The Cryptographic Layer of Biometric Authentication

Keng-Yu Chen

January 13, 2025

Abstract

In this project, we focus on the cryptographic layer which is added on the top of biometric authentication for privacy reasons. We first formalize a biometric authentication scheme and propose security models for two security properties of interest: *unforgeability* and *indistinguishability*. Unforgeability refers to an adversary's ability to impersonate a user, while indistinguishability evaluates the server's knowledge of users' biometrics, related to privacy preservation. Subsequently, we analyze two existing instantiations of biometric authentication built on two cryptographic primitives: function-hiding inner product functional encryption and relational hash. Our results demonstrate conditions under which these schemes achieve security within out security model, and we propose a simple way to strengthen the system based on functional encryption by adding a digital signature in the cryptographic layer.

1 Introduction

Biometric authentication offers an error-tolerant approach to user verification. Despite its convenience, unlike traditional authentication methods, servers have to verify users' identities by comparing the similarity of enrolled and probed data instead of their equivalence. An authentication method based on comparing hashes of two templates thus fails. Additionally, unlike a user-defined password, biometrics reveal sensitive personal information and cannot be changed, raising significant privacy concerns. Furthermore, the inherent nature of biometrics data can introduce a non-negligible false positive rate. These issues make designing a biometric authentication scheme and analyzing its security challenging and highlight the importance of a rigorous study in this domain.

Previous works have demonstrated several potential cryptographic primitives that can be utilized to instantiate a biometric authentication scheme, such as function-hiding inner-product functional encryption (fh-IPFE) [Kim+16; Lee+18; Che+21; Cac+22; EM23], homomorphic encryption [Yas+13; PM21], fuzzy extractor [Boy04; Li+17], oblivious transfer [BCP12], relational hash [MR14], etc. Some of them provide non-interactive protocols in the sense that only the clients transmit enrollment and probe messages to the server before the server decides the authentication results. On the other hand, an interactive protocol allows the server to send hints or challenges to the clients during the authentication process.

In this project, we provide a general framework for analyzing a non-interactive biometric authentication scheme. We first formally define a biometric authentication scheme in Section 3, which can be split into two layers: the biometric layer and the cryptographic layer. The biometric layer accounts for collecting biometric data from users, comparing the closeness of enrolled and probed biometric templates, and deciding the authentication result. The cryptographic layer, on the other hand, is to protect users' privacy and strengthen the security of the scheme. Next in Section 4, we describe two security games to model two security notions we consider relevant to a biometric authentication scheme: the unforgeability (UF) game and indistinguishability (IND) game. The UF game models an adversary's ability to impersonate the user by offering a (possibly invalid) probe message that can result in a succesful authentication. In the UF game, we consider several options for the adversary to add flexibility to our security model. The IND game evaluates the server's knowledge of users' biometrics, where we model the adversary's ability to recognize the biometrics through enrollment and probe messages.

In Section 5 and 6, we provide analyses of instantiations of biometric authentication schemes using two primitives: function-hiding inner-product functional encryption and relational hash, repsectively. We first introduce the security of these primitives, and then we provide reductions from the UF and IND security of their instantiated authentication schemes to the security of the primitives. Our results demonstrate necessary and sufficient conditions and provide transformations for these instantiations to achieve our desired security.

2 Preliminaries

In this project, we assume

- λ is the security parameter.
- $[m]$ denotes the set of integers $\{1, 2, \cdots, m\}$.
- $\mathbb{Z}_q$ is the finite field modulo a prime number q.
- A function $f(n)$ is called *negligible* iff for any integer c, $f(n) < \frac{1}{n^c}$ for all sufficiently large n. We write it as $f(n) = \mathsf{negl}$, and we may also use negl to represent an arbitrary negligible function.
- poly is the class of polynomial funcions. We may also use poly to represent an arbitrary polynomial function.
- We write sampling a value r from a distribution $\mathcal{D}$ as $r \leftarrow\!\!\$\ \mathcal{D}$. If S is a finite set, then $r \leftarrow\!\!\$\ S$ means sampling r uniformly from S.
- The distribution $\mathcal{D}^t$ denotes t identical and independent distributions of $\mathcal{D}$.
- A PPT algorithm denotes a probabilistic polynomial time algorithm. Unless otherwise specified, all algorithms run in PPT.

We introduce two primitives to instantiate a biometric authentication scheme: function-hiding inner product functional encryption and relatioanl hash.

Definition 1 (Function-Hiding Inner Product Functional Encryption (adapted from [Kim+16])). A *function-hiding inner product functional encryption* (fh-IPFE) scheme FE for a field $\mathbb{F}$ and input length k is composed of PPT algorithms $\mathsf{FE.Setup}$, $\mathsf{FE.KeyGen}$, $\mathsf{FE.Enc}$, and $\mathsf{FE.Dec}$:

- $\mathsf{FE.Setup}(1^\lambda) \to \mathsf{msk}, \mathsf{pp}$: It outputs the public parameter pp and the master secret key msk.
- $\mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{x}) \to f_\mathbf{x}$: It generates the functional decryption key $f_\mathbf{x}$ for an input vector $\mathbf{x} \in \mathbb{F}^k$.
- $\mathsf{FE.Enc}(\mathsf{msk}, \mathsf{pp}, \mathbf{y}) \to \mathbf{c}_\mathbf{y}$: It encrypts the input vector $\mathbf{y} \in \mathbb{F}^k$ to the ciphertext $\mathbf{c}_\mathbf{y}$.
- $\mathsf{FE.Dec}(\mathsf{pp}, f_\mathbf{x}, \mathbf{c}_\mathbf{y}) \to z$: It outputs a value $z \in \mathbb{F}$ or an error symbol $\perp$.

Correctness: An fh-IPFE scheme FE is *correct* if $\forall(\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{FE.Setup}(1^\lambda)$ and $\mathbf{x}, \mathbf{y} \in \mathbb{F}^k$, we have

$$\mathsf{FE.Dec}(\mathsf{pp}, \mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{x}), \mathsf{FE.Enc}(\mathsf{msk}, \mathsf{pp}, \mathbf{y})) = \mathbf{x}\mathbf{y}^T \in \mathbb{F}.$$

Instantiation using an fh-IPFE scheme is given in Section 3.2.1.

Definition 2 (Relational Hash (adapted from [MR14])). Let R_λ be a relation over sets X_λ, Y_λ, and Z_λ. A *relational hash* scheme RH for R_λ consists of PPT algorithms $\mathsf{RH.KeyGen}$, $\mathsf{RH.HASH}_1$, $\mathsf{RH.HASH}_2$, and $\mathsf{RH.Verify}$:

- $\mathsf{RH.KeyGen}(1^\lambda) \to \mathsf{pk}$: It outputs a public hash key pk.
- $\mathsf{RH.Hash}_1(\mathsf{pk}, \mathbf{x}) \to \mathbf{h}_\mathbf{x}$: Given a hash key pk and $\mathbf{x} \in X_\lambda$, it outputs a hash $\mathbf{h}_\mathbf{x}$.
- $\mathsf{RH.Hash}_2(\mathsf{pk}, \mathbf{y}) \to \mathbf{h}_\mathbf{y}$: Given a hash key pk and $\mathbf{y} \in Y_\lambda$, it outputs a hash $\mathbf{h}_\mathbf{y}$.
- $\mathsf{RH.Verify}(\mathsf{pk}, \mathbf{h}_\mathbf{x}, \mathbf{h}_\mathbf{y}, \mathbf{z}) \to r \in \{0,1\}$: Given a hash key pk, two hashes $\mathbf{h}_\mathbf{x}$ and $\mathbf{h}_\mathbf{y}$, and $\mathbf{z} \in Z_\lambda$, it verifies whether the relation among $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$ holds.

Correctness: A relational hash scheme RH is *correct* if $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in X_\lambda \times Y_\lambda \times Z_\lambda$,

$$\Pr\left[\begin{array}{l} \mathsf{pk} \leftarrow \mathsf{RH.KeyGen}(1^\lambda) \\ \mathbf{h}_\mathbf{x} \leftarrow \mathsf{RH.Hash}_1(\mathsf{pk}, \mathbf{x}) : \mathsf{RH.Verify}(\mathsf{pk}, \mathbf{h}_\mathbf{x}, \mathbf{h}_\mathbf{y}, \mathbf{z}) = R(\mathbf{x}, \mathbf{y}, \mathbf{z}) \\ \mathbf{h}_\mathbf{y} \leftarrow \mathsf{RH.Hash}_2(\mathsf{pk}, \mathbf{y}) \end{array}\right] = 1 - \mathsf{negl}.$$

Note that Z_λ is an auxiliary input. When the relation R is over two sets $X \times Y$, we ignore Z and write $\mathsf{RH.Verify}(\mathsf{pk}, \mathbf{h}_\mathbf{x}, \mathbf{h}_\mathbf{y})$.

Instantiation using a relational hash is given in Section 3.2.2.

3 Formalization

3.1 Biometric Authentication Scheme

In this section, we formally define a biometric authentication scheme. For this, we first define how we simulate biometric distributions of users.

Assume the existence of a family $\mathbb{B}$ of biometric distributions that are efficiently samplable. We have the following interfaces for all algorithms to interact with $\mathbb{B}$.

- $\mathsf{BioSamp}()$: Generate a random distribution $\mathcal{B}$ of $\mathbb{B}$. By this we mean providing either parameters of an efficiently samplable distribution or a PPT algorithm as the sampler. For simplicity, we write $\mathcal{B} \leftarrow \mathsf{BioSamp}()$ as $\mathcal{B} \leftarrow_{\$} \mathbb{B}$.
- $\mathsf{BioDelete}(\mathcal{B})$: Delete $\mathcal{B}$ from $\mathbb{B}$. Consequently, no further access to $\mathsf{BioSamp}$ can derive $\mathcal{B}$. For simplicity, we write $\mathsf{BioDelete}(\mathcal{B})$ as $\mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}$.
- $\mathsf{TempSamp}(\mathcal{B})$: Let $\mathcal{B}$ be a biometric distribution in $\mathbb{B}$. This algorithm samples a biometric template from $\mathcal{B}$. For simplicity, we write $\mathbf{b} \leftarrow \mathsf{TempSamp}(\mathcal{B})$ as $\mathbf{b} \leftarrow_{\$} \mathcal{B}$.

Definition 3 (Biometric Authentication Scheme)**.** A *biomtric authentication shceme* Π associated with a family $\mathbb{B}$ of biometric distributions is composed of the following algorithms.

