFE by extracting core parts of the construction in [1]. Also, we add an analysis on the proposed system.

2. Preliminaries

In this section, we review basic notations and definitions of primitives.

2.1 Basic Notation

Let $\mathbb{N}$ denote the set of all positive integers, and let $\mathbb{R}$ and $\mathbb{R}_{>0}$ denote the sets of real numbers and positive real numbers, respectively. Throughout the paper, λ denotes the security parameter. For a set $\mathcal{U}$, $u \leftarrow \mathcal{U}$ denotes choosing uniform randomly an element u from $\mathcal{U}$. For a probability distribution U, $u \leftarrow \mathrm{U}$ denotes choosing an element u according to U. A function $f : \mathbb{N} \to [0, 1]$ is said to be *negligible* if for all polynomials g and all sufficiently large k it holds $f(k) < 1/g(k)$. For random variables X and Y, the *min-entropy* $\mathbf{H}_\infty(X)$ of X is defined by $\mathbf{H}_\infty(X) := -\log_2(\max_x \Pr[X = x])$, and the average min-entropy $\tilde{\mathbf{H}}_\infty(X \mid Y)$ of X given Y is defined by $\tilde{\mathbf{H}}_\infty(X \mid Y) := -\log_2 \mathbf{E}_{y \leftarrow \mathrm{Y}}\big[\max_x \Pr[X = x \mid Y = y]\big]$, where Y denotes the probability distribution of Y. The *statistical distance* $\mathbf{SD}(X, Y)$ between random variables X and Y is defined by $\mathbf{SD}(X, Y) := (1/2)\sum_z |\Pr[X = z] - \Pr[Y = z]|$. For $\ell \in \mathbb{N}$, U_ℓ denotes a random variable uniformly distributed on $\{0, 1\}^\ell$. For a positive function f, we write $f(k) = \Theta(k)$ if there exist c_1, c_2, and k_0 such that for any k larger than k_0 it holds that $c_1 < f(k)/k < c_2$.

2.2 Fuzzy Extractors (FEs)

We review the definition of a FE introduced by Dodis et al. [10].

Definition 1. A $(\mathcal{Y}, \mu, \ell, t, \epsilon)$-FE Σ_{FE} is defined by the following algorithms $(\mathrm{Gen}_{\mathrm{FE}}, \mathrm{Rep}_{\mathrm{FE}})$.

- $\mathrm{Gen}_{\mathrm{FE}}(y) \to (r, p)$: The generation algorithm takes an element $y \in \mathcal{Y}$ as an input, and outputs an extracted string $r \in \{0, 1\}^\ell$ and a public string $p \in \{0, 1\}^*$. We require that for any random variable Y on $\mathcal{Y}$ of min-entropy μ, if $(R, P) \leftarrow \mathrm{Gen}_{\mathrm{FE}}(Y)$, then $\mathbf{SD}((R, P), (U_\ell, P)) \le \epsilon$.

[†]We define that $\mathrm{SRec}_{\mathrm{SRFE}}$ takes a feature y_{SR}, not a sample x_{SR}, as an input. This is because by adopting this definition we can show that our protocol of online biometric authentication using a SRFE in Sect. 5.2 satisfies IASR, as described in Sect. 5.2.

[††]In more detail, we treat security when a helper string and multiple additional references are stored.

- $\mathrm{Rep}_{\mathrm{FE}}(y', p) \to r'$: The reproduction algorithm $\mathrm{Rep}_{\mathrm{FE}}$ takes an element $y' \in \mathcal{Y}$ and a bit string $p \in \{0,1\}^*$ as inputs, and outputs a string r'. We require that for any $y, y' \in \mathcal{Y}$ satisfying $\mathtt{dist}(y, y') \leq t$ and (r, p) generated by $(r, p) \leftarrow \mathrm{Gen}_{\mathrm{FE}}(y)$, it holds that $\mathrm{Rep}_{\mathrm{FE}}(y', p) = r$.

2.3 Symmetric Encryption Schemes

A symmetric encryption scheme Σ_{SE} is defined by the following algorithms $(\mathrm{Gen}_{\mathrm{SE}}, \mathrm{Enc}_{\mathrm{SE}}, \mathrm{Dec}_{\mathrm{SE}})$.

- $\mathrm{Gen}_{\mathrm{SE}}(1^\lambda) \to sk_{\mathrm{SE}}$: The generation algorithm takes the security parameter 1^λ as an input, and outputs a symmetric key sk_{SE}.
- $\mathrm{Enc}_{\mathrm{SE}}(sk_{\mathrm{SE}}, m) \to c$: The encryption algorithm takes a symmetric key sk_{SE} and a message $m \in \mathcal{M}$ as inputs, and outputs a ciphertext c.
- $\mathrm{Dec}_{\mathrm{SE}}(sk_{\mathrm{SE}}, c) \to m'$ or $\perp$: The decryption algorithm takes a symmetric key sk_{SE} and a string c as inputs, and outputs a decrypted message $m' \in \mathcal{M}$ or $\perp$. We require that for any $m \in \mathcal{M}$, $\lambda \in \mathbb{N}$, and (sk_{SE}, c) generated by $sk_{\mathrm{SE}} \leftarrow \mathrm{Gen}_{\mathrm{SE}}(1^\lambda)$ and $c \leftarrow \mathrm{Enc}(sk_{\mathrm{SE}}, m)$, it holds that $\mathrm{Dec}(sk_{,\mathrm{SE}}, c) = m$.

In Sect. 4.2, we define a specific security property of Σ_{SE} in order to use it as a building block of SRFEs. The property can be considered as a kind of *Key Dependent Message (KDM) security* [18], which is on encryption of messages depending on the secret key[†]. Specifically, KDM security in the single key setting[††] is defined as follows. Consider the following KDM experiment $\mathrm{Exp}^{\mathrm{KDM}}_{\Sigma_{\mathrm{SE}},\mathcal{F},\mathcal{A}}(\lambda)$ for a symmetric encryption scheme $\Sigma_{\mathrm{SE}} = (\mathrm{Gen}_{\mathrm{SE}}, \mathrm{Enc}_{\mathrm{SE}}, \mathrm{Dec}_{\mathrm{SE}})$, a class $\mathcal{F}$ of functions, and an adversary $\mathcal{A}$:

$$\mathrm{Exp}^{\mathrm{KDM}}_{\Sigma_{\mathrm{SE}},\mathcal{F},\mathcal{A}}(\lambda) : [\ b \leftarrow \{0,1\};\ sk_{\mathrm{SE}} \leftarrow \mathrm{Gen}_{\mathrm{SE}}(1^\lambda);\ b' \leftarrow \mathcal{A}^{O_{\mathrm{Enc}_{\mathrm{SE}}}}();\ \text{return } b'\ ],$$

where $O_{\mathrm{Enc}_{\mathrm{SE}}}$ is a encryption oracle that takes a function $f \in \mathcal{F}$ as an input and operates as follows: $[\ m_1 := f(sk_{\mathrm{SE}});\ m_0 := 0^{|f(sk_{\mathrm{SE}})|};\ c \leftarrow \mathrm{Enc}_{\mathrm{SE}}(sk_{\mathrm{SE}}, c_b);\ \text{return } c\ ]$. We say that a symmetric encryption scheme Σ_{SE} is $\mathcal{F}$-KDM secure if for any PPT adversary $\mathcal{A}$,

$$\mathrm{Adv}^{\mathrm{KDM}}_{\Sigma_{\mathrm{SE}},\mathcal{F},\mathcal{A}}(\lambda) := \big|\Pr[\mathrm{Exp}^{\mathrm{KDM}}_{\Sigma_{\mathrm{SE}},\mathcal{F},\mathcal{A}}(\lambda) = 1 \mid b = 1] - \Pr[\mathrm{Exp}^{\mathrm{KDM}}_{\Sigma_{\mathrm{SE}},\mathcal{F},\mathcal{A}}(\lambda) = 1 \mid b = 0]\big|$$

is negligible.

2.4 Digital Signature Schemes

A digital signature scheme Σ_{DS} is defined by the following algorithms $(\mathrm{Gen}_{\mathrm{DS}}, \mathrm{Sign}_{\mathrm{DS}}, \mathrm{Ver}_{\mathrm{DS}})$.

- $\mathrm{Gen}_{\mathrm{DS}}(1^\lambda) \to (sk_{\mathrm{DS}}, vk_{\mathrm{DS}})$: The generation algorithm takes the security parameter 1^λ as an input, and outputs a signing key sk_{DS} and a verification key vk_{DS}.
- $\mathrm{Sign}_{\mathrm{DS}}(sk_{\mathrm{DS}}, m) \to \sigma_{\mathrm{DS}}$: The signing algorithm takes a signing key sk_{DS} and a message $m \in \mathcal{M}$ as inputs, and outputs a signature σ_{DS}.
- $\mathrm{Ver}_{\mathrm{DS}}(vk_{\mathrm{DS}}, m, \sigma_{\mathrm{DS}}) \to \mathit{result}$: The verification algorithm takes a verification key vk_{DS}, a message $m \in \mathcal{M}$, and a signature σ_{DS} as inputs, and outputs the verification result $\mathit{result} \in \{\top, \perp\}$. We require that for any $\lambda \in \mathbb{N}$, message $m \in \mathcal{M}$, and $(sk_{\mathrm{DS}}, vk_{\mathrm{DS}}, \sigma_{\mathrm{DS}})$ generated by $(sk_{\mathrm{DS}}, vk_{\mathrm{DS}}) \leftarrow \mathrm{Gen}_{\mathrm{DS}}(1^\lambda)$ and $\sigma_{\mathrm{DS}} \leftarrow \mathrm{Sign}_{\mathrm{DS}}(sk_{\mathrm{DS}}, m)$, it holds that $\mathrm{Ver}_{\mathrm{DS}}(vk_{\mathrm{DS}}, m, \sigma_{\mathrm{DS}}) = \top$.

EUF-CMA security for a digital signature scheme Σ_{DS} is defined as follows. Consider the following EUF-CMA experiment $\mathrm{Exp}^{\text{EUF-CMA}}_{\Sigma_{\mathrm{DS}},\mathcal{A}}(\lambda)$ for $\Sigma_{\mathrm{DS}} = (\mathrm{Gen}_{\mathrm{DS}}, \mathrm{Sign}_{\mathrm{DS}}, \mathrm{Ver}_{\mathrm{DS}}, \mathrm{KeyVer}_{\mathrm{DS}})$ and an adversary $\mathcal{A}$:

$$\mathrm{Exp}^{\text{EUF-CMA}}_{\Sigma_{\mathrm{DS}},\mathcal{A}}(\lambda) : [\ (sk_{\mathrm{DS}}, vk_{\mathrm{DS}}) \leftarrow \mathrm{Gen}_{\mathrm{DS}}(1^\lambda);\ Q := \emptyset;\ (m', \sigma'_{\mathrm{DS}}) \leftarrow \mathcal{A}^{O_{\mathrm{Sign}_{\mathrm{DS}}}(\cdot)}(vk_{\mathrm{DS}});\ \text{if } m' \notin Q \wedge \mathrm{Ver}(m', vk_{\mathrm{DS}}, \sigma'_{\mathrm{DS}}) = \top\ \text{then return 1 else return 0 }],$$

where $O_{\mathrm{Sign}_{\mathrm{DS}}}$ is a signing oracle that takes a message $m \in \mathcal{M}$ as an input and operates as follows: $[\ Q := Q \cup \{m\};\ \sigma_{\mathrm{DS}} \leftarrow \mathrm{Sign}_{\mathrm{DS}}(sk_{\mathrm{DS}}, m);\ \text{return } \sigma_{\mathrm{DS}}\ ]$. We say that a digital signature scheme Σ_{DS} is *EUF-CMA secure* if for any PPT adversary $\mathcal{A}$, $\mathrm{Adv}^{\text{EUF-CMA}}_{\Sigma_{\mathrm{DS}},\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{\text{EUF-CMA}}_{\Sigma_{\mathrm{DS}},\mathcal{A}}(\lambda) = 1]$ is negligible.

3. Definitions for SRFEs

In this section, we first define a biometric data setting, which specifies how fuzzy data such as biometric data are generated. Then, we define the syntax and requirements of SRFEs.

3.1 Biometric Data Setting

As described in Sect. 1.4, we consider generation, reproduction, and sample recovery processes for SRFEs. We define a setting of samples and features for these processes.

Let $\mathcal{X}$ be the sample space. Let X be the distribution of a sample for generation. We assume that when a sample x_{G} is captured from a user for generation process, a sample from the user for reproduction and sample recovery is generated according to the probability distribution denoted by $\Delta(x_{\mathrm{G}})$. Also, we assume that a feature generation algorithm Feat is given, and $\mathcal{Y}$ denotes the feature space. That is, a feature $y \in \mathcal{Y}$ is generated from a sample $x \in \mathcal{X}$ by $y \leftarrow \mathrm{Feat}(x)$. Features generated from samples of the genuine user should be close with high probability. This can be formalized with a threshold $t \in \mathbb{R}_{>0}$ and an error parameter $\alpha \in [0,1]$ as follows: $\Pr\{x_{\mathrm{G}} \leftarrow \mathrm{X};\ x' \leftarrow \Delta(x_{\mathrm{G}}) : \mathtt{dist}(\mathrm{Feat}(x_{\mathrm{G}}), \mathrm{Feat}(x')) > t\} \leq \alpha$. We

[†]When $\mathcal{F}$ contains all constant functions, $\mathcal{F}$-KDM security implies IND-CPA [19].

[††]In general, KDM security is considered in the multiple key setting. However, the property we define in Sect. 4.2 is related to KDM security in the single key setting, so we explain the setting here.

call $\mathcal{B} = (\mathcal{X}, \mathrm{X}, \Delta, \mathcal{Y}, \mathrm{Feat}, \mathtt{dist}, t, \alpha)$ satisfying the above conditions a *biometric data setting*.

Remark 1. If the system adopt multi-sample fusion, multiple samples may be required for each algorithm. We can treat this multi-sample setting by modifying the definition of the feature generation algorithm. We describe it in Sect. 5.3.

3.2 SRFEs

For a given biometric data setting, we define SRFEs as a tuple $(\mathrm{Gen}_{\mathrm{SRFE}}, \mathrm{Rep}_{\mathrm{SRFE}}, \mathrm{SRec}_{\mathrm{SRFE}})$ of generation, reproduction, and sample recovery algorithms so that (Property 1)–(Property 3) described in Sect. 1.4 are satisfied. As described in Sect. 1.4, $\mathrm{Gen}_{\mathrm{SRFE}}$ and $\mathrm{Rep}_{\mathrm{SRFE}}$ take a sample, not a feature, as an input, while $\mathrm{SRec}_{\mathrm{SRFE}}$ takes a feature. Also, $\mathrm{Rep}_{\mathrm{SRFE}}$ outputs an *additional reference* q. Furthermore, in general, online biometric authentication systems performs an enrollment process once and an authentication process multiple times. Considering these, we assume that $\mathrm{Gen}_{\mathrm{SRFE}}$ is performed once and $\mathrm{Rep}_{\mathrm{SRFE}}$ is performed multiple times[†], and the additional reference is stored only when the authentication is successful, i.e., only when the correct secret key r is reproduced. Then, when $i_1, \ldots, i_k$-th calls of $\mathrm{Rep}_{\mathrm{SRFE}}$ reproduce the correct secret key and additional references $q_{i_1}, \ldots, q_{i_k}$ are output respectively, the "stored data" in (Property 1) and (Property 2) are $(p, q_{i_1}, \ldots, q_{i_k})$. We define a security requirement for SRFEs so that r and a uniform random number are indistinguishable for any PPT adversary who obtains the stored data. By formulating the above, we define SRFEs as follows.

Definition 2. For a biometric data setting $\mathcal{B} = (\mathcal{X}, \mathrm{X}, \Delta, \mathcal{Y}, \mathrm{Feat}, \mathtt{dist}, t, \alpha)$ and a positive integer ℓ, $(\mathcal{B}, \ell)$-SRFE is a tuple $(\mathrm{Gen}_{\mathrm{SRFE}}, \mathrm{Rep}_{\mathrm{SRFE}}, \mathrm{SRec}_{\mathrm{SRFE}})$ satisfying the correctness and security requirements below.

- $\mathrm{Gen}_{\mathrm{SRFE}}(1^\lambda, x_{\mathrm{G}}) \to (r, p)$: The generation algorithm takes a security parameter 1^λ and a sample x_{G} as inputs, and outputs a secret key $r \in \{0,1\}^\ell$ and a helper string p.
- $\mathrm{Rep}_{\mathrm{SRFE}}(x_{\mathrm{R}}, p) \to (r', q)$: The reproduction algorithm takes a sample x_{R} and a helper string p as inputs, and outputs a reproduced secret key $r' \in \{0,1\}^\ell$ and an additional reference q.
- $\mathrm{SRec}_{\mathrm{SRFE}}(y_{\mathrm{SR}}, p, q) \to (x'_{\mathrm{G}}, x'_{\mathrm{R}})$ or $\perp$: The sample recovery algorithm takes a feature y_{SR}, a helper string p, and an additional reference q as inputs, and outputs either a pair of recovered samples $(x'_{\mathrm{G}}, x'_{\mathrm{R}})$ or a designated symbol $\perp$.

Correctness : We define the correctness requirement as follows: if samples $x_{\mathrm{G}}, x_{\mathrm{R}} \in \mathcal{X}$ and a feature $y_{\mathrm{SR}} \in \mathcal{Y}$ satisfy $\mathtt{dist}(\mathrm{Feat}(x_{\mathrm{G}}), \mathrm{Feat}(x_{\mathrm{R}})) \le t$ and $\mathtt{dist}(\mathrm{Feat}(x_{\mathrm{G}}), y_{\mathrm{SR}}) \le t$, then $r, r', x'_{\mathrm{G}}, x'_{\mathrm{R}}$ generated by $(r, p) \leftarrow \mathrm{Gen}_{\mathrm{SRFE}}(1^\lambda, x_{\mathrm{G}})$, $(r', q) \leftarrow \mathrm{Rep}(x_{\mathrm{R}}, p)$, and $(x'_{\mathrm{G}}, x'_{\mathrm{R}}) \leftarrow \mathrm{SRec}_{\mathrm{SRFE}}(y_{\mathrm{SR}}, p, q)$ satisfy $(r, x_{\mathrm{G}}, x_{\mathrm{R}}) = (r', x'_{\mathrm{G}}, x'_{\mathrm{R}})$.

Security : To define the security requirement, we define the SRFE experiment $\mathrm{Exp}^{\mathrm{SRFE}}_{\Sigma_{\mathrm{SRFE}}, \mathcal{A}}(\lambda)$ as follows:

- The challenger generates a random bit b by $b \leftarrow \{0,1\}$.
- The challenger generates (p, r_b) as follows: [$x_{\mathrm{G}} \leftarrow \mathrm{X}$; $(r, p) \leftarrow \mathrm{Gen}_{\mathrm{SRFE}}(1^\lambda, x_{\mathrm{G}})$; $r_0 \leftarrow r$; $r_1 \leftarrow \{0,1\}^\ell$; return (p, r_b).]
- For $i \in [n]$, the challenger generates q_i as follows: [$x_{\mathrm{R}} \leftarrow \Delta(x_{\mathrm{G}})$; $(r', q) \leftarrow \mathrm{Rep}_{\mathrm{SRFE}}(x_{\mathrm{R}}, p)$; if $r' \neq r$ then $q_i \leftarrow \perp$ else $q_i \leftarrow q$; return q_i.]
- The adversary $\mathcal{A}$ obtains $(p, r_b, q_1, \ldots, q_n)$, and generates $b' \in \{0,1\}$ using them. The output of $\mathrm{Exp}^{\mathrm{SRFE}}_{\Sigma_{\mathrm{SRFE}}, \mathcal{A}}(\lambda)$ is defined by b'.

We define the security requirement as follows: for any PPT adversary $\mathcal{A}$, the advantage $\mathrm{Adv}^{\mathrm{SRFE}}_{\Sigma_{\mathrm{SRFE}}, \mathcal{A}}(\lambda)$ defined by

$$\mathrm{Adv}^{\mathrm{SRFE}}_{\Sigma_{\mathrm{SRFE}}, \mathcal{A}}(\lambda) := \Big| \Pr[\mathrm{Exp}^{\mathrm{SRFE}}_{\Sigma_{\mathrm{SRFE}}, \mathcal{A}}(\lambda) = 1 \mid b = 1] - \Pr[\mathrm{Exp}^{\mathrm{SRFE}}_{\Sigma_{\mathrm{SRFE}}, \mathcal{A}}(\lambda) = 1 \mid b = 0] \Big|$$

is negligible.

4. Construction

In this section, we give a generic construction and instantiation of a SRFE scheme $\Sigma_{\mathrm{SRFE}} = (\mathrm{Gen}_{\mathrm{SRFE}}, \mathrm{Rep}_{\mathrm{SRFE}}, \mathrm{SRec}_{\mathrm{SRFE}})$. For construction, we use a $(\mathcal{Y}, \mu, \ell, t, \epsilon)$-fuzzy extractor $\Sigma_{\mathrm{FE}} = (\mathrm{Gen}_{\mathrm{FE}}, \mathrm{Rep}_{\mathrm{FE}})$ and a symmetric encryption scheme $\Sigma_{\mathrm{SE}} = (\mathrm{Gen}_{\mathrm{SE}}, \mathrm{Enc}_{\mathrm{SE}}, \mathrm{Dec}_{\mathrm{SE}})$ as building blocks.