- $\mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}() \to \mathbf{b}$: Given an oracle $\mathcal{O}_\mathcal{B}$, which samples biometric data from a distribution $\mathcal{B} \in \mathbb{B}$, it outputs a biometric template $\mathbf{b}$ for enrollment.
- $\mathsf{getProbe}^{\mathcal{O}_\mathcal{B}}() \to \mathbf{b}'$: Given an oracle $\mathcal{O}_\mathcal{B}$, which samples biometric data from a distribution $\mathcal{B} \in \mathbb{B}$, it outputs a biometric template $\mathbf{b}'$ for probe.
- $\mathsf{BioCompare}(\mathbf{b}, \mathbf{b}') \to s$: Given two biometric templates $\mathbf{b}$ and $\mathbf{b}'$, it outputs a score s.
- $\mathsf{Verify}(s) \to r \in \{0, 1\}$: It is a deterministic algorithm that reads the comparison score s and determines whether this is a successful authentication ($r = 1$) or not ($r = 0$).

We also call these algorithms the *biometric layer* of Π. We will add a *cryptographic layer* on top of it in Section 3.2

Given an authentication scheme Π, we can consider its true positive rate and false positive rate.

Definition 4 (True Positive Rate)**.** For a biometric distribution $\mathcal{B} \in \mathbb{B}$ and $\mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}()$, define the *true positive rate* TP.

$$\mathsf{TP}(\mathcal{B}, \mathbf{b}) := \Pr[\mathbf{b}' \leftarrow \mathsf{getProbe}^{\mathcal{O}_\mathcal{B}}() \quad : \mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}, \mathbf{b}')) = 1]$$

$$\begin{aligned}\mathsf{TP}(\mathcal{B}) &:= \Pr\left[\begin{array}{l}\mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}() \\ \mathbf{b}' \leftarrow \mathsf{getProbe}^{\mathcal{O}_\mathcal{B}}()\end{array} : \mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}, \mathbf{b}')) = 1\right] \\ &= \mathbb{E}_{\mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}()}[\mathsf{TP}(\mathcal{B}, \mathbf{b})]\end{aligned}$$

$$\begin{aligned}\mathsf{TP} &:= \Pr\left[\begin{array}{l}\mathcal{B} \leftarrow_\$ \mathbb{B} \\ \mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}() \\ \mathbf{b}' \leftarrow \mathsf{getProbe}^{\mathcal{O}_\mathcal{B}}()\end{array} : \mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}, \mathbf{b}')) = 1\right] \\ &= \mathbb{E}_{\mathcal{B} \leftarrow_\$ \mathbb{B}}[\mathsf{TP}(\mathcal{B})]\end{aligned}$$

Definition 5 (False Positive Rate)**.** For a biometric distribution $\mathcal{B} \in \mathbb{B}, \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}$ and $\mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}()$, define the *false positive rate* FP.

$$\mathsf{FP}(\mathbf{b}) := \Pr\left[\begin{array}{l}\mathcal{B}' \leftarrow_\$ \mathbb{B} \\ \mathbf{b}' \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}'}}()\end{array} : \mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}, \mathbf{b}')) = 1\right]$$

$$\begin{aligned}\mathsf{FP}(\mathcal{B}) &:= \Pr\left[\begin{array}{l}\mathcal{B}' \leftarrow_\$ \mathbb{B} \\ \mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}() \\ \mathbf{b}' \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}'}}()\end{array} : \mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}, \mathbf{b}')) = 1\right] \\ &= \mathbb{E}_{\mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}()}[\mathsf{FP}(\mathbf{b})]\end{aligned}$$

$$\begin{aligned}\mathsf{FP} &:= \Pr\left[\begin{array}{l}\mathcal{B} \leftarrow_\$ \mathbb{B}, \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}, \mathcal{B}' \leftarrow_\$ \mathbb{B} \\ \mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}() \\ \mathbf{b} \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}'}}()\end{array} : \mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}, \mathbf{b}')) = 1\right] \\ &= \mathbb{E}_{\mathcal{B} \leftarrow_\$ \mathbb{B}}[\mathsf{FP}(\mathcal{B})]\end{aligned}$$

Ideally, we hope TP to be 1, and $\mathsf{FP}(\mathcal{B})$ to be negligible for any $\mathcal{B} \in \mathbb{B}$. However, due to the inherent nature of biometrics, there might be a nonzero false negative rate $1 - \mathsf{TP} > 0$ and a non-negligible $\mathsf{FP}(\mathcal{B})$. Our security model and analysis also take these possibilities into consideration.

3.2 Cryptographic Layer

In this work, we add a cryptographic layer on top of Π to protect privacy of users. The cryptographic layer includes the following algorithms.

- $\mathsf{Setup}(1^\lambda) \rightarrow \mathsf{esk}, \mathsf{psk}, \mathsf{csk}$: It outputs the enrollment secret key esk, probe secret key psk, and compare secret key csk.

- $\mathsf{Enroll}(\mathsf{esk}, \mathbf{b}) \to \mathbf{c_x}$: On input a biometric template $\mathbf{b}$, it encodes it into a vector $\mathbf{x}$ and outputs the enrollment message $\mathbf{c_x}$.
- $\mathsf{Probe}(\mathsf{psk}, \mathbf{b}') \to \mathbf{c_y}$: On input a biometric template $\mathbf{b}'$, it encodes it into a vector $\mathbf{y}$ and outputs the probe message $\mathbf{c_y}$.
- $\mathsf{Compare}(\mathsf{csk}, \mathbf{c_x}, \mathbf{c_y}) \to s$: It compares the enrollment message $\mathbf{c_x}$ and probe message $\mathbf{c_y}$ and outputs a score s.

Correctness: An authentication scheme Π is *correct* if for any biometric distributions $\mathcal{B}$ and $\mathcal{B}'$, let $\mathsf{esk}, \mathsf{psk}, \mathsf{csk} \leftarrow \mathsf{Setup}(1^\lambda)$, $\mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}()$, $\mathbf{b}' \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}'}}()$, $\mathbf{c_x} \leftarrow \mathsf{Enroll}(\mathsf{esk}, \mathbf{b})$, $\mathbf{c_y} \leftarrow \mathsf{Probe}(\mathsf{psk}, \mathbf{b}')$. Then

$$\Pr\left[\mathsf{Compare}(\mathsf{csk}, \mathbf{c_x}, \mathbf{c_y}) = \mathsf{BioCompare}(\mathbf{b}, \mathbf{b}')\right] = 1 - \mathsf{negl}.$$

In a real-world biometric system, these algorithms may be run by different parties such as a biometric scanner, a user's secure hardware, a trusted authority that issues keys, and the server.

Now, we provide two instantiations of a biometric authentication scheme with the cryptographic layer.

3.2.1 Instantiation with an fh-IPFE Scheme

Let $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.KeyGen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ be an fh-IPFE scheme we defined in Definition 1. Following [EM23], we can instantiate a biometric authentication scheme using FE with the distance metric the Euclidean distance. Let $\mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}()$ and $\mathsf{getProbe}^{\mathcal{O}_\mathcal{B}}()$ both output vectors in $\{0, 1, \cdots, m\}^k$ for all biometric distributions $\mathcal{B} \in \mathbb{B}$. For a pre-defined real number $\tau \geq 0$, define

$$\mathsf{BioCompare}(\mathbf{b}, \mathbf{b}') \to \|\mathbf{b} - \mathbf{b}'\|^2 \quad \text{and} \quad \mathsf{Verify}(s) \to \begin{cases} 1 & \text{if } \sqrt{s} \leq \tau \\ 0 & \text{if } \sqrt{s} > \tau \end{cases}.$$

Now, let the associated field of FE be $\mathbb{Z}_q$, where q is a prime number larger than the maximum possible Euclidean distance $m^2 \cdot k$. The scheme is instantiated as follows.

- $\mathsf{Setup}(1^\lambda)$: It calls $\mathsf{FE.Setup}(1^\lambda) \to \mathsf{msk}, \mathsf{pp}$ and outputs $\mathsf{esk} \leftarrow (\mathsf{msk}, \mathsf{pp})$, $\mathsf{psk} \leftarrow (\mathsf{msk}, \mathsf{pp})$ and $\mathsf{csk} \leftarrow \mathsf{pp}$.
- $\mathsf{Enroll}(\mathsf{esk}, \mathbf{b})$: On input a template vector $\mathbf{b} = (b_1, b_2, \cdots, b_k)$, the algorithm first encodes it as $\mathbf{x} = (x_1, x_2, \cdots, x_{k+2}) = (b_1, b_2, \cdots, b_k, 1, \|\mathbf{b}\|^2)$. Next, it calls $\mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{x}) \to f_\mathbf{x}$ and outputs $\mathbf{c_x} \leftarrow f_\mathbf{x}$.
- $\mathsf{Probe}(\mathsf{psk}, \mathbf{b}')$: On input a template vector $\mathbf{b}' = (b'_1, b'_2, \cdots, b'_k)$, the algorithm first encodes it as $\mathbf{y} = (y_1, y_2, \cdots, y_{k+2}) = (-2b'_1, -2b'_2, \cdots, -2b'_k, \|\mathbf{b}'\|^2, 1)$. Next, it calls $\mathsf{FE.Enc}(\mathsf{msk}, \mathsf{pp}, \mathbf{y}) \to \mathbf{c_y}$ and outputs $\mathbf{c_y}$.
- $\mathsf{Compare}(\mathsf{csk}, \mathbf{c_x}, \mathbf{c_y})$: It calls $\mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c_x}, \mathbf{c_y}) \to s$ and outputs the value s.

By the correctness of the functional encryption scheme FE, we have

$$s = \mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c_x}, \mathbf{c_y}) = \mathbf{x}\mathbf{y}^T = \sum_{i=1}^{k} -2b_i b_i' + \|\mathbf{b}\|^2 + \|\mathbf{b}'\|^2 = \|\mathbf{b} - \mathbf{b}'\|^2.$$

which is equal to $\mathsf{BioCompare}(\mathbf{b}, \mathbf{b}')$. Therefore, if two templates $\mathbf{b}$ and $\mathbf{b}'$ are close enough such that $\|\mathbf{b} - \mathbf{b}'\| \leq \tau$, the scheme results in $r = 1$, a successful authentication.

Instantiated with an fh-IPFE scheme in this way, the comparison secret key csk is public, and the enrollment secret key esk and probe secret key psk are the same. Anyone with access to the enrollment message $\mathbf{c_x}$ and either esk or psk can probe any (invalidly encoded) $\mathbf{y}' \in \mathbb{Z}_q^{k+2}$ and find $\mathbf{x}\mathbf{y}'^T$ to get partial or full information about the biometric template $\mathbf{b}$. Even if the adversary has no esk or psk, if it can sample ciphertexts $\mathbf{c_y}$ corresponding to some unknown random vectors $\mathbf{y}$, and if the field size q is not large enough, it can also find a forged $\mathbf{c}_{\mathbf{y}^*}$ such that $\mathbf{x}\mathbf{y}^{*T} \leq \tau$ with a non-negligible probability to impersonate the user by sampling many times offline.

A security analysis of this instantiation in our security model is given in Section 5.

3.2.2 Instantiation with a Relational Hash Scheme

Let $\mathsf{RH} = (\mathsf{RH.KeyGen}, \mathsf{RH.Hash}_1, \mathsf{RH.Hash}_2, \mathsf{RH.Verify})$ be a relational hash scheme we defined in Definition 2 for the relation R^τ of Hamming distance proximity parametrized by a constant τ.

$$R^\tau = \{(\mathbf{x}, \mathbf{y}) \mid \mathsf{HD}(\mathbf{x}, \mathbf{y}) \leq \tau \wedge \mathbf{x}, \mathbf{y} \in \{0,1\}^k\}$$

Note that here we ignore the third parameter Z. Let $\mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}()$ and $\mathsf{getProbe}^{\mathcal{O}_\mathcal{B}}()$ both output vectors in $\{0,1\}^k$ for all biometric distributions $\mathcal{B} \in \mathbb{B}$, and let

$$\mathsf{BioCompare}(\mathbf{b}, \mathbf{b}') \to \begin{cases} 1 & \text{if } (\mathbf{b}, \mathbf{b}') \in R^\tau \\ 0 & \text{if } (\mathbf{b}, \mathbf{b}') \notin R^\tau \end{cases} \quad \text{and} \quad \mathsf{Verify}(s) \to s.$$

Following [MR14], we can instantiate a biometric authentication scheme using RH. Let the biometric distribution $\mathcal{B} \subseteq \{0,1\}^k$.

- $\mathsf{Setup}(1^\lambda)$: It calls $\mathsf{RH.KeyGen}(1^\lambda) \to \mathsf{pk}$ and outputs $\mathsf{esk} \leftarrow \mathsf{pk}$, $\mathsf{psk} \leftarrow \mathsf{pk}$, and $\mathsf{csk} \leftarrow \mathsf{pk}$.

- $\mathsf{Enroll}(\mathsf{esk}, \mathbf{b})$: Let $\mathbf{x} \leftarrow \mathbf{b}$. It calls $\mathsf{RH.Hash}_1(\mathsf{pk}, \mathbf{x}) \to \mathbf{h_x}$ and outputs $\mathbf{c_x} \leftarrow \mathbf{h_x}$.

- $\mathsf{Probe}(\mathsf{psk}, \mathbf{b}')$: Let $\mathbf{y} \leftarrow \mathbf{b}$. It calls $\mathsf{RH.Hash}_2(\mathsf{pk}, \mathbf{y}) \to \mathbf{h_y}$ and outputs $\mathbf{c_y} \leftarrow \mathbf{h_y}$.

- $\mathsf{Compare}(\mathsf{csk}, \mathbf{c_x}, \mathbf{c_y})$: It calls $\mathsf{RH.Verify}(\mathsf{pk}, \mathbf{h_x}, \mathbf{h_y}) \to s$ and outputs the value s.

By the correctness of the relational hash scheme RH, we have (except for a negligible probability),

$$r = 1 \Leftrightarrow (\mathbf{x}, \mathbf{y}) = (\mathbf{b}, \mathbf{b}') \in R^\tau \Leftrightarrow \mathsf{HD}(\mathbf{b}, \mathbf{b}') \leq \tau$$

A security analysis of this instantiation in our security model is given in Section 6.

4 Security Games

In this section, we discuss two security notions of a biometric authentication scheme: *unforgeability* and *indistinguishability*.

4.1 Unforgeability

To describe the unforgeability of an authentication scheme, we model the ability of an adversary who tries to impersonate a user. The adversary $\mathcal{A}$ is given auxiliary information option that depends on our threat model and tries to find a valid probe message $\tilde{\mathbf{z}}$. The whole game $\mathsf{UF}_{\Pi,\mathbb{B},\mathsf{option}}$ is defined in Algorithm 1.

Algorithm 1 $\mathsf{UF}_{\Pi,\mathbb{B},\mathsf{option}}(\mathcal{A})$

```
1: B ←$ 𝔹, 𝔹 ← 𝔹 \ B
2: esk, psk, csk ← Setup(1^λ)
3: b ← getEnroll^{O_B}()
4: c_x ← Enroll(esk, b)
5: z̃ ← A(option)
6: if z̃ is equal to any output of O_Probe then
7:     return 0
8: end if
9: s ← Compare(csk, c_x, z̃)
10: return Verify(s)
```

The auxiliary information option can be nothing or include $\mathbf{c_x}, \mathsf{esk}, \mathsf{psk}, \mathsf{csk}$ or the following oracles:

- $\mathcal{O}_{\mathcal{B}}$: It outputs a biometric sample $\mathbf{b} \leftarrow\$ \mathcal{B}$. This oracle and psk should not be given at the same time; otherwise, there exists a trivial attack with a winning rate TP by returning $\mathsf{Probe}(\mathsf{psk}, \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}}}())$.
- $\mathcal{O}_{\mathsf{Enroll}}(\mathsf{esk}, \cdot)$: On input $\mathbf{b}'$, it outputs the enrollment message $\mathsf{Enroll}(\mathsf{esk}, \mathbf{b}')$.
- $\mathcal{O}_{\mathsf{Probe}}(\mathsf{psk}, \cdot)$: On input $\mathbf{b}'$, it outputs the probe message $\mathsf{Probe}(\mathsf{psk}, \mathbf{b}')$. If this oracle is given, we require the adversary to return a $\tilde{\mathbf{z}}$ that is not equal to any previous answer of $\mathcal{O}_{\mathsf{Probe}}$.
- $\mathcal{O}_{\mathsf{log}}(\mathsf{csk}, \mathbf{c_x}, \cdot)$: On input $\mathbf{b}'$, it first computes $\mathbf{c_z} \leftarrow \mathsf{Probe}(\mathsf{psk}, \mathbf{b}')$ and outputs $\mathsf{Verify}(\mathsf{Compare}(\mathsf{csk}, \mathbf{c_x}, \mathbf{c_z}))$.

- $\mathcal{O}'_{\mathsf{Enroll}}(\cdot)$: On input esk', it first samples $\mathbf{b}' \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}}}()$ and outputs $\mathsf{Enroll}(\mathsf{esk}', \mathbf{b}')$. This oracle is only useful when option does not include $\mathcal{O}_{\mathcal{B}}$.
- $\mathcal{O}'_{\mathsf{Probe}}(\cdot)$: On input psk', it first samples $\mathbf{b}' \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}}}()$ and outputs $\mathsf{Probe}(\mathsf{psk}', \mathbf{b}')$. This oracle is only useful when option does not include $\mathcal{O}_{\mathcal{B}}$, and this oracle and psk should not be given at the same time; otherwise, there exists a trivial attack with a winning rate TP by returning $\mathcal{O}'_{\mathsf{Probe}}(\mathsf{psk})$.

The requirement that the adversary should return a $\tilde{\mathbf{z}}$ that is not equal to any previous answer of $\mathcal{O}_{\mathsf{Probe}}$ is to prevent a trivial attack that leverages TP or FP when it is non-negligible. If option includes $\mathcal{O}_{\mathcal{B}}$ and either psk or $\mathcal{O}_{\mathsf{Probe}}$, the adversary can enjoy a winning rate TP. Therefore, we rule out the case that option includes both psk and $\mathcal{O}_{\mathcal{B}}$, and we forbid the adversary to return what $\mathcal{O}_{\mathsf{Probe}}$ returns. If option has only psk or $\mathcal{O}_{\mathsf{Probe}}$, the UF adversary $\mathcal{A}$ in Algorithm 2 can still enjoy a winning rate FP, if we place no restriction on the adversary's answer. Therefore, we only consider psk in option when FP is non-negligible, and we restrict the adversary's answer when $\mathcal{O}_{\mathsf{Probe}}$ is given.

Algorithm 2 $\mathcal{A}(\mathsf{psk})$ (or $\mathcal{A}^{\mathcal{O}_{\mathsf{Probe}}}$)

1: $\mathcal{B}' \leftarrow_{\$} \mathbb{B}$
2: $\mathbf{b}' \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}'}}()$
3: $\mathbf{c_y} \leftarrow \mathsf{Probe}(\mathsf{psk}, \mathbf{b}')$ $\triangleright$ or $\mathbf{c_y} \leftarrow \mathcal{O}_{\mathsf{Probe}}(\mathbf{b}')$
4: **return** $\mathbf{c_y}$

We define the advantage of an adversary $\mathcal{A}$ in the $\mathsf{UF}_{\Pi,\mathbb{B},\mathsf{option}}$ game of a scheme Π associated with a family $\mathbb{B}$ of distributions as

$$\mathbf{Adv}^{\mathsf{UF}}_{\Pi,\mathbb{B},\mathcal{A},\mathsf{option}} := \Pr[\mathsf{UF}_{\Pi,\mathbb{B},\mathsf{option}}(\mathcal{A}) \to 1]$$

An authentication scheme Π associated with a family $\mathbb{B}$ of distributions is called *option-unforgeable* (option-UF) if for any PPT adversary $\mathcal{A}$,

$$\mathbf{Adv}^{\mathsf{UF}}_{\Pi,\mathbb{B},\mathcal{A},\mathsf{option}} = \mathsf{negl}.$$

For the rest of this work, if the scheme Π, the family $\mathbb{B}$ of distributions, and the auxiliary information option are clear from context, we omit the subscript and write the game as $\mathsf{UF}(\mathcal{A})$. This abbreviation also holds for all other games.

UF Security with Digital Signature We note that we can achieve UF security by adding a digital signature scheme. Given any authentication scheme Π and an sEUF-CMA digital signature scheme $\mathsf{Sig} = (\mathsf{Sig.KeyGen}, \mathsf{Sig.Sign}, \mathsf{Sig.Verify})$, consider the following scheme Π'.

- $\mathsf{Setup}'(1^\lambda)$: Run $(\mathsf{esk}, \mathsf{psk}, \mathsf{csk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and $(\mathsf{sk_{Sig}}, \mathsf{pk_{Sig}}) \leftarrow \mathsf{Sig.KeyGen}(1^\lambda)$. Output $\mathsf{esk}' \leftarrow \mathsf{esk}$, $\mathsf{psk}' \leftarrow (\mathsf{psk}, \mathsf{sk_{Sig}})$, $\mathsf{csk}' \leftarrow \mathsf{csk}$.