4.1 Insecure Construction and Our Idea

Before giving our construction, we give an example of insecure construction. To satisfy the requirements, one might come up with the following construction. The output secret key r of Σ_{SRFE} is defined by the secret key r_{FE} of Σ_{FE}. Also, the captured sample during each process is encrypted with r_{FE} as the symmetric key, and the encrypted sample is contained in p or q. Concretely, this construction is described as follows:

- $\mathrm{Gen}_{\mathrm{SRFE}}(1^\lambda, x_{\mathrm{G}}) \to (r, p)$:
  $y_{\mathrm{G}} \leftarrow \mathrm{Feat}(x_{\mathrm{G}})$;
  $(r_{\mathrm{FE}}, p_{\mathrm{FE}}) \leftarrow \mathrm{Gen}_{\mathrm{FE}}(y_{\mathrm{G}})$;
  $r \leftarrow r_{\mathrm{FE}}$;
  $c_{\mathrm{G}} \leftarrow \mathrm{Enc}_{\mathrm{SE}}(r_{\mathrm{FE}}, x_{\mathrm{G}})$;
  $p \leftarrow (p_{\mathrm{FE}}, c_{\mathrm{G}})$;
  return (r, p).
- $\mathrm{Rep}_{\mathrm{SRFE}}(x_{\mathrm{R}}, p) \to (r', q)$:
  $y_{\mathrm{R}} \leftarrow \mathrm{Feat}(x_{\mathrm{R}})$;
  parse p as $(p_{\mathrm{FE}}, c_{\mathrm{G}})$;
  $r'_{\mathrm{FE}} \leftarrow \mathrm{Rep}_{\mathrm{FE}}(y_{\mathrm{R}}, p_{\mathrm{FE}})$;
  $r' \leftarrow r'_{\mathrm{FE}}$;
  $q \leftarrow \mathrm{Enc}_{\mathrm{SE}}(r'_{\mathrm{FE}}, x_{\mathrm{R}})$;

[†]In cases such as template update, $\mathrm{Gen}_{\mathrm{SRFE}}$ may also be performed multiple times. We can also satisfy security as described in Remark 3. To simplify the description, we mainly treat the case where $\mathrm{Gen}_{\mathrm{SRFE}}$ is performed only once.

return (r', q).

- $\mathrm{SRec}_{\mathrm{SRFE}}(y_{\mathrm{SR}}, p, q) \to (x'_{\mathrm{G}}, x'_{\mathrm{R}})$:
  $y_{\mathrm{SR}} \leftarrow \mathrm{Feat}(x_{\mathrm{SR}})$;
  parse p as $(p_{\mathrm{FE}}, c_{\mathrm{G}})$;
  $r'_{\mathrm{FE}} \leftarrow \mathrm{Rep}_{\mathrm{FE}}(y_{\mathrm{SR}}, p_{\mathrm{FE}})$;
  $x'_{\mathrm{G}} \leftarrow \mathrm{Dec}_{\mathrm{SE}}(r'_{\mathrm{FE}}, c_{\mathrm{G}})$;
  $x'_{\mathrm{R}} \leftarrow \mathrm{Dec}_{\mathrm{SE}}(r'_{\mathrm{FE}}, q)$;
  return $(x'_{\mathrm{G}}, x'_{\mathrm{R}})$.

It satisfies the correctness, but does not satisfy the security requirement. Indeed, in the SRFE experiment, the attacker can identify whether $b = 0$ or $b = 1$ by decrypting c_G in p with r_b because the attacker can recover the correct sample x_{G} if $b = 0$, while the attacker cannot recover it if $b = 1$. Also, we can observe that $\mathrm{Gen}_{\mathrm{SRFE}}$ encrypts the sample x_{G} with a secret key r_{FE} related to x_{G}. In such a construction, IND-CCA security is insufficient to make the encrypted sample indistinguishable with a uniform random number [18]. Considering them, our idea to satisfy the requirements is as follows:

- $\mathrm{Gen}_{\mathrm{SRFE}}$ generates a fresh random secret key for the output secret key r. Then, r is encrypted with r_{FE} and included into p.
- We require a kind of KDM security for Σ_{SE}, which we define as the *Biometric Encryption (BE) security*.

4.2 Generic Construction

Based on the idea described in Sect. 4.1, we give a generic construction of Σ_{SRFE} satisfying the requirements. We define the BE security as follows. We assume that all samples $x \in \mathcal{X}$ are represented by the same bit-length $|x|$. Consider the following BE experiment $\mathrm{Exp}^{\mathrm{BE}}_{\mathcal{B},\Sigma_{\mathrm{FE}},\Sigma_{\mathrm{SE}},\mathcal{A}}(\lambda)$ for $\mathcal{B}, \Sigma_{\mathrm{FE}}$, Σ_{SE}, and an adversary $\mathcal{A}$:

$$
\begin{aligned}
&\mathrm{Exp}^{\mathrm{BE}}_{\mathcal{B},\Sigma_{\mathrm{FE}},\Sigma_{\mathrm{SE}},\mathcal{A}}(\lambda) :\\
&[\ b \leftarrow \{0,1\};\ x_{\mathrm{G}} \leftarrow \mathrm{X};\ y_{\mathrm{G}} \leftarrow \mathrm{Feat}(x_{\mathrm{G}});\\
&\quad (r_{\mathrm{FE}}, p_{\mathrm{FE}}) \leftarrow \mathrm{Gen}_{\mathrm{FE}}(y_{\mathrm{G}});\\
&\quad c_1 \leftarrow \mathrm{Enc}_{\mathrm{SE}}(r_{\mathrm{FE}}, x_{\mathrm{G}});\ c_0 \leftarrow \mathrm{Enc}_{\mathrm{SE}}(r_{\mathrm{FE}}, 0^{|x|});\\
&\quad b' \leftarrow \mathcal{A}^{\widetilde{O}_{\mathrm{Enc}_{\mathrm{SE}}}, \widetilde{O}'_{\mathrm{Enc}_{\mathrm{SE}}}}(p_{\mathrm{FE}}, c_b);\ \text{return } b'\ ],
\end{aligned}
$$

where $\widetilde{O}_{\mathrm{Enc}_{\mathrm{SE}}}$ and $\widetilde{O}'_{\mathrm{Enc}_{\mathrm{SE}}}$ are defined as follows.

- The oracle $\widetilde{O}_{\mathrm{Enc}_{\mathrm{SE}}}$ takes a message m as an input, and operates as follows: $[\ m_1 := m;\ m_0 := 0^{|m|};\ c \leftarrow \mathrm{Enc}_{\mathrm{SE}}(r, m_b);\ \text{return } c\ ]$.
- The oracle $\widetilde{O}'_{\mathrm{Enc}_{\mathrm{SE}}}$ takes no input, and operates as follows: $[\ x_{\mathrm{R}} \leftarrow \Delta(x_{\mathrm{G}});\ x_1 := x_{\mathrm{R}};\ x_0 := 0^{|x|};\ c' \leftarrow \mathrm{Enc}_{\mathrm{SE}}(r_{\mathrm{FE}}, x_b);\ \text{return } c'\ ]$.

We say that a symmetric encryption scheme Σ_{SE} is *BE secure* with respect to a biometric data setting $\mathcal{B}$ and a fuzzy extractor scheme Σ_{FE} if for any PPT adversary $\mathcal{A}$,

$$
\begin{aligned}
&\mathrm{Adv}^{\mathrm{BE}}_{\mathcal{B},\Sigma_{\mathrm{FE}},\Sigma_{\mathrm{SE}},\mathcal{A}}(\lambda)\\
&:= \Big|\Pr[\mathrm{Exp}^{\mathrm{BE}}_{\mathcal{B},\Sigma_{\mathrm{FE}},\Sigma_{\mathrm{SE}},\mathcal{A}}(\lambda) = 1 \mid b = 1]\\
&\quad - \Pr[\mathrm{Exp}^{\mathrm{BE}}_{\mathcal{B},\Sigma_{\mathrm{FE}},\Sigma_{\mathrm{SE}},\mathcal{A}}(\lambda) = 1 \mid b = 0]\Big|
\end{aligned}
$$

is negligible.

For the building blocks Σ_{FE} and Σ_{SE}, we require the following:

1. The $(\mathcal{Y}, \mu, \ell, t, \epsilon)$-fuzzy extractor Σ_{FE} satisfies $\ell = \Theta(\lambda)$ and $\mu = \mathbf{H}_\infty(Y_{\mathrm{G}})$, where X_{G} denotes a random variable distributed according to X, and Y_{G} denotes a random variable defined by $Y_{\mathrm{G}} := \mathrm{Feat}(X_{\mathrm{G}})$.

2. The symmetric encryption scheme Σ_{SE}

   (2-a) has the key space $\{0,1\}^\ell$,
   (2-b) satisfies that for any m and any $sk_{\mathrm{SE}}, sk'_{\mathrm{SE}} \in \{0,1\}^\ell$ with $sk_{\mathrm{SE}} \neq sk'_{\mathrm{SE}}$, $\Pr[c \leftarrow \mathrm{Enc}_{\mathrm{SE}}(sk_{\mathrm{SE}}, m);\ m' \leftarrow \mathrm{Dec}_{\mathrm{SE}}(sk'_{\mathrm{SE}}, c) : m' = m]$ is negligible, and
   (2-c) is BE secure with respect to $\mathcal{B}$ and Σ_{FE}.

Using them, we construct $\Sigma_{\mathrm{SRFE}} = (\mathrm{Gen}_{\mathrm{SRFE}}, \mathrm{Rep}_{\mathrm{SRFE}}, \mathrm{SRec}_{\mathrm{SRFE}})$ as Fig. 3. For any algorithm, if it fails to operate "parse", then it returns $\perp$.

Theorem 1. $\Sigma_{\mathrm{SRFE}} = (\mathrm{Gen}_{\mathrm{SRFE}}, \mathrm{Rep}_{\mathrm{SRFE}}, \mathrm{SRec}_{\mathrm{SRFE}})$ defined by the above construction is a $(\mathcal{B}, \ell)$-SRFE.

Proof. Correctness holds because if samples $x_{\mathrm{G}}, x_{\mathrm{R}} \in \mathcal{X}$ and a feature y_{SR} satisfy $\mathtt{dist}(\mathrm{Feat}(x_{\mathrm{G}}), \mathrm{Feat}(x_{\mathrm{R}})) \le t$ and $\mathtt{dist}(\mathrm{Feat}(x_{\mathrm{G}}), y_{\mathrm{SR}}) \le t$, then r_{FE} is used as the symmetric key for all $\mathrm{Enc}_{\mathrm{SE}}$ and $\mathrm{Dec}_{\mathrm{SE}}$ operations.