- $\mathsf{Probe}'(\mathsf{psk}', \mathbf{b}')$: Run $\mathbf{c_y} \leftarrow \mathsf{Probe}(\mathsf{psk}, \mathbf{b}')$ and $\sigma \leftarrow \mathsf{Sig.Sign}(\mathsf{sk}_{\mathsf{Sig}}, \mathbf{c_y})$. Output $\mathbf{c_y}' \leftarrow (\mathbf{c_y}, \sigma)$.
- $\mathsf{Compare}'(\mathsf{csk}, \mathbf{c_x}, \mathbf{c_y}')$: If $\mathsf{Sig.Verify}(\mathsf{pk}_{\mathsf{Sig}}, \mathbf{c_y}, \sigma) = 1$, output $\mathsf{Compare}(\mathsf{csk}, \mathbf{c_x}, \mathbf{c_y})$; otherwise, output $\perp$.

An $\mathsf{UF}_{\mathsf{option}}$ adversary has to forge a signature σ to win the game, so the scheme is option-UF for any option that does not include psk.

Theorem 1. *Let* $\mathsf{option} = \{\mathbf{c_x}, \mathsf{esk}, \mathsf{csk}, \mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Probe}}\}$. *For any authentication scheme* Π, Π' *is* option*-UF.*

4.2 Indistinguishability

In the game of indistinguishability, we model the ability of an authentication server who tries to identify the user, which describes the privacy leakage of the scheme. The adversary $\mathcal{A}$ is given oracles to two biometric distributions $\mathcal{B}^{(0)}$ and $\mathcal{B}^{(1)}$, the comparison key csk, an enrollment message $\mathbf{c_x}$, and a list of t probe messages $\{\mathbf{c_y}^{(i)}\}_{i=1}^{t}$. It tries to guess from either $\mathcal{B}^{(0)}$ or $\mathcal{B}^{(1)}$ these messages are generated. The whole game is defined in Algorithm 3.

Algorithm 3 $\mathsf{IND}_{\Pi,\mathbb{B}}(\mathcal{A})$

1: $b \leftarrow_{\$} \{0, 1\}$
2: $\mathcal{B}^{(0)} \leftarrow_{\$} \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}^{(0)}$
3: $\mathcal{B}^{(1)} \leftarrow_{\$} \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}^{(1)}$
4: $\mathsf{esk}, \mathsf{psk}, \mathsf{csk} \leftarrow \mathsf{Setup}(1^\lambda)$
5: $\mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}^{(b)}}}()$
6: $\mathbf{c_x} \leftarrow \mathsf{Enroll}(\mathsf{esk}, \mathbf{b})$
7: **for** $i = 1$ to t **do**
8: $\quad \mathbf{b}'^{(i)} \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}^{(b)}}}()$
9: $\quad \mathbf{c_y}^{(i)} \leftarrow \mathsf{Probe}(\mathsf{psk}, \mathbf{b}'^{(i)})$
10: **end for**
11: $\tilde{b} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{B}^{(0)}}, \mathcal{O}_{\mathcal{B}^{(1)}}}(\mathsf{csk}, \mathbf{c_x}, \{\mathbf{c_y}^{(i)}\}_{i=1}^{t})$
12: **return** $1_{\tilde{b}=b}$

We define the advantage of an adversary $\mathcal{A}$ in the IND game of a scheme Π associated with a family of distributions $\mathbb{B}$ as

$$\mathbf{Adv}^{\mathsf{IND}}_{\Pi,\mathbb{B},\mathcal{A}} := \left| \Pr[\mathsf{IND}_{\Pi}(\mathcal{A}) \to 1] - \frac{1}{2} \right|.$$

An authentication scheme Π associated with a family $\mathbb{B}$ of distributions is called *indistinguishable (IND)* if for any PPT adversary $\mathcal{A}$,

$$\mathbf{Adv}^{\mathsf{IND}}_{\Pi,\mathbb{B},\mathcal{A}} = \mathsf{negl}.$$

Necessity of IND Security Consider the following authentication scheme for any biometric layer. Let $\mathsf{esk} = \mathsf{psk} = \mathsf{csk}$ be empty strings and

$$\mathsf{Enroll}(\mathsf{esk}, \mathbf{b}) \to \mathbf{b}, \quad \mathsf{Probe}(\mathsf{psk}, \mathbf{b}') \to \mathbf{b}'$$
$$\mathsf{Compare}(\mathsf{csk}, \mathbf{b}, \mathbf{b}') = \mathsf{BioCompare}(\mathbf{b}, \mathbf{b}')$$

By the transformation using an sEUF-CMA digital signature scheme we introduced in Section 4.1, we can obtain an authentication scheme Π' that is option-UF for any option that does not include psk. However, the enrollment and probe messages leak biometric vectors $\mathbf{b}$ and $\mathbf{b}'$ and compromise privacy. Obviously, this scheme is not IND, and we use this example emphasize the necessity of the game of indistinguishability.

IND Security for a Particular Biometric Layer Let $\mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}(), \mathsf{getProbe}^{\mathcal{O}_\mathcal{B}}()$ be such that

$$\mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}() \to \mathbf{b}^* + \mathcal{E}_{\mathsf{Enroll}} \quad \text{and} \quad \mathsf{getProbe}^{\mathcal{O}_\mathcal{B}}() \to \mathbf{b}^* + \mathcal{E}_{\mathsf{Probe}}$$

where $\mathbf{b}^* \in \{0,1\}^k$ is a fixed vector only dependent on $\mathcal{B}$, and $\mathcal{E}_{\mathsf{Enroll}}, \mathcal{E}_{\mathsf{Probe}} \subseteq \{0,1\}^k$ are some *error distributions* independent of $\mathcal{B}$. Let $\mathsf{BioCompare}(\mathbf{b}, \mathbf{b}') \to 1_{\mathsf{HD}(\mathbf{b},\mathbf{b}') \leq \tau}$. Then

$$\mathsf{TP} = \Pr[\mathsf{HW}(\mathbf{b}^* + \mathcal{E}_{\mathsf{Enroll}} + \mathbf{b}^* + \mathcal{E}_{\mathsf{Probe}}) \leq \tau] = \Pr[\mathsf{HW}(\mathcal{E}_{\mathsf{Enroll}} + \mathcal{E}_{\mathsf{Probe}}) \leq \tau]$$

We note that previous works such as [Boy04; MR14] model biometric template vectors in a similar way.

Now, for this biometric layer, we can construct a simple but IND secure authentication scheme Π with the following cryptographic layer.

- $\mathsf{Setup}(1^\lambda)$: Sample $\mathbf{r} \leftarrow_\$ \{0,1\}^k$. Output $\mathsf{esk} = \mathsf{psk} \leftarrow \mathbf{r}$, $\mathsf{csk} \leftarrow \epsilon$.
- $\mathsf{Enroll}(\mathsf{esk}, \mathbf{b})$: Output $\mathbf{b} + \mathbf{r}$.
- $\mathsf{Probe}(\mathsf{psk}, \mathbf{b}')$: Output $\mathbf{b}' + \mathbf{r}$.
- $\mathsf{Compare}(\mathsf{csk}, \mathbf{c_x}, \mathbf{c_y})$: If $\mathsf{HD}(\mathbf{c_x}, \mathbf{c_y}) \leq \tau$, return 1; otherwise, return 0.

The correctness of Π holds by

$$\mathsf{HD}(\mathbf{c_x}, \mathbf{c_y}) = \mathsf{HW}(\mathbf{b} + \mathbf{r} + \mathbf{b}' + \mathbf{r}) = \mathsf{HW}(\mathbf{b} + \mathbf{b}') = \mathsf{BioCompare}(\mathbf{b}, \mathbf{b}').$$

Theorem 2. *The authentication scheme Π is IND.*

Proof. Let $\mathbf{b}_0^*$ and $\mathbf{b}_1^*$ be the fixed vectors of $\mathcal{B}^{(0)}$ and $\mathcal{B}^{(1)}$ in the IND game, respectively. For any $\mathbf{v}, \mathbf{v}^{(1)}, \cdots, \mathbf{v}^{(t)}$,

$$\begin{aligned}
&\Pr[\mathbf{c_x} = \mathbf{v},\ \mathbf{c_y}^{(1)} = \mathbf{v}^{(1)},\ \cdots,\ \mathbf{c_y}^{(t)} = \mathbf{v}^{(t)} \mid b = 0,\ \mathbf{b}_0^*,\ \mathbf{b}_1^*] \\
&= \Pr[\mathbf{b}_0^* + \mathcal{E}_{\mathsf{Enroll}} + \mathbf{r} = \mathbf{v},\ \mathbf{b}_0^* + \mathcal{E}_{\mathsf{Probe}} + \mathbf{r} = \mathbf{v}^{(1)},\ \cdots,\ \mathbf{b}_0^* + \mathcal{E}_{\mathsf{Probe}} + \mathbf{r} = \mathbf{v}^{(t)} \mid \mathbf{b}_0^*,\ \mathbf{b}_1^*] \\
&= \Pr[\mathbf{r} = \mathbf{v} - \mathbf{b}_0^* - \mathcal{E}_{\mathsf{Enroll}} = \mathbf{v}^{(1)} - \mathbf{b}_0^* - \mathcal{E}_{\mathsf{Probe}} = \cdots = \mathbf{v}^{(t)} - \mathbf{b}_0^* - \mathcal{E}_{\mathsf{Probe}} \mid \mathbf{b}_0^*,\ \mathbf{b}_1^*] \\
&= \Pr[\mathbf{r} = \mathbf{v} - \mathbf{b}_1^* - \mathcal{E}_{\mathsf{Enroll}} = \mathbf{v}^{(1)} - \mathbf{b}_1^* - \mathcal{E}_{\mathsf{Probe}} = \cdots = \mathbf{v}^{(t)} - \mathbf{b}_1^* - \mathcal{E}_{\mathsf{Probe}} \mid \mathbf{b}_0^*,\ \mathbf{b}_1^*] \\
&= \Pr[\mathbf{b}_1^* + \mathcal{E}_{\mathsf{Enroll}} + \mathbf{r} = \mathbf{v},\ \mathbf{b}_1^* + \mathcal{E}_{\mathsf{Probe}} + \mathbf{r} = \mathbf{v}^{(1)},\ \cdots,\ \mathbf{b}_1^* + \mathcal{E}_{\mathsf{Probe}} + \mathbf{r} = \mathbf{v}^{(t)} \mid \mathbf{b}_0^*,\ \mathbf{b}_1^*] \\
&= \Pr[\mathbf{c_x} = \mathbf{v},\ \mathbf{c_y}^{(1)} = \mathbf{v}^{(1)},\ \cdots,\ \mathbf{c_y}^{(t)} = \mathbf{v}^{(t)} \mid b = 1,\ \mathbf{b}_0^*,\ \mathbf{b}_1^*]
\end{aligned}$$

Hence, the adversary cannot distinguish between $\mathbf{c_x}, \mathbf{c_y}^{(1)}, \cdots, \mathbf{c_y}^{(t)}$ generated from $\mathcal{B}^{(0)}$ and $\mathcal{B}^{(1)}$.

$\square$

5 Security Analysis: fh-IPFE-based Instantiation

Let Π be an authentication scheme instantiated by an fh-IPFE scheme FE as in Section 3.2.1. We discuss the UF and IND security of Π in this section. For this, we first define two security notions of FE.

5.1 fh-IND Security of FE

Given an fh-IPFE scheme FE, we define the $\mathsf{fh\text{-}IND}$ game [Kim+16] in Algorithm 4.

Algorithm 4 $\mathsf{fh\text{-}IND}_{\mathsf{FE}}(\mathcal{A})$

1: $b \leftarrow_\$ \{0, 1\}$
2: $\mathsf{msk}, \mathsf{pp} \leftarrow \mathsf{FE.Setup}(1^\lambda)$
3: $\tilde{b} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}, \mathcal{O}_{\mathsf{Enc}}}(\mathsf{pp})$
4: **return** $1_{\tilde{b}=b}$

- $\mathcal{O}_{\mathsf{KeyGen}}(\cdot, \cdot)$: On input pair $(\mathbf{x}^{(0)}, \mathbf{x}^{(1)})$, it outputs $\mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{x}^{(b)})$.
- $\mathcal{O}_{\mathsf{Enc}}(\cdot, \cdot)$: On input pair $(\mathbf{y}^{(0)}, \mathbf{y}^{(1)})$, it outputs $\mathsf{FE.Enc}(\mathsf{msk}, \mathsf{pp}, \mathbf{y}^{(b)})$.

To avoid trivial attacks, we consider *admissible adversaries*.

Definition 6 (Admissible Adversary)**.** Let $\mathcal{A}$ be an adversary in an $\mathsf{fh\text{-}IND}$ game, and let $(\mathbf{x}_1^{(0)}, \mathbf{x}_1^{(1)}), \cdots, (\mathbf{x}_{Q_K}^{(0)}, \mathbf{x}_{Q_K}^{(1)})$ be its queries to $\mathcal{O}_{\mathsf{KeyGen}}$ and $(\mathbf{y}_1^{(0)}, \mathbf{y}_1^{(1)}), \cdots, (\mathbf{y}_{Q_E}^{(0)}, \mathbf{y}_{Q_E}^{(1)})$ be its queries to $\mathcal{O}_{\mathsf{Enc}}$. We say $\mathcal{A}$ is *admissible* if $\forall i \in [Q_K], \forall j \in [Q_E]$,

$$\mathbf{x}_i^{(0)} {\mathbf{y}_j^{(0)}}^T = \mathbf{x}_i^{(1)} {\mathbf{y}_j^{(1)}}^T$$

Definition 7 (fh-IND Security)**.** An fh-IPFE scheme FE is called fh-IND secure if for any admissible adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in the $\mathsf{fh\text{-}IND}$ game in Algorithm 4 is

$$\mathbf{Adv}_{\mathsf{FE},\mathcal{A}}^{\mathsf{fh\text{-}IND}} := \left| \Pr[\mathsf{fh\text{-}IND}_{\mathsf{FE}}(\mathcal{A}) \to 1] - \frac{1}{2} \right| = \mathsf{negl}.$$

We note that fh-IND security is a standard notion for an fh-IPFE, and constructions in [DDM15; TAO16; Kim+16] are proven fh-IND. However, fh-IND security may not be sufficient for the UF security of the instantiation in Section 3.2.1.

Theorem 3. *An instantiation Π using the construction in [Kim+16] is not option-UF for any option.*

We recall the construction in [Kim+16] in Appendix A.

Proof. Let $\mathcal{A}$ be a UF game adversary that returns $(K_1, K_2) = (1, (1, \cdots, 1))$. Then, in the decryption,

$$D_1 = e(g_1, g_2)^0 = 1 \quad \text{and} \quad D_2 = e(g_1, g_2)^0 = 1$$

As $D_1^0 = D_2$, the decryption returns 0 and let the adversary win the game with probability 1. □

5.2 RUF Security of FE

We also define the $\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O},\gamma}$ game in Algorithm 5 for a real number γ.

Algorithm 5 $\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O},\gamma}(\mathcal{A})$

1: $\mathbf{r} \leftarrow_{\$} \mathbb{F}^k$
2: $\mathsf{msk}, \mathsf{pp} \leftarrow \mathsf{FE.Setup}(1^\lambda)$
3: $\mathbf{c} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{r})$
4: $\tilde{\mathbf{z}} \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}, \mathbf{c})$
5: **if** $\tilde{\mathbf{z}}$ is equal to any output of $\mathcal{O}'_{\mathsf{Enc}}$ **then**
6:     **return** 0
7: **end if**
8: $s \leftarrow \mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c}, \tilde{\mathbf{z}})$
9: **return** $1_{s \leq \gamma}$

The oracle $\mathcal{O}$ can be nothing or include the following options based on the threat model.

- $\mathcal{O}'_{\mathsf{KeyGen}}(\cdot)$: On input $\mathbf{x}'$, it outputs $\mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{x}')$.
- $\mathcal{O}'_{\mathsf{Enc}}(\cdot)$: On input $\mathbf{y}'$, it outputs $\mathsf{FE.Enc}(\mathsf{msk}, \mathsf{pp}, \mathbf{y}')$. The adversary is required to return $\tilde{\mathbf{z}}$ that is not equal to any output of this oracle.

Definition 8 (RUF Security)**.** An fh-IPFE scheme FE is called $\mathcal{O}$-RUF secure for a real number γ if for any adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in the $\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O},\gamma}$ game in Algorithm 5 is

$$\mathbf{Adv}_{\mathsf{FE},\mathcal{A}}^{\mathsf{RUF},\mathcal{O},\gamma} := \Pr[\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O},\gamma}(\mathcal{A}) \to 1] = \mathsf{negl}.$$

We say FE is RUF secure if it is $\{\mathcal{O}'_{\mathsf{KeyGen}}, \mathcal{O}'_{\mathsf{Enc}}\}$-RUF secure.

5.2.1 Achievability of RUF Security

We note that RUF security is a new security notion of fh-IPFE. In this section, we provide two theorems to obtain an $\mathcal{O}'_{\mathsf{KeyGen}}$-RUF and an RUF scheme, respectively.

Assumption 1. Let $\mathbf{x} \in \mathbb{F}^k, \mathbf{c} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{x})$. Assume that $\mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c}, \mathbf{z})$ only returns when $\mathbf{z}$ corresponds to a *nonzero* vector $\mathbf{v} \in \mathbb{F}^k$. That is, assume that for any $\mathbf{z}$, there can only be two possibilities.

- There exists a vector $\mathbf{v} \in \mathbb{F}^k \backslash \{\mathbf{0}\}$ such that for any $\mathbf{x} \in \mathbb{F}^k, \mathbf{c} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{x})$, and $\mathbf{c}_\mathbf{v} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{v})$,
$$\mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c}, \mathbf{z}) = \mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c}, \mathbf{c}_\mathbf{v}).$$
- For any $\mathbf{x} \in \mathbb{F}^k$ and $\mathbf{c} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{x})$, $\mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c}, \mathbf{z}) \to \perp$.

Note that this implies FE rejects zero vector $\mathbf{0}$ as the input of $\mathsf{FE.Enc}$.

Theorem 4. *Given Assumption 1. If FE is fh-IND, then FE is $\mathcal{O}'_{\mathsf{KeyGen}}$-RUF for any $\gamma \leq \|\mathbb{F}\|$.*

Proof. Given an adversary $\mathcal{A}$ in the $\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{KeyGen}}, \gamma}$ game for any $\gamma < \|\mathbb{F}\|$. Let t be an integer, consider the reduction adversary $\mathcal{R}$ in Algorithm 6 which plays the $\mathsf{fh\text{-}IND}$ game. $\mathcal{R}$ simulates $\mathcal{O}'_{\mathsf{KeyGen}}(\mathbf{x}')$ by $\mathcal{O}_{\mathsf{KeyGen}}(\mathbf{x}', \mathbf{x}')$. If there exists an $s_i \neq \perp$ in Line 7, by Assumption 1, let $\tilde{\mathbf{z}}$ correspond to a vector $\tilde{\mathbf{v}}$.

Algorithm 6 $\mathcal{R}^{\mathcal{O}_{\mathsf{KeyGen}}, \mathcal{O}_{\mathsf{Enc}}}(\mathsf{pp})$

1: $\mathbf{r}^{(0)}, \mathbf{r}^{(1)} \leftarrow_\$ \mathbb{F}^k$
2: $\mathbf{c} \leftarrow \mathcal{O}_{\mathsf{KeyGen}}(\mathbf{r}^{(0)}, \mathbf{r}^{(1)})$
3: $\tilde{\mathbf{z}} \leftarrow \mathcal{A}^{\mathcal{O}'_{\mathsf{KeyGen}}}(\mathsf{pp}, \mathbf{c})$
4: **for** $i = 1$ to t **do**
5: $\quad \mathbf{r}_i \leftarrow_\$ \mathbb{F}^k$
6: $\quad \mathbf{c}_i \leftarrow \mathcal{O}_{\mathsf{KeyGen}}(\mathbf{r}^{(0)}, \mathbf{r}_i)$
7: $\quad s_i \leftarrow \mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c}_i, \tilde{\mathbf{z}})$
8: **end for**
9: **if** $\bigwedge_{i=1}^t s_i \leq \gamma$ **then**
10: $\quad$ **return** $\tilde{b} = 0$
11: **else**
12: $\quad$ **return** $\tilde{b} \leftarrow_\$ \{0, 1\}$
13: **end if**

If the challenge bit $b = 0$, then by Assumption 1, any $s_i \neq \perp$ in Line 7 implies all $s_i \neq \perp$ and $s_i = s_j$ for any i, j. Therefore, the probability that all $s_i \leq \gamma$ in Line 9 is

$$\begin{aligned}
\Pr\left[\bigwedge_{i=1}^t s_i \leq \gamma \mid b = 0\right] &= \Pr\left[s_1 \neq \perp \mid b = 0\right] \cdot \Pr\left[s_1 \leq \gamma \mid b = 0 \wedge s_1 \neq \perp\right] \\
&= \Pr\left[s_1 \neq \perp \mid b = 0\right] \cdot \Pr\left[\mathbf{r}^{(0)} \tilde{\mathbf{v}}^T \leq \gamma \mid b = 0 \wedge s_1 \neq \perp\right] \\
&= \Pr\left[s_1 \neq \perp \mid b = 0\right] \cdot \Pr\left[\mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c}, \tilde{\mathbf{z}}) \leq \gamma \mid b = 0 \wedge s_1 \neq \perp\right] \\
&= \Pr\left[s_1 \neq \perp \mid b = 0\right] \cdot \Pr\left[\mathsf{RUF}^{\mathcal{O}'_{\mathsf{KeyGen}}, \gamma}(\mathcal{A}) \to 1 \mid b = 0 \wedge s_1 \neq \perp\right] \\
&= \Pr\left[\mathsf{RUF}^{\mathcal{O}'_{\mathsf{KeyGen}}, \gamma}(\mathcal{A}) \to 1\right]
\end{aligned}$$