We show the security requirement. Let $\mathcal{A}$ be any PPT adversary, and let A_i be the advantage of $\mathcal{A}$ in Game i.
Game 1 : We define Game 1 as the actual SRFE experiment. In this game, (p, r_b) obtained by the adversary is generated as follows:

$$
\begin{aligned}
&x_{\mathrm{G}} \leftarrow \mathrm{X};\ y_{\mathrm{G}} \leftarrow \mathrm{Feat}(x_{\mathrm{G}});\\
&(r_{\mathrm{FE}}, p_{\mathrm{FE}}) \leftarrow \mathrm{Gen}_{\mathrm{FE}}(y_{\mathrm{G}});\ r \leftarrow \{0,1\}^\ell;\\
&c_{\mathrm{G}} \leftarrow \mathrm{Enc}_{\mathrm{SE}}(r_{\mathrm{FE}}, x_{\mathrm{G}});\ z \leftarrow \mathrm{Enc}_{\mathrm{SE}}(r_{\mathrm{FE}}, r);\\
&p \leftarrow (p_{\mathrm{FE}}, c_{\mathrm{G}}, z);\ r_0 \leftarrow r;\ r_1 \leftarrow \{0,1\}^\ell;\\
&\text{return } (p, r_b).
\end{aligned}
$$

Furthermore, for $i \in [n]$, i-th additional reference q_i is generated as follows:

$$
\begin{aligned}
&x_{\mathrm{R}} \leftarrow \Delta(x_{\mathrm{G}});\ y_{\mathrm{R}} \leftarrow \mathrm{Feat}(x_{\mathrm{R}});\\
&\text{parse } p \text{ as } (p_{\mathrm{FE}}, c_{\mathrm{G}}, z);\ r'_{\mathrm{FE}} \leftarrow \mathrm{Rep}_{\mathrm{FE}}(y_{\mathrm{R}}, p_{\mathrm{FE}});\\
&r' \leftarrow \mathrm{Dec}_{\mathrm{SE}}(r'_{\mathrm{FE}}, z);\ q \leftarrow \mathrm{Enc}_{\mathrm{SE}}(r'_{\mathrm{FE}}, x_{\mathrm{R}});\\
&\text{if } r' \neq r \text{ then } q_i \leftarrow \perp \text{ else } q_i \leftarrow q;\ \text{return } q_i.
\end{aligned}
$$

Game 2 : In this game, we modify the generation of q_i as follows:

$$
\begin{aligned}
&x_{\mathrm{R}} \leftarrow \Delta(x_{\mathrm{G}});\ y_{\mathrm{R}} \leftarrow \mathrm{Feat}(x_{\mathrm{R}});\\
&\text{parse } p \text{ as } (p_{\mathrm{FE}}, c_{\mathrm{G}}, z);\ r'_{\mathrm{FE}} \leftarrow \mathrm{Rep}_{\mathrm{FE}}(y_{\mathrm{R}}, p_{\mathrm{FE}});\\
&\_;\ q \leftarrow \mathrm{Enc}_{\mathrm{SE}}(r_{\mathrm{FE}}, x_{\mathrm{R}});
\end{aligned}
$$

$\mathrm{Gen}_{\mathrm{SRFE}}(1^\lambda, x_\mathrm{G}) \to (r, p)$:
$y_\mathrm{G} \leftarrow \mathrm{Feat}(x_\mathrm{G})$;
$(r_\mathrm{FE}, p_\mathrm{FE}) \leftarrow \mathrm{Gen}_\mathrm{FE}(y_\mathrm{G})$;
$r \leftarrow \{0,1\}^\ell$;
$c_\mathrm{G} \leftarrow \mathrm{Enc}_\mathrm{SE}(r_\mathrm{FE}, x_\mathrm{G})$;
$z \leftarrow \mathrm{Enc}_\mathrm{SE}(r_\mathrm{FE}, r)$;
$p \leftarrow (p_\mathrm{FE}, c_\mathrm{G}, z)$;
return (r, p).

$\mathrm{Rep}_{\mathrm{SRFE}}(x_\mathrm{R}, p) \to (r', q)$:
$y_\mathrm{R} \leftarrow \mathrm{Feat}(x_\mathrm{R})$;
parse p as $(p_\mathrm{FE}, c_\mathrm{G}, z)$;
$r'_\mathrm{FE} \leftarrow \mathrm{Rep}_\mathrm{FE}(y_\mathrm{R}, p_\mathrm{FE})$;
$r' \leftarrow \mathrm{Dec}_\mathrm{SE}(r'_\mathrm{FE}, z)$;
$q \leftarrow \mathrm{Enc}_\mathrm{SE}(r'_\mathrm{FE}, x_\mathrm{R})$;
return (r', q).

$\mathrm{SRec}_{\mathrm{SRFE}}(y_\mathrm{SR}, p, q) \to (x'_\mathrm{G}, x'_\mathrm{R})$:
$y_\mathrm{SR} \leftarrow \mathrm{Feat}(x_\mathrm{SR})$;
parse p as $(p_\mathrm{FE}, c_\mathrm{G}, z)$;
$r'_\mathrm{FE} \leftarrow \mathrm{Rep}_\mathrm{FE}(y_\mathrm{SR}, p_\mathrm{FE})$;
$x'_\mathrm{G} \leftarrow \mathrm{Dec}_\mathrm{SE}(r'_\mathrm{FE}, c_\mathrm{G})$;
$x'_\mathrm{R} \leftarrow \mathrm{Dec}_\mathrm{SE}(r'_\mathrm{FE}, q)$;
return $(x'_\mathrm{G}, x'_\mathrm{R})$.

Fig. 3 Generic Construction of a SRFE.

if $\underline{r'_\mathrm{FE} \neq r_\mathrm{FE}}$ then $q_i \leftarrow \perp$ else $q_i \leftarrow q$; return q_i.

where the difference is underlined, and r_FE is the value generated during the generation of (p, r_b). Because $\Pr[(r'_\mathrm{FE} \neq r_\mathrm{FE}) \wedge (r' = r)] = \mathrm{negl}(\lambda)$ follows from (2-b), $|A_2 - A_1| = \mathrm{negl}(\lambda)$.

Game 3 : In this game, we modify the generation of (p, r_b) as follows:

$x_\mathrm{G} \leftarrow \mathrm{X}$; $y_\mathrm{G} \leftarrow \mathrm{Feat}(x_\mathrm{G})$;
$(r_\mathrm{FE}, p_\mathrm{FE}) \leftarrow \mathrm{Gen}_\mathrm{FE}(y_\mathrm{G})$; $r \leftarrow \{0,1\}^\ell$;
$c_\mathrm{G} \leftarrow \mathrm{Enc}_\mathrm{SE}(r_\mathrm{FE}, \underline{0^{|x|}})$; $z \leftarrow \mathrm{Enc}_\mathrm{SE}(r_\mathrm{FE}, \underline{0^\ell})$;
$p \leftarrow (p_\mathrm{FE}, c_\mathrm{G}, z)$; $r_0 \leftarrow r$; $r_1 \leftarrow \{0,1\}^\ell$;
return (p, r_b).

Furthermore, we modify the generation of q_i as follows:

$x_\mathrm{R} \leftarrow \Delta(x_\mathrm{G})$; $y_\mathrm{R} \leftarrow \mathrm{Feat}(x_\mathrm{R})$;
parse p as $(p_\mathrm{FE}, c_\mathrm{G}, z)$; $r'_\mathrm{FE} \leftarrow \mathrm{Rep}_\mathrm{FE}(y_\mathrm{R}, p_\mathrm{FE})$;
$q \leftarrow \mathrm{Enc}_\mathrm{SE}(r_\mathrm{FE}, \underline{0^{|x|}})$;
if $r'_\mathrm{FE} \neq r_\mathrm{FE}$ then $q_i \leftarrow \perp$ else $q_i \leftarrow q$; return q_i.

In other words, ciphertexts (c_G, z, q_i) in the data $(p_\mathrm{FE}, c_\mathrm{G}, z, q_i)$ obtained by the adversary are changed into dummy ones. Because of BE security, $|A_3 - A_2| = \mathrm{negl}(\lambda)$. In addition, both r_0 and r_1 in Game 3 are uniform random numbers on $\{0,1\}^\ell$, and $\mathcal{A}$ cannot obtain information on r_0 nor r_1 other than r_b, so $A_3 = \mathrm{negl}(\lambda)$ holds. Therefore, $\mathrm{Adv}^{\mathrm{SRFE}}_{\Sigma_\mathrm{SRFE}, \mathcal{A}}(\lambda) = A_1 \leq \sum_{i=2}^{3} |A_i - A_{i-1}| + A_3 = \mathrm{negl}(\lambda)$. □

4.3 Instantiation

In this subsection, we show that we can construct a SRFE scheme in the random oracle model by giving an example of the building blocks Σ_FE and Σ_SE satisfying the requirements in the generic construction.

We use an $(\mathcal{Y}, \mu, \ell, t, \epsilon)$-fuzzy extractor Σ_FE such that $\ell = \Theta(\lambda)$, $\mu = \mathbf{H}_\infty(Y_\mathrm{G})$ and $\epsilon = \mathrm{negl}(\lambda)$. Also, as Σ_SE, we use a symmetric encryption scheme by Black et al. [18][†] with the key space $\{0,1\}^\ell$. It can be described as follows. Let $\mathcal{H}$ be the random oracle, and let $\mathcal{H}_n(x)$ be the first n bits of $\mathcal{H}(x)$. Let $\oplus$ be the bitwise XOR operation. Then, $\Sigma_\mathrm{SE} = (\mathrm{Gen}_\mathrm{SE}, \mathrm{Enc}_\mathrm{SE}, \mathrm{Dec}_\mathrm{SE})$ is defined as follows:

- $\mathrm{Gen}_\mathrm{SE}(1^\ell)$: $sk_\mathrm{SE} \leftarrow \{0,1\}^\ell$; return sk_SE.
- $\mathrm{Enc}_\mathrm{SE}(sk_\mathrm{SE}, m)$: $v \leftarrow \{0,1\}^\ell$; $d \leftarrow \mathcal{H}_{|m|}(sk_\mathrm{SE} \,\|\, v) \oplus m$; $c \leftarrow (v, d)$; return c.
- $\mathrm{Dec}_\mathrm{SE}(sk_\mathrm{SE}, c)$: parse c as (v, d); $m \leftarrow \mathcal{H}_{|d|}(sk_\mathrm{SE} \,\|\, v) \oplus d$; return m.

Theorem 2. Σ_SRFE by the generic construction in Sect. 4.2 with the above Σ_FE and Σ_SE is a $(\mathcal{B}, \ell)$-SRFE in the random oracle model.

Proof. It suffices to show that Σ_FE and Σ_SE defined as above satisfy the requirements in Sect. 4.2. Requirements 1 and (2-a) follow from the definition of Σ_FE and Σ_SE, respectively. (2-b) holds because $[c \leftarrow \mathrm{Enc}_\mathrm{SE}(sk_\mathrm{SE}, m); m' \leftarrow \mathrm{Dec}_\mathrm{SE}(sk'_\mathrm{SE}, c) : m' = m]$ is satisfied only when $\mathcal{H}_{|m|}(sk_\mathrm{SE} \,\|\, v) = \mathcal{H}_{|m|}(sk'_\mathrm{SE} \,\|\, v)$ holds for a randomly chosen $v \leftarrow \{0,1\}^\ell$, and this probability is $\mathrm{negl}(\lambda)$.

We prove (2-c). Let $\mathcal{A}$ be any PPT adversary, and let A_i be the advantage of $\mathcal{A}$ in Game i. We define Game 1 as the actual BE experiment. To define Game 2, we define random variables R and P by $(R, P) \leftarrow \mathrm{Gen}_\mathrm{FE}(Y_\mathrm{G})$. Then, because $\mathbf{SD}((R, P), (U_\ell, P)) \leq \epsilon$ by the definition of Σ_FE, we can construct a probabilistic function F such that $\hat{R}$ defined by $\hat{R} \leftarrow F(R, P)$ satisfies the following two:

- $\Pr[\hat{R} \neq R] \leq \epsilon$, and
- $\hat{R}$ is uniformly distributed on $\{0,1\}^\ell$ given P.

By using F, in Game 2, we modify the generation of b' as follows:

$[\, b \leftarrow \{0,1\}$; $x_\mathrm{G} \leftarrow \mathrm{X}$; $y_\mathrm{G} \leftarrow \mathrm{Feat}(x_\mathrm{G})$;
$(r_\mathrm{FE}, p_\mathrm{FE}) \leftarrow \mathrm{Gen}_\mathrm{FE}(y_\mathrm{G})$; $\underline{\hat{r}_\mathrm{FE} \leftarrow F(r_\mathrm{FE}, p_\mathrm{FE})}$;
$c_1 \leftarrow \mathrm{Enc}_\mathrm{SE}(\underline{\hat{r}_\mathrm{FE}}, x_\mathrm{G})$; $c_0 \leftarrow \mathrm{Enc}_\mathrm{SE}(\underline{\hat{r}_\mathrm{FE}}, 0^{|x|})$;
$b' \leftarrow \mathcal{A}^{\hat{O}_{\mathrm{Enc}_\mathrm{SE}}, \hat{O}'_{\mathrm{Enc}_\mathrm{SE}}}(p_\mathrm{FE}, c_b)$; return b'],

where $\hat{O}_{\mathrm{Enc}_\mathrm{SE}}$ and $\hat{O}'_{\mathrm{Enc}_\mathrm{SE}}$ are defined as follows:

$\hat{O}_{\mathrm{Enc}_\mathrm{SE}}$:[$m_1 := m$; $m_0 := 0^{|m|}$;
$c \leftarrow \mathrm{Enc}_\mathrm{SE}(\underline{\hat{r}_\mathrm{FE}}, m_b)$; return c],

$\hat{O}'_{\mathrm{Enc}_\mathrm{SE}}$:[$x_\mathrm{R} \leftarrow \Delta(x_\mathrm{G})$; $x_1 := x_\mathrm{R}$; $x_0 := 0^{|x|}$;

[†]The scheme in [18] is $\mathcal{F}^{\mathrm{all}}$-KDM secure in the random oracle model, where $\mathcal{F}^{\mathrm{all}}$ consists of all functions such that the bit-length $|f(sk_\mathrm{SE})|$ is independent of sk_SE and the random oracle accessed in f.

$$c' \leftarrow \mathrm{Enc}_{\mathrm{SE}}(\underline{\hat{r}_{\mathrm{FE}}}, x_b);\ \text{return } c' \ ].$$

The output differs in Game 2 and Game 1 only when the value of $\hat{r}_{\mathrm{FE}}$ is changed from r_{FE}, and this happens with probability less than ϵ. Hence, $|A_1 - A_2| \leq \epsilon = \mathrm{negl}(\lambda)$. Furthermore, in Game 2, $\hat{r}_{\mathrm{FE}}$ is uniformly distributed given p_{FE}. From this and the definition of Σ_{SE}, the probability that $\mathcal{A}$ who obtains p_{FE} identifies the correct ciphertexts $(\mathrm{Enc}_{\mathrm{SE}}(\hat{r}_{\mathrm{FE}}, x_{\mathrm{G}}), \mathrm{Enc}_{\mathrm{SE}}(\hat{r}_{\mathrm{FE}}, m), \mathrm{Enc}_{\mathrm{SE}}(\hat{r}_{\mathrm{FE}}, x_{\mathrm{R}}))$ and the dummy ones $(\mathrm{Enc}_{\mathrm{SE}}(u, 0^{|x|}), \mathrm{Enc}_{\mathrm{SE}}(u, 0^{|m|}), \mathrm{Enc}_{\mathrm{SE}}(u, 0^{|x|}))$ is $\mathrm{negl}(\lambda)$. Hence, $A_2 = \mathrm{negl}(\lambda)$. Therefore, the advantage of $\mathcal{A}$ in $\mathrm{Exp}^{\mathrm{BE}}_{\mathcal{B},\Sigma_{\mathrm{FE}},\Sigma_{\mathrm{SE}},\mathcal{A}}(\lambda)$ is $A_1 = |A_1 - A_2| + A_2 = \mathrm{negl}(\lambda)$, i.e., (2-c) is satisfied. □

Remark 2. Theorem 2 holds in the random oracle model because Σ_{SE} relies on a random oracle. On the other hand, $\mathcal{F}$-KDM security without resorting to the use of random oracles is studied [20][21][22], which aims at enlarging the function class $\mathcal{F}$ or deriving a bound on $\mathcal{F}$. It is future work to construct a SRFE scheme without random oracles.

5. Application to Online Biometric Authentication

In this section, we propose a protocol of online biometric authentication system satisfying IASR using a SRFE, and analyze the system. To describe the protocol, we first define *Sample Recoverable Biometric Signatures (SRBSs)* and give a construction of a SRBS using a SRFE. Then, we describe the proposed protocol using a SRBS.

5.1 Sample Recoverable Biometric Signatures (SRBSs)

In online biometric authentication systems based on conventional FEs, a client device uses the reproduced secret key for signing a random message, called a challenge, generated by the server [11]. We refer to schemes generating a digital signature using a biometric-based secret as *Biometric Signatures (BSs)*, such as this combination of a FE and a digital signature[†]. To realize online biometric authentication with IASR, we define *Sample Recoverable Biometric Signatures (SRBSs)* as BSs which can recover samples $(x_{\mathrm{G}}, x_{\mathrm{S}})$ for key generation and signing processes from the verification key vk_{SRBS}, signatures Σ_{SRBS}, and a feature y_{SR} satisfying $\mathtt{dist}(\mathrm{Feat}(x_{\mathrm{G}}), y_{\mathrm{SR}}) \leq t$.

5.1.1 Definition

We define a SRBS scheme Σ_{SRBS} for a biometric data setting $\mathcal{B}$ by the following algorithms $(\mathrm{Gen}_{\mathrm{SRBS}}, \mathrm{Sign}_{\mathrm{SRBS}}, \mathrm{Ver}_{\mathrm{SRBS}}, \mathrm{SRec}_{\mathrm{SRBS}})$.

- $\mathrm{Gen}_{\mathrm{SRBS}}(1^\lambda, x_{\mathrm{G}}) \to vk_{\mathrm{SRBS}}$: The key generation algorithm takes a sample $x_{\mathrm{G}} \in \mathcal{X}$ as an input, and outputs a verification key vk_{SRBS}.
- $\mathrm{Sign}_{\mathrm{SRBS}}(x_{\mathrm{S}}, vk_{\mathrm{SRBS}}, m) \to \sigma_{\mathrm{SRBS}}$: The signing algorithm takes a sample $x_{\mathrm{S}} \in \mathcal{X}$, a verification key vk_{SRBS}, and a message $m \in \mathcal{M}$ as inputs, and outputs a signature σ_{SRBS} The signature σ_{SRBS} may be $\perp$.
- $\mathrm{Ver}_{\mathrm{SRBS}}(vk_{\mathrm{SRBS}}, m, \sigma_{\mathrm{SRBS}}) \to \mathit{result}$: The verification algorithm takes a verification key vk_{SRBS}, a message $m \in \mathcal{M}$, a signature σ_{SRBS} as inputs, and outputs the verification result $\mathit{result} \in \{\top, \perp\}$.
- $\mathrm{SRec}_{\mathrm{SRBS}}(y_{\mathrm{SR}}, vk_{\mathrm{SRBS}}, \sigma_{\mathrm{SRBS}}) \to (x'_{\mathrm{G}}, x'_{\mathrm{S}})$ or $\perp$: The sample recovery algorithm takes a feature $y_{\mathrm{SR}} \in \mathcal{Y}$, a verification key vk_{SRBS}, and a signature σ_{SRBS} as inputs, and outputs recovered samples $(x'_{\mathrm{G}}, x'_{\mathrm{S}}) \in \mathcal{X} \times \mathcal{X}$ or $\perp$.

We require correctness defined as follows: for any $x_{\mathrm{G}}, x_{\mathrm{S}} \in \mathcal{X}$ and $y_{\mathrm{SR}} \in \mathcal{Y}$ satisfying $\mathtt{dist}(\mathrm{Feat}(x_{\mathrm{G}}), \mathrm{Feat}(x_{\mathrm{S}})) \leq t$ and $\mathtt{dist}(\mathrm{Feat}(x_{\mathrm{G}}), y_{\mathrm{SR}}) \leq t$, any $m \in \mathcal{M}$, and any $(vk_{\mathrm{SRBS}}, \sigma_{\mathrm{SRBS}})$ generated by $vk_{\mathrm{SRBS}} \leftarrow \mathrm{Gen}_{\mathrm{SRBS}}(1^\lambda, x_{\mathrm{G}})$ and $\sigma_{\mathrm{SRBS}} \leftarrow \mathrm{Sign}_{\mathrm{SRBS}}(x_{\mathrm{S}}, vk_{\mathrm{SRBS}}, m)$, $\mathrm{Ver}_{\mathrm{SRBS}}(vk_{\mathrm{SRBS}}, m, \sigma_{\mathrm{SRBS}}) = \top$ and $\mathrm{SRec}_{\mathrm{SRBS}}(x_{\mathrm{SR}}, vk_{\mathrm{SRBS}}, \sigma_{\mathrm{SRBS}}) = (x_{\mathrm{G}}, x_{\mathrm{S}})$.