If the challenge bit $b = 1$, for any $i \in [t]$,

$$\begin{aligned}\Pr[s_i \leq \gamma \mid b = 1] &= \Pr[s_i \neq \perp \mid b = 1] \cdot \Pr[s_i \leq \gamma \mid b = 1 \wedge s_i \neq \perp] \\ &= \Pr[s_i \neq \perp \mid b = 1] \cdot \Pr[\mathbf{r}_i \tilde{\mathbf{v}}^T \leq \gamma \mid b = 1 \wedge s \neq \perp]\end{aligned}$$

Note that $\mathbf{r}_i$ is independent of $\tilde{\mathbf{z}}$ and thus independent of $\tilde{\mathbf{v}}$. Hence, $\Pr[\mathbf{r}_i \tilde{\mathbf{v}}^T \leq \gamma \mid b = 1 \wedge s_i \neq \perp] = \frac{\gamma}{\|\mathbb{F}\|}$ and

$$\Pr\left[\bigwedge_{i=1}^{t} s_i \leq \gamma \mid b = 1\right] = \Pr\left[\bigwedge_{i=1}^{t} s_i \neq \perp \mid b = 1\right] \cdot \left(\frac{\gamma}{\|\mathbb{F}\|}\right)^t \leq \left(\frac{\gamma}{\|\mathbb{F}\|}\right)^t$$

In conclusion,

$$\begin{aligned}\Pr[\textsf{fh-IND}(\mathcal{R}) \to 1] &= \frac{1}{2} + \frac{1}{4}\left(\Pr\left[\bigwedge_{i=1}^{t} s_i \leq \gamma \mid b = 0\right] - \Pr\left[\bigwedge_{i=1}^{t} s_i \leq \gamma \mid b = 1\right]\right) \\ &\geq \frac{1}{2} + \frac{1}{4}\left(\Pr[\textsf{RUF}^{\mathcal{O}'_{\textsf{KeyGen}},\gamma}(\mathcal{A}) \to 1] - \left(\frac{\gamma}{\|\mathbb{F}\|}\right)^t\right) \\ &\geq \frac{1}{2} + \frac{1}{4}\left(\Pr[\textsf{RUF}^{\mathcal{O}'_{\textsf{KeyGen}},\gamma}(\mathcal{A}) \to 1] - e^{-t\cdot(1-\frac{\gamma}{\|\mathbb{F}\|})}\right)\end{aligned}$$

Take t be any integer larger than $\frac{\lambda}{1-\frac{\gamma}{\|\mathbb{F}\|}}$. Since $\mathbf{Adv}^{\textsf{fh-IND}}_{\textsf{FE},\mathcal{R}} = \left|\Pr[\textsf{fh-IND}(\mathcal{R}) \to 1] - \frac{1}{2}\right|$ and $e^{-t\cdot(1-\frac{\gamma}{\|\mathbb{F}\|})} \leq e^{-\lambda}$ are negligible,

$$\Pr[\textsf{RUF}^{\mathcal{O}'_{\textsf{KeyGen}},\gamma}(\mathcal{A}) \to 1] \leq e^{-t\cdot(1-\frac{\gamma}{\|\mathbb{F}\|})} + 4 \cdot \mathbf{Adv}^{\textsf{fh-IND}}_{\textsf{FE},\mathcal{R}} = \textsf{negl}.$$

□

Let $\textsf{Sig} = (\textsf{Sig.KeyGen}, \textsf{Sig.Sign}, \textsf{Sig.Verify})$ be an sEUF-CMA signature scheme. By adding $\textsf{Sig}$, an fh-IPFE scheme $\textsf{FE}$ can be upgraded to an RUF scheme $\textsf{FE}'$.

- $\textsf{FE}'.\textsf{Setup}(1^\lambda)$: Run $\textsf{FE.Setup}(1^\lambda) \to (\textsf{msk}, \textsf{pp})$ and $\textsf{Sig.KeyGen}(1^\lambda) \to (\textsf{sk}_{\textsf{Sig}}, \textsf{pk}_{\textsf{Sig}})$. Output $\textsf{msk}' = (\textsf{msk}, \textsf{sk}_{\textsf{Sig}})$ and $\textsf{pp}' = (\textsf{pp}, \textsf{pk}_{\textsf{Sig}})$.
- $\textsf{FE}'.\textsf{KeyGen}(\textsf{msk}', \mathbf{x})$: Run $\textsf{FE.KeyGen}(\textsf{msk}, \mathbf{x}) \to f_{\mathbf{x}}$ and output $f_{\mathbf{x}}$.
- $\textsf{FE}'.\textsf{Enc}(\textsf{msk}', \mathbf{y})$: Run $\textsf{FE.Enc}(\textsf{msk}, \mathbf{y}) \to \mathbf{c_y}$ and sign $\mathbf{c_y}$ by $\textsf{Sig.Sign}(\textsf{sk}_{\textsf{Sig}}, \mathbf{c_y}) \to \sigma$. Output $\mathbf{c_y}' = (\mathbf{c_y}, \sigma)$.
- $\textsf{FE}'.\textsf{Dec}(\textsf{pp}', f_{\mathbf{x}}, \mathbf{c_y}')$: Output the decryption $\textsf{FE.Dec}(\textsf{pp}, f_{\mathbf{x}}, \mathbf{c_y})$ if the verification $\textsf{Sig.Verify}(\textsf{pk}_{\textsf{Sig}}, \mathbf{c_y}, \sigma) = 1$. Otherwise, output $\perp$.

Theorem 5. *For any fh-IPFE $\textsf{FE}$, $\textsf{FE}'$ is an RUF fh-IPFE for any γ.*

Proof. Given an adversary $\mathcal{A}$ in the $\textsf{RUF}^{\mathcal{O}'_{\textsf{KeyGen}},\mathcal{O}'_{\textsf{Enc}},\gamma}_{\textsf{FE}'}$ game, consider the reduction adversary $\mathcal{R}$ in Algorithm 7 which plays the $\textsf{sEUF-CMA}$ game of $\textsf{Sig}$. $\mathcal{R}$ is given a verification public key $\textsf{pk}_{\textsf{Sig}}$ and a signing oracle $\mathcal{O}_{\textsf{Sig}}$ and returns a forged message-signature pair that is not equal to any previous answer of $\mathcal{O}_{\textsf{Sig}}$. To run $\mathcal{A}$, $\mathcal{R}$ simulates each oracle in the following way.

Algorithm 7 $\mathcal{R}^{\mathcal{O}_{\mathsf{Sign}}}(\mathsf{pk}_{\mathsf{Sig}})$

$\mathbf{r} \leftarrow \mathbb{F}^k$
$\mathsf{msk}, \mathsf{pp} \leftarrow \mathsf{FE.Setup}(1^\lambda)$
$\mathbf{c} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{r})$
$\mathsf{pp}' \leftarrow (\mathsf{pp}, \mathsf{pk}_{\mathsf{Sig}})$
$\tilde{\mathbf{z}} \leftarrow \mathcal{A}^{\mathcal{O}'_{\mathsf{KeyGen}}, \mathcal{O}'_{\mathsf{Enc}}}(\mathsf{pp}', \mathbf{c})$
Parse $(\mathbf{c}_{\mathbf{z}}, \sigma') \leftarrow \tilde{\mathbf{z}}$
return $(\mathbf{c}_{\mathbf{z}}, \sigma')$

- $\mathcal{O}'_{\mathsf{KeyGen}}(\mathbf{x}')$: Return $\mathsf{FE.KeyGen}(\mathsf{msk}, \mathbf{x})$.
- $\mathcal{O}'_{\mathsf{Enc}}(\mathbf{y}')$: Run $\mathsf{FE.Enc}(\mathsf{msk}, \mathbf{y}) \to \mathbf{c}_{\mathbf{y}}$ and call the signing oracle $\mathcal{O}_{\mathsf{Sign}}(\mathbf{c}_{\mathbf{y}}) \to \sigma$. Output $\mathbf{c}_{\mathbf{y}}' = (\mathbf{c}_{\mathbf{y}}, \sigma)$.

$\mathcal{R}$ perfectly simulates a RUF game for $\mathcal{A}$, and if $\mathcal{A}$ wins the RUF game, $(\mathbf{c}_{\mathbf{z}}, \sigma')$ is not equal to any previous answer of $\mathcal{O}'_{\mathsf{Enc}}$, and therefoere not equal to any previous message-signature pair $(\mathbf{c}_{\mathbf{y}}, \sigma)$ given from the signing oracle $\mathcal{O}_{\mathsf{Sign}}$. Now, since Sig is sEUF-CMA,

$$\Pr[\mathsf{RUF}^{\mathcal{O}'_{\mathsf{KeyGen}}, \gamma}(\mathcal{A}) \to 1] \leq \Pr[\mathsf{Sig.Verify}(\mathsf{pk}_{\mathsf{Sig}}, \mathbf{c}_{\mathbf{z}}, \sigma') = 1] = \mathsf{negl}.$$

□

5.3 UF Security of Π

We first consider option-UF security when option includes $\mathcal{O}_{\mathsf{Enroll}}$. Note that in this instantiation, csk is the public parameter pp of FE and assumed to be given to all adversaries.

Theorem 6. *Let $\mathsf{option} = \{\mathbf{c}_{\mathbf{x}}, \mathsf{csk}, \mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Enroll}}\}$. For any distribution family $\mathbb{B}$, if FE is fh-IND and $\mathcal{O}'_{\mathsf{KeyGen}}$-RUF for a $\gamma \geq \tau^2$, then Π is option-UF.*

Proof. Given an adversary $\mathcal{A}$ in the $\mathsf{UF}_{\mathsf{option}}$ game, consider the reduction adversary $\mathcal{R}$ in Algorithm 8 which plays the fh-IND game. $\mathcal{R}$ runs $\mathcal{A}$ and simulates $\mathcal{O}_{\mathsf{Enroll}}(\mathsf{esk}, \mathbf{b}')$ by first encoding $\mathbf{b}' = (b'_1, \cdots, b'_k)$ into $\mathbf{x}' = (b'_1, \cdots, b'_k, 1, \|\mathbf{b}'\|^2)$ and calling $\mathcal{O}_{\mathsf{KeyGen}}(\mathbf{x}', \mathbf{x}')$ given in the fh-IND game. Note that since $\mathcal{R}$ never calls $\mathcal{O}_{\mathsf{Enc}}$, it is an admissible adversary.