Next, we define EUF-CMA security for Σ_{SRBS}. Consider the following EUF-CMA experiment $\mathrm{Exp}^{\text{EUF-CMA}}_{\Sigma_{\mathrm{SRBS}},\mathcal{A}}(\lambda)$ for a SRBS scheme $\Sigma_{\mathrm{SRBS}} = (\mathrm{Gen}_{\mathrm{SRBS}}, \mathrm{Sign}_{\mathrm{SRBS}}, \mathrm{Ver}_{\mathrm{SRBS}}, \mathrm{SRec}_{\mathrm{SRBS}})$ and an adversary $\mathcal{A}$:

$$\begin{aligned}
&\mathrm{Exp}^{\text{EUF-CMA}}_{\Sigma_{\mathrm{SRBS}},\mathcal{A}}(\lambda) : \\
&\quad [\ x_{\mathrm{G}} \leftarrow \mathrm{X};\ vk_{\mathrm{SRBS}} \leftarrow \mathrm{Gen}_{\mathrm{SRBS}}(1^\lambda, x_{\mathrm{G}}); \\
&\quad\ Q := \emptyset;\ (m', \sigma'_{\mathrm{SRBS}}) \leftarrow \mathcal{A}^{O_{\mathrm{Sign}_{\mathrm{SRBS}}}}(vk_{\mathrm{SRBS}}); \\
&\quad\ \text{if } m' \notin Q \wedge \mathrm{Ver}(vk_{\mathrm{SRBS}}, m', \sigma'_{\mathrm{SRBS}}) = \top \\
&\quad\ \text{then return 1 else return 0 }],
\end{aligned}$$

where $O_{\mathrm{Sign}_{\mathrm{SRBS}}}$ is a signing oracle that takes a message $m \in \mathcal{M}$ as an input and operates as follows: $[\, Q := Q \cup \{m\};\ x_{\mathrm{S}} \leftarrow \Delta(x_{\mathrm{G}});\ \sigma_{\mathrm{SRBS}} \leftarrow \mathrm{Sign}_{\mathrm{SRBS}}(x_{\mathrm{S}}, vk_{\mathrm{SRBS}}, m);\ \text{return } \sigma_{\mathrm{SRBS}}\ ]$.

We say that a SRBS scheme Σ_{SRBS} is EUF-CMA secure if for any PPT adversary $\mathcal{A}$, $\mathrm{Adv}^{\text{EUF-CMA}}_{\Sigma_{\mathrm{SRBS}},\mathcal{A}}(\lambda) := \Pr[\mathrm{Exp}^{\text{EUF-CMA}}_{\Sigma_{\mathrm{SRBS}},\mathcal{A}}(\lambda) = 1]$ is negligible.

5.1.2 Construction Using SRFEs

Let $\Sigma_{\mathrm{DS}} = (\mathrm{Gen}_{\mathrm{DS}}, \mathrm{Sign}_{\mathrm{DS}}, \mathrm{Ver}_{\mathrm{DS}})$ be a digital signature scheme satisfying the following properties:

- $\mathrm{Gen}_{\mathrm{DS}}(1^\lambda)$ can be expressed by a deterministic algorithm Gen^* as $\mathrm{Gen}_{\mathrm{DS}}(1^\lambda) = \mathrm{Gen}^*(\mathit{seed})$, where seed is a uniform random number on $\{0,1\}^\lambda$.
- For any $\mathit{seed}, \mathit{seed}' \in \{0,1\}^\lambda$ with $\mathit{seed} \neq \mathit{seed}'$, $\Pr[(sk, vk) \leftarrow \mathrm{Gen}^*(\mathit{seed});\ (sk', vk') \leftarrow \mathrm{Gen}^*(\mathit{seed}') : vk = vk'] = \mathrm{negl}(\lambda)$.
- EUF-CMA secure.

By using Σ_{DS} and a $(\mathcal{B}, \lambda)$-SRFE $\Sigma_{\mathrm{SRFE}} = (\mathrm{Gen}_{\mathrm{SRFE}}, \mathrm{Rep}_{\mathrm{SRFE}}, \mathrm{SRec}_{\mathrm{SRFE}})$, we construct a EUF-CMA secure SRBS $\Sigma_{\mathrm{SRBS}} = (\mathrm{Gen}_{\mathrm{SRBS}}, \mathrm{Sign}_{\mathrm{SRBS}}, \mathrm{Ver}_{\mathrm{SRBS}}, \mathrm{Rec}_{\mathrm{SRBS}})$ as follows:

- $\mathrm{Gen}_{\mathrm{SRBS}}(1^\lambda, x_{\mathrm{G}}) \to vk_{\mathrm{SRBS}}$:
  $(r_{\mathrm{SRFE}}, p_{\mathrm{SRFE}}) \leftarrow \mathrm{Gen}_{\mathrm{SRFE}}(1^\lambda, x_{\mathrm{G}})$;

[†]BSs realized by combining a fuzzy extractor and a digital signature require the helper string for signing. On the other hand, BSs not requiring the helper string for signing can be realized by fuzzy signatures [23]. It is an open problem to realize a SRBS not requiring vk_{SRBS} for signing.

$(sk_{\mathrm{DS}}, vk_{\mathrm{DS}}) \leftarrow \mathrm{Gen}^*(r_{\mathrm{SRFE}})$;
$vk_{\mathrm{SRBS}} \leftarrow (vk_{\mathrm{DS}}, p_{\mathrm{SRFE}})$;
return vk_{SRBS}.

- $\mathrm{Sign}_{\mathrm{SRBS}}(x_{\mathrm{S}}, vk_{\mathrm{SRBS}}, m) \rightarrow \sigma_{\mathrm{SRBS}}$:
  parse vk_{SRBS} as $(vk_{\mathrm{DS}}, p_{\mathrm{SRFE}})$;
  $(r'_{\mathrm{SRFE}}, q_{\mathrm{SRFE}}) \leftarrow \mathrm{Rep}_{\mathrm{SRFE}}(x_{\mathrm{S}}, p_{\mathrm{SRFE}})$;
  $(sk'_{\mathrm{DS}}, vk'_{\mathrm{DS}}) \leftarrow \mathrm{Gen}^*(r'_{\mathrm{SRFE}})$;
  if $vk'_{\mathrm{DS}} \neq vk_{\mathrm{DS}}$ then return $\perp$;
  $\sigma_{\mathrm{DS}} \leftarrow \mathrm{Sign}_{\mathrm{DS}}(sk'_{\mathrm{DS}}, m)$;
  $\sigma_{\mathrm{SRBS}} \leftarrow (\sigma_{\mathrm{DS}}, q_{\mathrm{SRFE}})$;
  return σ_{SRBS}.
- $\mathrm{Ver}_{\mathrm{SRBS}}(vk_{\mathrm{SRBS}}, m, \sigma_{\mathrm{SRBS}}) \rightarrow \mathit{result}$:
  parse vk_{SRBS} as $(vk_{\mathrm{DS}}, p_{\mathrm{SRFE}})$;
  parse σ_{SRBS} as $(\sigma_{\mathrm{DS}}, q_{\mathrm{SRFE}})$;
  $\mathit{result} \leftarrow \mathrm{Ver}_{\mathrm{DS}}(vk_{\mathrm{DS}}, m, \sigma_{\mathrm{DS}})$;
  return result.
- $\mathrm{SRec}_{\mathrm{SRBS}}(y_{\mathrm{SR}}, vk_{\mathrm{SRBS}}, \sigma_{\mathrm{SRBS}}) \rightarrow (x'_{\mathrm{G}}, x'_{\mathrm{S}})$ or $\perp$:
  parse vk_{SRBS} as $(vk_{\mathrm{DS}}, p_{\mathrm{SRFE}})$;
  parse σ_{SRBS} as $(\sigma_{\mathrm{DS}}, q_{\mathrm{SRFE}})$;
  return $\mathrm{SRec}_{\mathrm{SRFE}}(y_{\mathrm{SR}}, p_{\mathrm{SRFE}}, q_{\mathrm{SRFE}})$.

Theorem 3. $\Sigma_{\mathrm{SRBS}} = (\mathrm{Gen}_{\mathrm{SRBS}}, \mathrm{Sign}_{\mathrm{SRBS}}, \mathrm{Ver}_{\mathrm{SRBS}}, \mathrm{SRec}_{\mathrm{SRBS}})$ constructed as above is a SRBS with EUF-CMA security.

Proof. Correctness follows from that of Σ_{SRFE} and Σ_{DS}. We prove EUF-CMA security. Let $\mathcal{A}$ be a PPT adversary, and A_i be the advantage of $\mathcal{A}$ in Game i.

Game 1 : We define Game 1 as the actual EUF-CMA experiment for Σ_{SRBS}. In Game 1, the verification key vk_{SRBS} is generated as follows:

$$(r_{\mathrm{SRFE}}, p_{\mathrm{SRFE}}) \leftarrow \mathrm{Gen}_{\mathrm{SRFE}}(1^\lambda, x_{\mathrm{G}});$$
$$(sk_{\mathrm{DS}}, vk_{\mathrm{DS}}) \leftarrow \mathrm{Gen}^*(r_{\mathrm{SRFE}});\ vk_{\mathrm{SRBS}} \leftarrow (vk_{\mathrm{DS}}, p_{\mathrm{SRFE}}).$$

Furthermore, when $\mathcal{A}$ makes the i-th signing query on a message m_i, the challenger generates a signature σ_i as follows:

$$x_{\mathrm{S}} \leftarrow \Delta(x_{\mathrm{G}});\ \text{parse } vk_{\mathrm{SRBS}} \text{ as } (vk_{\mathrm{DS}}, p_{\mathrm{SRFE}});$$
$$(r'_{\mathrm{SRFE}\,i}, q_{\mathrm{SRFE}\,i}) \leftarrow \mathrm{Rep}_{\mathrm{FE}}(x_{\mathrm{S}}, p_{\mathrm{SRFE}});$$
$$(sk'_{\mathrm{DS}\,i}, vk'_{\mathrm{DS}\,i}) \leftarrow \mathrm{Gen}^*(r'_{\mathrm{SRFE}});$$
$$\text{if } vk'_{\mathrm{DS}\,i} \neq vk_{\mathrm{DS}} \text{ then return } \perp;$$
$$\sigma_{\mathrm{DS}\,i} \leftarrow \mathrm{Sign}_{\mathrm{DS}}(sk'_{\mathrm{DS}\,i}, m_i);\ \sigma_i \leftarrow (\sigma_{\mathrm{DS}\,i}, q_{\mathrm{SRFE}\,i}).$$

Game 2 : In this game, the challenger generates $\sigma_{\mathrm{SRBS},i}$ as follows:

$$x_{\mathrm{S}} \leftarrow \Delta(x_{\mathrm{G}});\ \text{parse } vk_{\mathrm{SRBS}} \text{ as } (vk_{\mathrm{DS}}, p_{\mathrm{SRFE}});$$
$$(r'_{\mathrm{SRFE}\,i}, q_{\mathrm{SRFE}\,i}) \leftarrow \mathrm{Rep}_{\mathrm{FE}}(x_{\mathrm{S}}, p_{\mathrm{SRFE}});$$
$$(sk'_{\mathrm{DS}\,i}, vk'_{\mathrm{DS}\,i}) \leftarrow \mathrm{Gen}^*(r'_{\mathrm{SRFE}});$$
$$\text{if } \underline{r'_{\mathrm{SRFE}\,i} \neq r_{\mathrm{SRFE}}} \text{ then return } \perp;$$
$$\sigma_{\mathrm{DS}\,i} \leftarrow \mathrm{Sign}_{\mathrm{DS}}(\underline{sk_{\mathrm{DS}\,i}}, m_i);\ \sigma_i \leftarrow (\sigma_{\mathrm{DS}\,i}, q_{\mathrm{SRFE}\,i}).$$

where r_{SRFE} is the value generated during vk_{SRBS} generation. Because $\Pr[r'_{\mathrm{SRFE}\,i} \neq r_{\mathrm{SRFE}} \wedge vk'_{\mathrm{DS}\,i} = vk_{\mathrm{DS}}] = \mathrm{negl}(\lambda)$, $|A_2 - A_1| \leq \mathrm{negl}(\lambda)$.

Game 3 : In this game, the challenger generates vk_{SRBS} as follows:

$$(r_{\mathrm{SRFE}}, p_{\mathrm{SRFE}}) \leftarrow \mathrm{Gen}_{\mathrm{SRFE}}(1^\lambda, x_{\mathrm{G}});\ \underline{u \leftarrow \{0,1\}^\lambda};$$
$$(sk_{\mathrm{DS}}, vk_{\mathrm{DS}}) \leftarrow \mathrm{Gen}^*(\underline{u});\ vk_{\mathrm{SRBS}} \leftarrow (vk_{\mathrm{DS}}, p_{\mathrm{SRFE}}).$$

Furthermore, the challenge generates σ_i as follows:

$$x_{\mathrm{S}} \leftarrow \Delta(x_{\mathrm{G}});\ \text{parse } vk_{\mathrm{SRBS}} \text{ as } (vk_{\mathrm{DS}}, p_{\mathrm{SRFE}});$$
$$(r'_{\mathrm{SRFE}\,i}, q_{\mathrm{SRFE}\,i}) \leftarrow \mathrm{Rep}_{\mathrm{FE}}(x_{\mathrm{S}}, p_{\mathrm{SRFE}});$$
$$(sk'_{\mathrm{DS}\,i}, vk'_{\mathrm{DS}\,i}) \leftarrow \mathrm{Gen}^*(\underline{u});$$
$$\text{if } r'_{\mathrm{SRFE}\,i} \neq r_{\mathrm{SRFE}} \text{ then return } \perp;$$
$$\sigma_{\mathrm{DS}\,i} \leftarrow \mathrm{Sign}_{\mathrm{DS}}(sk_{\mathrm{DS}\,i}, m_i);\ \sigma_i \leftarrow (\sigma_{\mathrm{DS}\,i}, q_{\mathrm{SRFE}\,i}).$$

where u is the value generated during vk_{SRBS} generation. Because of the security requirement of a $(\mathcal{B}, \lambda)$-SRFE $\Sigma_{\mathrm{SRFE}} = (\mathrm{Gen}_{\mathrm{SRFE}}, \mathrm{Rep}_{\mathrm{SRFE}}, \mathrm{SRec}_{\mathrm{SRFE}})$, $|A_3 - A_2| = \mathrm{negl}(\lambda)$ holds. Also, from EUF-CMA security of Σ_{DS} and the fact that $(p_{\mathrm{SRFE}}, \{q_{\mathrm{SRFE}\,i}\}_{i\in[n]})$ obtained by $\mathcal{A}$ is independent of $(vk_{\mathrm{DS}}, \{\sigma_{\mathrm{DS}\,i}\}_{i\in[n]})$, $A_3 = \mathrm{negl}(\lambda)$ holds. Therefore, $\mathrm{Adv}^{\text{EUF-CMA}}_{\Sigma_{\mathrm{SRBS}},\mathcal{A}}(\lambda) = A_1 \leq \sum_{i=2}^{3} |A_i - A_{i-1}| + A_3 = \mathrm{negl}(\lambda)$ holds. □

5.2 Online Biometric Authentication System with IASR

We give a protocol of an online biometric authentication with IASR using a SRBS with EUF-CMA security. We consider an online biometric authentication system which consists of client devices and an authentication server. We make following assumptions:

- A biometric data setting $\mathcal{B} = (\mathcal{X}, \mathrm{X}, \Delta, \mathcal{Y}, \mathrm{Feat}, \mathtt{dist}, t, \alpha)$ is given, and captured samples are according to $\mathcal{B}$. Concretely, a sample captured at enrollment is according to the distribution X, and when a sample x_{E} is captured from a user at enrollment, a sample x_{A} captured from the user at authentication and a sample x_{SR} at sample recovery are according to the distribution $\Delta(x_{\mathrm{E}})$.
- The server follows the protocol but data that appears in the server may leak.
- The risk of leaking data that appears temporarily in client devices during processes for legitimate users are sufficiently low.

Under these assumptions, by using an SRBS scheme $\Sigma_{\mathrm{SRBS}} = (\mathrm{Gen}_{\mathrm{SRBS}}, \mathrm{Sign}_{\mathrm{SRBS}}, \mathrm{Ver}_{\mathrm{SRBS}}, \mathrm{Rec}_{\mathrm{SRBS}})$ with EUF-CMA security, we describe the process flow of authentication system.

- Enrollment: A client device creates a user ID id, captures a sample $x_{\mathrm{E}} \leftarrow \mathrm{X}$, and generates vk_{SRBS} by $vk_{\mathrm{SRBS}} \leftarrow \mathrm{Gen}_{\mathrm{SRBS}}(1^\lambda, x_{\mathrm{E}})$. Then, id and vk_{SRBS} are sent to and stored in the server. vk_{SRBS} is the enrolled template in our scheme.
- Authentication: A client device gets the user ID id, and sends it to the server. The server sends a fresh

challenge m and vk_{SRBS}[†] corresponding to id to the client device. Then, the client device captures a sample x_{A} from the user, generates σ_{SRBS} by $\sigma_{\mathrm{SRBS}} \leftarrow \mathrm{Sign}_{\mathrm{SRBS}}(x_{\mathrm{A}}, vk_{\mathrm{SRBS}}, m)$, and sends σ_{SRBS} to the server. The server determines the authentication result $result$ by $result \leftarrow \mathrm{Ver}_{\mathrm{SRBS}}(vk_{\mathrm{SRBS}}, m, \sigma_{\mathrm{SRBS}})$. If $result = \top$, then the server stores σ_{SRBS}.

- Sample Recovery: A client device gets the user ID id, and sends it to the server. The server sends vk_{SRBS} and σ_{SRBS} corresponding to id to the client device. Then, the client device captures a sample x_{SR} from the user, extracts a feature y_{SR} by $y_{\mathrm{SR}} \leftarrow \mathrm{Feat}(x_{\mathrm{SR}})$, and recovers samples $(x'_{\mathrm{E}}, x'_{\mathrm{A}})$ by $(x'_{\mathrm{E}}, x'_{\mathrm{A}}) \leftarrow \mathrm{Rec}_{\mathrm{SRBS}}(y_{\mathrm{SR}}, vk_{\mathrm{SRBS}}, \sigma_{\mathrm{SRBS}})$.

The above system satisfies IASR as follows. Irreversibility is satisfied because the adversary who obtains $\mathrm{Feat}(x_{\mathrm{E}})$ or $\mathrm{Feat}(x_{\mathrm{A}})$ can recover x_{E} from the stored data $(vk_{\mathrm{SRBS}}, \sigma_{\mathrm{SRBS}})$ by the sample recovery process, and can forge a signature using the recovered sample. Also, from a sample x_{SR} the system can recover samples $(x_{\mathrm{E}}, x_{\mathrm{A}})$ having captured during past enrollment and authentication processes by the sample recovery process using the stored $(vk_{\mathrm{SRBS}}, \sigma_{\mathrm{SRBS}})$ and a sample x_{SR} which can succeed in the authentication, i.e., which satisfies $\mathtt{dist}(\mathrm{Feat}(x_{\mathrm{E}}), \mathrm{Feat}(x_{\mathrm{SR}})) \le t$.

Remark 3. To ensure security even when the same user enrolls multiple times, we can use a reusable FE [11] for Σ_{FE}. Also, to ensure security even when the server modifies the helper string p maliciously, we can use a robust FE [24], which can detect the modification and abort the reproduction process.

5.3 Application to Template Update

In this subsection, we describe the differences on the biometric data setting and process flows for application to template update. A sample recovery process is executed during a template update process, and the system is required to check whether the user trying to update a template is the genuine user. We assume the check is done by biometric authentication, and consider enrollment, authentication, and template update with authentication.

To generate a template with multi-sample fusion, an enrollment process has to capture N_{E} samples, where $N_{\mathrm{E}} > 1$. In other words, the enrollment process requires a sample set $\boldsymbol{x}_{\mathrm{E}} \in \mathcal{X}^{N_{\mathrm{E}}}$. On feature extraction, we assume that feature extractors $\mathrm{Feat}_{\mathrm{E}}$ for enrollment is given, and a feature y_{E} for enrollment is extracted by $y_{\mathrm{E}} \leftarrow \mathrm{Feat}_{\mathrm{E}}(x_{\mathrm{E}})$[††].

By modifying the settings as above and processes of Σ_{SRBS} accordingly, enrollment and authentication processes can be performed as well as those in Sect. 5.2. We describe the flow of template update with authentication process. A client device gets the user ID id, and sends it to the server. The server sends vk_{SRBS} and σ_{SRBS} corresponding to id to the client device. Then, the client device captures a sample x_{SR} from the user, extracts a feature $y_{\mathrm{SR}} \in \mathcal{Y}$ by $y_{\mathrm{SR}} \leftarrow \mathrm{Feat}(x_{\mathrm{SR}})$, and recovers samples $(\boldsymbol{x}'_{\mathrm{E}}, x'_{\mathrm{A}})$ by $(\boldsymbol{x}'_{\mathrm{E}}, x'_{\mathrm{A}}) \leftarrow \mathrm{Rec}_{\mathrm{SRBS}}(y_{\mathrm{SR}}, vk_{\mathrm{SRBS}}, \sigma_{\mathrm{SRBS}})$. If the user has succeeded in authentication k times, then k samples can be recovered as x'_{A}, so the client device can obtain $N_{\mathrm{E}} + k + 1$ samples in total. It chooses N_{E} samples from them and generates a new verification key vk'_{SRBS} using the chosen samples. Then, vk'_{SRBS} is sent to and stored in the server as a new template.

With this flow, the system can perform the template update process simply by a user inputting a sample as a usual authentication process when he/she wants to be authenticated, e.g., when he/she wants to log in to a online service. Furthermore, because the client device obtains $N_{\mathrm{E}} + k + 1$ samples, it can choose samples more appropriate for generating a template, e.g., clearly captured ones. On the other hand, systems which do not support sample recovery have to require a user to input at least $N_{\mathrm{E}} - 1$ additional samples for the update process, which is a burden for users.

We note that, as described in Remark 3, using a reusable FE as a building block ensures security even when multiple templates may leak.

5.4 Performance Analysis

We analyze the performance of the proposed system. Specifically, we analyze impact on storage size, authentication accuracy, and processing time.

Impact on Storage Size:

Because the system using our SRFE stores encrypted samples additionally, it requires additional resource accordingly. We estimate the impact on storage size for (Case A)–(Case C) described in Sect. 1. We assume that the system has 10,000 enrolled users, and an encrypted sample size is 100 kB.