If the challenge bit $b = 0$, then $\mathcal{R}$ perfectly simulates a $\mathsf{UF}_{\mathsf{option}}$ game for $\mathcal{A}$. Therefore, the probability that $\mathsf{Verify}(s) = 1$ in Line 8 is $\Pr[\mathsf{UF}_{\mathsf{option}}(\mathcal{A}) \to 1]$.

For the case when the challenge bit $b = 1$, consider an adversary $\mathcal{A}'$ in Algorithm 9 in the $\mathsf{RUF}^{\mathcal{O}'_{\mathsf{KeyGen}}}$ game. $\mathcal{A}'$ runs Line 1 and 6 of $\mathcal{R}$ and simulates $\mathcal{O}_{\mathsf{Enroll}}(\mathsf{esk}, \mathbf{b}')$ by first encoding $\mathbf{b}'$ into $\mathbf{x}'$ as before and calling $\mathcal{O}'_{\mathsf{KeyGen}}(\mathbf{x}')$ given in the $\mathsf{RUF}^{\mathcal{O}'_{\mathsf{KeyGen}}}$ game.

Now, if the challenge bit $b = 1$, then $\mathcal{R}$ perfectly simulates $\mathcal{A}'$ in the $\mathsf{RUF}^{\mathcal{O}'_{\mathsf{KeyGen}}}$ game. The probability that $\mathsf{Verify}(s) = 1$, which is equivalent to $s \leq \tau^2$, in Line 8 is $\Pr[\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{KeyGen}}, \tau^2}(\mathcal{A}') \to 1]$

Algorithm 8 $\mathcal{R}^{\mathcal{O}_{\mathsf{KeyGen}},\mathcal{O}_{\mathsf{Enc}}}(\mathsf{pp})$

1: $\mathcal{B} \leftarrow_\$ \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}$
2: $\mathbf{b} = (b_1, \cdots, b_k) \leftarrow \mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}()$
3: $\mathbf{x} \leftarrow (b_1, \cdots, b_k, 1, \|\mathbf{b}\|^2)$
4: $\mathbf{r} \leftarrow_\$ \mathbb{F}^{k+2}$
5: $\mathbf{c} \leftarrow \mathcal{O}_{\mathsf{KeyGen}}(\mathbf{x}, \mathbf{r})$
6: $\tilde{\mathbf{z}} \leftarrow \mathcal{A}^{\mathcal{O}_\mathcal{B},\mathcal{O}_{\mathsf{Enroll}}}(\mathbf{c}, \mathsf{pp})$
7: $s \leftarrow \mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c}, \tilde{\mathbf{z}})$
8: **if** $\mathsf{Verify}(s) = 1$ **then**
9: **return** $\tilde{b} = 0$
10: **else**
11: **return** $\tilde{b} \leftarrow_\$ \{0, 1\}$
12: **end if**

Algorithm 9 $\mathcal{A}'^{\mathcal{O}'_{\mathsf{KeyGen}}}(\mathsf{pp}, \mathbf{c})$

1: $\mathcal{B} \leftarrow_\$ \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}$
2: $\tilde{\mathbf{z}} \leftarrow \mathcal{A}^{\mathcal{O}_\mathcal{B},\mathcal{O}_{\mathsf{Enroll}}}(\mathbf{c}, \mathsf{pp})$
3: **return** $\tilde{\mathbf{z}}$

In conclusion, since $\gamma \geq \tau^2$,

$$\begin{aligned}
\Pr[\mathsf{fh\text{-}IND}(\mathcal{R}) \to 1] &= \Pr[b = 0] \cdot \left(\Pr[\mathsf{Verify}(s) = 1 \mid b = 0] + \frac{1}{2} \cdot \Pr[\mathsf{Verify}(s) = 0 \mid b = 0] \right) \\
&\quad + \Pr[b = 1] \cdot \frac{1}{2} \cdot \Pr[\mathsf{Verify}(s) = 0 \mid b = 1] \\
&= \frac{1}{2} + \frac{1}{4} \left(\Pr[\mathsf{Verify}(s) = 1 \mid b = 0] - \Pr[\mathsf{Verify}(s) = 1 \mid b = 1] \right) \\
&= \frac{1}{2} + \frac{1}{4} \left(\Pr[\mathsf{UF}_{\mathsf{option}}(\mathcal{A}) \to 1] - \Pr[\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{KeyGen}},\tau^2}(\mathcal{A}') \to 1] \right) \\
&\geq \frac{1}{2} + \frac{1}{4} \left(\Pr[\mathsf{UF}_{\mathsf{option}}(\mathcal{A}) \to 1] - \Pr[\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{KeyGen}},\gamma}(\mathcal{A}') \to 1] \right)
\end{aligned}$$

Since both $\mathbf{Adv}^{\mathsf{fh\text{-}IND}}_{\mathsf{FE},\mathcal{R}} = \left| \Pr[\mathsf{fh\text{-}IND}(\mathcal{R}) \to 1] - \frac{1}{2} \right|$ and $\mathbf{Adv}^{\mathsf{RUF},\mathcal{O}'_{\mathsf{KeyGen}},\gamma}_{\mathsf{FE},\mathcal{A}'} = \Pr[\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{KeyGen}},\gamma}(\mathcal{A}') \to 1]$ are negligilbe,

$$\Pr[\mathsf{UF}_{\mathsf{option}}(\mathcal{A}) \to 1] \leq 4 \cdot \mathbf{Adv}^{\mathsf{fh\text{-}IND}}_{\mathsf{FE},\mathcal{R}} + \mathbf{Adv}^{\mathsf{RUF},\mathcal{O}'_{\mathsf{KeyGen}},\gamma}_{\mathsf{FE},\mathcal{A}'} = \mathsf{negl}.$$

□

For option that includes $\mathcal{O}_{\mathsf{Probe}}$, we first note that for any $d \in \mathbb{Z}_q$ and any nonzero vector $\mathbf{r} \in \mathbb{Z}_q^{k+2}$, there exists a vector $\mathbf{y} \in \mathbb{Z}_q^{k+2}$ such that $\mathbf{r}\mathbf{y}^T = d$.

Theorem 7. *Let $\mathsf{option} = \{\mathbf{c}_\mathbf{x}, \mathsf{csk}, \mathcal{O}_\mathcal{B}, \mathcal{O}_{\mathsf{Probe}}\}$. For any distribution family $\mathbb{B}$, if FE is fh-IND and $\mathcal{O}'_{\mathsf{Enc}}$-RUF for a $\gamma \geq \tau^2$, then Π is option-UF.*

Proof. Given an adversary $\mathcal{A}$ in the $\mathsf{UF}_{\mathsf{option}}$ game, consider the reduction adversary $\mathcal{R}$ in Algorithm 10 which plays the fh-IND game. $\mathcal{R}$ runs $\mathcal{A}$ and simulates $\mathcal{O}_{\mathsf{Probe}}$ in the following way.

- $\mathcal{O}_{\mathsf{Probe}}(\mathsf{psk}, \mathbf{b}')$: On input $\mathbf{b}' = (b'_1, \cdots, b'_k)$, it first encodes it as $\mathbf{y}' = (-2b'_1, \cdots, -2b'_k, \|\mathbf{b}'\|^2, 1)$. Next, it computes $d \leftarrow \mathbf{x}\mathbf{y}'^T$ and finds a vector $\mathbf{y}''$ such that $\mathbf{r}\mathbf{y}''^T = d$. Finally, it calls $\mathcal{O}_{\mathsf{Enc}}(\mathbf{y}', \mathbf{y}'')$, which is given by the fh-IND game, and returns the result.

Note that $(\mathbf{x}, \mathbf{r})$ is the only query of $\mathcal{R}$ to $\mathcal{O}_{\mathsf{KeyGen}}$, and for any query $(\mathbf{y}', \mathbf{y}'')$ to $\mathcal{O}_{\mathsf{Enc}}$, it satisfies $\mathbf{x}\mathbf{y}'^T = \mathbf{r}\mathbf{y}''^T$. Hence, $\mathcal{R}$ is an admissible adversary.

Algorithm 10 $\mathcal{R}^{\mathcal{O}_{\mathsf{KeyGen}}, \mathcal{O}_{\mathsf{Enc}}}(\mathsf{pp})$

1: $\mathcal{B} \leftarrow_\$ \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}$
2: $\mathbf{b} = (b_1, \cdots, b_k) \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}}}()$
3: $\mathbf{x} \leftarrow (b_1, \cdots, b_k, 1, \|\mathbf{b}\|^2)$
4: $\mathbf{r} \leftarrow_\$ \mathbb{F}^{k+2}$
5: $\mathbf{c} \leftarrow \mathcal{O}_{\mathsf{KeyGen}}(\mathbf{x}, \mathbf{r})$
6: $\tilde{\mathbf{z}} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Probe}}}(\mathbf{c}, \mathsf{pp})$
7: **if** $\tilde{\mathbf{z}}$ is equal to any output of $\mathcal{O}_{\mathsf{Probe}}$ **then**
8:     **return** $\perp$
9: **end if**
10: $s \leftarrow \mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c}, \tilde{\mathbf{z}})$
11: **if** $\mathsf{Verify}(s) = 1$ **then**
12:     **return** $\tilde{b} = 0$
13: **else**
14:     **return** $\tilde{b} \leftarrow_\$ \{0, 1\}$
15: **end if**

If the challenge bit $b = 0$, then $\mathcal{R}$ perfectly simulates a $\mathsf{UF}_{\mathsf{option}}$ game for $\mathcal{A}$. Therefore, the probability that $\mathsf{Verify}(s) = 1$ in Line 11 is $\Pr[\mathsf{UF}_{\mathsf{option}}(\mathcal{A}) \to 1]$.

For the case when the challenge bit $b = 1$, consider an adversary $\mathcal{A}'$ in Algorithm 11 in the $\mathsf{RUF}^{\mathcal{O}'_{\mathsf{Enc}}}$ game. $\mathcal{A}'$ runs $\mathcal{A}$ and simulates $\mathcal{O}_{\mathsf{Probe}}$ in the following way.