In (Case A), it suffices that the system stores encrypted samples of users relatively frequently rejected, and the system only needs to store a sufficient number of their encrypted samples to identify the cause. We assume that 1% of the enrolled users are relatively frequently rejected, and the system stores 10 samples for each of them for the investigation of the cause. Then, the additional storage cost by our scheme can be estimated as 0.1 GB. With this cost, it becomes possible to investigate a cause of the frequent rejection in more detail

[†]For this process, the server does not have to send $\mathrm{Enc}_{\mathrm{SE}}(r_{\mathrm{FE}}, x_{\mathrm{E}})$ contained in $vk_{\mathrm{SRBS}} = (vk_{\mathrm{DS}}, p_{\mathrm{SRFE}}) = (vk_{\mathrm{DS}}, p_{\mathrm{FE}}, \mathrm{Enc}_{\mathrm{SE}}(r_{\mathrm{FE}}, x_{\mathrm{E}}), \mathrm{Enc}_{\mathrm{SE}}(r_{\mathrm{FE}}, r))$.

[††]An example of generating a feature y from multiple samples $(x_1, \ldots, x_N)$ is to select the most appropriate one by a specific rule. Another example when $\mathcal{Y} = \{0,1\}^N$ is as follows. For each sample x_i, the algorithm extracts a feature y_i by $y_i = \mathrm{Feat}(x_i)$. Then, for each $j \in [N]$ it determines the j-th bit $y(j)$ of y by 1 if the j-th bit is 1 for more than half of $(y_i)_{i\in[N]}$, and determines $y(i)$ by 0 otherwise. Also, an example when $\mathcal{Y} = \mathbb{R}^N$ is to determine y by the average of $(y_i)_{i\in[N]}$.

with samples having captured during past processes. This will lead to fewer false rejects, meaning that user convenience can be improved.

In (Case B), the system only needs to store a sufficient number of encrypted samples for template update. We assume that the system stores 5 samples for each user, which is the number adopted in [16]. Then, the additional storage cost by our scheme can be estimated as 5 GB. By paying this cost, the system can improve authentication accuracy by multi-sample fusion during the template update process without the burden of scanning samples many times.

In (Case C), we assume that the system stores the sample for enrollment and ones for authentication during the past one week, and the authentication is performed once a day. Then, the additional storage cost can be estimated as 8 GB. With this cost, it becomes possible to post-verify a past process using the sample having captured during the process.

We believe that the system with 10,000 enrolled users can tolerate these storage costs. In particular, in (Case A) and (Case B), by paying these costs, the system can improve user convenience, which is important for widespread use of the system.

Authentication Accuracy:

In general, biometric authentication has the characteristic of probabilistic acceptance of imposters and probabilistic rejection of the genuine user. The rates of them are called False Acceptance Rate (FAR) and False Rejection Rate (FRR), respectively. Also, for our system, similar metrics can be considered on the sample recovery process. We refer to the rate that an imposter succeeds in the sample recovery as False Recovery Success Rate (FRSR), and the rate that the genuine user fails in the sample recovery as False Recovery Failure Rate (FRFR)[†]. These rates indicate limitations on security and correctness, so they have to be sufficiently lowered.

FAR and FRR of our system are equal to those of Σ_{FE} respectively. Also, FRSR and FRFR are equal to FAR and FRR of Σ_{FE}, respectively. Therefore, we have to use a FE with low FAR and FRR as Σ_{FE}. Studies have been done [25][26] for evaluating and improving accuracy on FEs, so it suffices to choose one with low FAR and FRR as Σ_{FE}.

Authentication Time:

In an authentication process, our system additionally executes decryption of the seed and encryption of the sample for authentication. Both are operations on a symmetric encryption algorithm. On the other hand, authentication process usually includes sub-process which takes more time, e.g., feature extraction using technique such as deep learning. Therefore, we believe that the symmetric encryption and decryption processes have only a small impact on the total authentication time.

[†]On these rates for the sample recovery, we consider the situation in which authentication has already been performed successfully.

Sample Recovery Time:

The operations in the sample recovery other than the decryption of samples are included in the authentication process, and the decryption of samples takes only a short time. Therefore, the sample recovery time is less than or near to the authentication time, and practical if authentication time is practical.

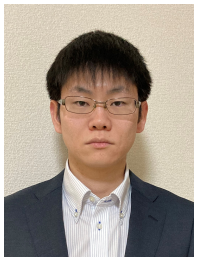
6. Conclusion

In this paper, to realize an online biometric authentication system satisfying Irreversibility with Authenticated Sample Recoverability (IASR), we introduced Sample Recoverable Fuzzy Extractors (SRFEs). In Sect. 3, we gave a formal definition of a SRFE so that it satisfies the following (1)–(3): (1) From the stored data and a sample close to one for the generation process, the secret key can be correctly reproduced. (2) From the stored data and a feature extracted to a sample close to one for the generation process, the sample recovery process can correctly recover the samples for generation and successful reproduction processes. (3) It is difficult for any PPT adversary who obtains the stored data to distinguish the secret key and a uniform random number. In Sect. 4, we gave a generic construction and an instantiation of SRFEs. In Sect. 5, by using a SRFE, we proposed a protocol of an online biometric authentication system satisfying IASR using a SRFE, and analyzed the proposed system. We believe that the system is useful in various situations in which past samples are desired to be utilized, while it prevents biometric data from leakage.

References

[1] W. Nakamura and K. Takahashi, "A biometric signature scheme with template protection and authenticated sample recoverability," 2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2023), pp.1784–1791, 2023.
[2] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication," Proceedings of the IEEE, vol.91, no.12, pp.2021–2040, 2003.
[3] W. Meng, D.S. Wong, S. Furnell, and J. Zhou, "Surveying the development of biometric user authentication on mobile phones," IEEE Communications Surveys & Tutorials, vol.17, no.3, pp.1268–1293, 2014.
[4] FIDO Alliance, "How FIDO works." https://fidoalliance.org/how-fido-works/ (Accessed: 2022/03/21).
[5] G. of India, "What is aadhaar." https://uidai.gov.in/my-aadhaar/about-your-aadhaar.html (Accessed: 2022/03/21), 2019.
[6] T. Kawakami and Y. Hinata, "Pay with your face: 100m chinese switch from smartphones." https://asia.nikkei.com/Business/China-tech/Pay-with-your-face-100m-Chinese-switch-from-smartphones (Accessed: 2022/03/21), 2019.
[7] V. Doshi, "A security breach in india has left a billion people at risk of identity theft." https://www.washingtonpost.com/news/worldviews/wp/2018/01/04/a-security-breach-in-india-has-left-a-billion-people-at-risk-of-identity-theft/ (Accessed: 2022/03/21), 2018.
[8] ISO, "ISO/IEC 24745:2011 information technology — security techniques — biometric information protection," 2011.

[9] ISO, "ISO/IEC 30136:2018 information technology — performance testing of biometric template protection schemes," 2018.
[10] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," SIAM Journal on Computing, vol.38, no.1, pp.97–139, 2008.
[11] X. Boyen, "Reusable cryptographic fuzzy extractors," The 11th ACM Conference on Computer and Communications Security (CCS2004), pp.82–91, 2004.
[12] U. Uludag, A. Ross, and A. Jain, "Biometric template selection and update: a case study in fingerprints," Pattern Recognition, vol.37, no.7, pp.1533–1542, 2004.
[13] Z. Akhtar, A. Ahmed, C. E. Erdem, and G. L. Foresti, "Biometric template update under facial aging," 2014 IEEE Symposium on Computational Intelligence in Biometrics and Identity Management (CIBIM), pp. 9–15, 2014.
[14] K.I. Chang, K.W. Bowyer, and P.J. Flynn, "An evaluation of multimodal 2d+ 3d face biometrics," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.27, no.4, pp.619–624, 2005.
[15] C. Rathgeb, T. Schlett, N. Buchmann, H. Baier, and C. Busch, "Multi-sample compression of iris images using high efficiency video coding," 2018 International Conference on Biometrics (ICB2018), pp.291–296, 2018.
[16] Y. Kaga, K. Takahashi, and T. Murakami, "Accuracy improvement of biometric authentication based on decision fusion with GLMM-based template quality estimation," IEICE IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.J96-A, no.12, pp.815–828, 2013 (in Japanese).
[17] S. Marcel, M.S. Nixon, J. Fierrez, and N. Evans, Handbook of biometric anti-spoofing: Presentation attack detection, 2019.
[18] J. Black, P. Rogaway, and T. Shrimpton, "Encryption-scheme security in the presence of key-dependent messages," International Workshop on Selected Areas in Cryptography (SAC2002), pp.62–75, 2002.
[19] T. Malkin, I. Teranishi, and M. Yung, "Key dependent message security: recent results and applications," The First ACM Conference on Data and Application Security and Privacy (CODASPY2011), pp.3–12, 2011.
[20] D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky, "Circular-secure encryption from decision diffie-hellman," Annual International Cryptology Conference (CRYPTO2008), pp.108–125, 2008.
[21] I. Haitner and T. Holenstein, "On the (im) possibility of key dependent encryption," Theory of Cryptography Conference (TCC2009), pp.202–219, 2009.
[22] B. Barak, I. Haitner, D. Hofheinz, and Y. Ishai, "Bounded key-dependent message security," 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT2010), pp.423–444, 2010.
[23] K. Takahashi, T. Matsuda, T. Murakami, G. Hanaoka, and M. Nishigaki, "Signature schemes with a fuzzy private key," International Journal of Information Security, vol.18, no.5, pp.581–617, 2019.
[24] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith, "Secure remote authentication using biometric data," 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT2005), pp.147–163, 2005.
[25] T. Murakami, T. Ohki, and K. Takahashi, "Optimal sequential fusion for multibiometric cryptosystems," Information fusion, vol.32, pp.93–108, 2016.
[26] T. Kaur and M. Kaur, "Cryptographic key generation from multimodal template using fuzzy extractor," 2017 Tenth International Conference on Contemporary Computing (IC3), pp.1–6, IEEE, 2017.



Wataru Nakamura received the Bachelor's degree in engineering and the Master's degree in information science and technology in 2015 and 2017, respectively, from the University of Tokyo, Japan. He is now with the Research & Development Group, Hitachi, Ltd. His research interests are biometrics and information security.

Kenta Takahashi received the Ph.D. in Computer Science from the University of Tokyo in 2012. He is a Principal Researcher with the Research & Development Group, Hitachi, Ltd. He is also an expert of ISO/IEC JTC1 SC37 WG5. He was a visiting associate professor with the Graduate School of Information Science and Technology, University of Tokyo from 2015 to 2020. His current research interests are biometrics, cryptography, information security and digital identity. He received several awards including the Ichimura Prize in Industry in 2021, the DOCOMO Mobile Science Award in 2016 and the IPSJ Nagao Special Researcher Award in 2015.