- $\mathcal{O}_{\mathsf{Probe}}(\mathsf{psk}, \mathbf{b}')$: It first encodes $\mathbf{b}'$ into $\mathbf{y}'$ as before. Next, it computes $d \leftarrow \mathbf{x}^{(*)}\mathbf{y}'^T$ and finds a vector $\mathbf{y}''$ such that $\mathbf{r}\mathbf{y}''^T = d$. Finally, it calls $\mathcal{O}'_{\mathsf{Enc}}(\mathbf{y}'')$, which is given by the $\mathsf{RUF}^{\mathcal{O}'_{\mathsf{Enc}}}$ game, and returns the result.

To make $\mathcal{R}$ simulate $\mathcal{A}'$ in the $\mathsf{RUF}^{\mathcal{O}'_{\mathsf{Enc}}}$ game, we still need to ensure two conditions.

- $\mathbf{r} \neq \mathbf{0}$. Otherwise, $\mathcal{A}'$ cannot simulate $\mathcal{O}_{\mathsf{Probe}}$.
- $\tilde{\mathbf{z}} \neq \mathbf{c}^{(i)}$ for all i. The answers of $\mathcal{O}_{\mathsf{Probe}}$ have already been checked in $\mathcal{R}$.

Algorithm 11 $\mathcal{A}'^{\mathcal{O}'_{\mathsf{Enc}}}(\mathsf{pp}, \mathbf{c})$

$\mathcal{B} \leftarrow_{\$} \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}$
$\mathbf{b}^{(*)} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}}}()$
$\mathbf{x}^{(*)} \leftarrow (b_1^{(*)}, \cdots, b_k^{(*)}, 1, \|\mathbf{b}^{(*)}\|^2)$
Sample $k+2$ linearly independent vectors $\{\mathbf{e}^{(i)}\}_{i=1}^{k+2}$.
for $i = 1$ to $k+2$ **do**
  $\mathbf{c}^{(i)} \leftarrow \mathcal{O}'_{\mathsf{Enc}}(\mathbf{e}^{(i)})$.
  $d_i \leftarrow \mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c}, \mathbf{c}^{(i)})$.
end for
Find the vector $\mathbf{r}$ by solving the linear system $\{\mathbf{r}\mathbf{e}^{(i)T} = d_i\}_{i=1}^{k+2}$.
if $\mathbf{r} = \mathbf{0}$ **then**
  return $\perp$
end if
$\tilde{\mathbf{z}} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Probe}}}(\mathbf{c}, \mathsf{pp})$
return $\tilde{\mathbf{z}}$

Let $\mathcal{A}'$ play a tweaked $\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{Enc}}, \tau^2}$ game which does not check that $\tilde{\mathbf{z}}$ is not equal to $\mathbf{c}^{(i)}$ for all i. That is, the game only checks whether $\tilde{\mathbf{z}}$ is not equal to any output of $\mathcal{O}'_{\mathsf{Enc}}$ called by $\mathcal{O}_{\mathsf{Probe}}$ of $\mathcal{A}$. Let the returned value of this game be V. We have Equation 1 and 2. The former one is a relation between $\mathcal{R}$ playing $\mathsf{fh\text{-}IND}$ game when the challenge bit $b = 1$ and V, and the latter is a relation between $\mathcal{A}'$ playing a regular $\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{Enc}}, \tau^2}$ game and the tweaked one.

$$\Pr[\mathsf{Verify}(s) = 1 \mid b = 1 \wedge \mathbf{r} \neq \mathbf{0}] = \Pr[V = 1] \tag{1}$$

$$\Pr[\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{Enc}}, \tau^2}(\mathcal{A}') \to 1] = \Pr\left[V = 1 \mid \bigwedge_{i=1}^{k+2} \tilde{\mathbf{z}} \neq \mathbf{c}^{(i)}\right] \tag{2}$$

For Equation 1, consider that

$$\begin{aligned}\Pr[\mathsf{Verify}(s) = 1 \mid b = 1] &= \Pr[\mathsf{Verify}(s) = 1 \mid b = 1 \wedge \mathbf{r} \neq \mathbf{0}] \cdot \Pr[\mathbf{r} \neq \mathbf{0}] \\ &\quad + \Pr[\mathsf{Verify}(s) = 1 \mid b = 1 \wedge \mathbf{r} = \mathbf{0}] \cdot \Pr[\mathbf{r} = \mathbf{0}] \\ &\leq \Pr[V = 1] + \Pr[\mathbf{r} = 0] \\ &= \Pr[V = 1] + \frac{1}{q^{k+2}}\end{aligned}$$

For Equation 2, consider that

$$\begin{aligned}
\Pr[\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{Enc}},\tau^2}(\mathcal{A}') \to 1] &= \Pr\left[V = 1 \mid \bigwedge_{i=1}^{k+2} \tilde{\mathbf{z}} \neq \mathbf{c}^{(i)}\right] \\
&\geq \Pr[V = 1] - \Pr\left[\neg\left(\bigwedge_{i=1}^{k+2} \tilde{\mathbf{z}} \neq \mathbf{c}^{(i)}\right)\right] \\
&= \Pr[V = 1] - \Pr\left[\bigvee_{i=1}^{k+2} \tilde{\mathbf{z}} = \mathbf{c}^{(i)}\right] \\
&\geq \Pr[V = 1] - \sum_{i=1}^{k+2} \Pr[\tilde{\mathbf{z}} = \mathbf{c}^{(i)}].
\end{aligned}$$

Note that each $\mathbf{c}^{(i)} = \mathsf{FE.Enc}(\mathsf{msk}, \mathsf{pp}, \mathbf{e}^{(i)})$ for some uniform nonzero vector $\mathbf{e}^{(i)}$. Also note that distinct vectors in $\mathbb{Z}_q^{k+2}$ will have different encryptions due to the correctness of FE. Therefore, $\Pr[\tilde{\mathbf{z}} = \mathbf{c}^{(i)}] \leq \frac{1}{q^{k+2}-1}$ and

$$\Pr[\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{Enc}},\tau^2}(\mathcal{A}') \to 1] \geq \Pr[V = 1] - \frac{k+2}{q^{k+2}-1}.$$

Combining both results from Equation 1 and 2, we derive

$$\Pr[\mathsf{Verify}(s) = 1 \mid b = 1] \leq \Pr[V = 1] + \frac{1}{q^{k+2}} \leq \Pr[\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{Enc}},\tau^2}(\mathcal{A}') \to 1] + \frac{k+2}{q^{k+2}-1} + \frac{1}{q^{k+2}}.$$

Finally, similar to the proof of Theorem 6, we derive

$$\begin{aligned}
\Pr[\mathsf{fh\text{-}IND}(\mathcal{R}) \to 1] &= \frac{1}{2} + \frac{1}{4}\left(\Pr[\mathsf{Verify}(s) = 1 \mid b = 0] - \Pr[\mathsf{Verify}(s) = 1 \mid b = 1]\right) \\
&\geq \frac{1}{2} + \frac{1}{4}\left(\Pr[\mathsf{UF}_{\mathsf{option}}(\mathcal{A}) \to 1] - \Pr[\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{Enc}},\tau^2}(\mathcal{A}') \to 1] - \frac{k+2}{q^{k+2}-1} - \frac{1}{q^{k+2}}\right) \\
&\geq \frac{1}{2} + \frac{1}{4}\left(\Pr[\mathsf{UF}_{\mathsf{option}}(\mathcal{A}) \to 1] - \Pr[\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{Enc}},\gamma}(\mathcal{A}') \to 1] - \frac{k+2}{q^{k+2}-1} - \frac{1}{q^{k+2}}\right)
\end{aligned}$$

Since both $\mathbf{Adv}_{\mathsf{FE},\mathcal{R}}^{\mathsf{fh\text{-}IND}} = \left|\Pr[\mathsf{fh\text{-}IND}(\mathcal{R}) \to 1] - \frac{1}{2}\right|$ and $\mathbf{Adv}_{\mathsf{FE},\mathcal{A}'}^{\mathsf{RUF},\mathcal{O}'_{\mathsf{Enc}},\gamma} = \Pr[\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{Enc}},\gamma}(\mathcal{A}') \to 1]$ are negligible,

$$\Pr[\mathsf{UF}_{\mathsf{option}}(\mathcal{A}) \to 1] \leq 4 \cdot \mathbf{Adv}_{\mathsf{FE},\mathcal{R}}^{\mathsf{fh\text{-}IND}} + \mathbf{Adv}_{\mathsf{FE},\mathcal{A}'}^{\mathsf{RUF},\mathcal{O}'_{\mathsf{Enc}},\gamma} + \frac{k+2}{q^{k+2}-1} + \frac{1}{q^{k+2}} = \mathsf{negl}.$$

□

Unfortunately, for the instantiation in Section 3.2.1, we cannot achieve UF security when the adversary has psk, even if the false positive rate is negligible. The adversary can simply compute $\mathbf{c} \leftarrow \mathsf{Probe}(\mathsf{psk}, \mathbf{0})$ and return $\mathbf{c}$. The same results also hold for option that includes esk since both psk and esk are equal to msk and allow the adversary to run $\mathsf{FE.Enc}(\mathsf{msk}, \mathsf{pp}, \mathbf{v})$ for any vector $\mathbf{v}$. We state this result formally in the following theorem.

Theorem 8. *Let option include esk or psk. For any distribution family $\mathbb{B}$ and functional encryption FE, Π is not option-UF.*

5.4 IND Security of Π

For the IND security, we first consider the following definition and assumption on the biometric distribution family $\mathbb{B}$.

Definition 9. For an authentication scheme Π, a distribution $\mathcal{B} \in \mathbb{B}$, and an integer t, define the distribution $\mathcal{D}_{\mathcal{B}}(t)$ as

$$\mathcal{D}_{\mathcal{B}}(t) = \big(\mathsf{BioCompare}(\mathbf{b}, \mathbf{b}^{(1)}), \mathsf{BioCompare}(\mathbf{b}, \mathbf{b}^{(2)}), \cdots, \mathsf{BioCompare}(\mathbf{b}, \mathbf{b}^{(t)})\big)$$

where $\mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}}}()$ and $\mathbf{b}^{(i)} \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}}}()$ for all $i \in [t]$.

Assumption 2. Let t be an integer. Assume that for any two distributions $\mathcal{B}^{(0)}$ and $\mathcal{B}^{(1)}$ in the biometric distribution family $\mathbb{B}$, $\mathcal{D}_{\mathcal{B}^{(0)}}(t)$ and $\mathcal{D}_{\mathcal{B}^{(1)}}(t)$ are the same.

Note that indistinguishability between $\mathcal{D}_{\mathcal{B}^{(0)}}(t)$ and $\mathcal{D}_{\mathcal{B}^{(1)}}(t)$ is a necessary condition to achieve IND security because

$$\big(\mathsf{Compare}(\mathsf{csk}, \mathbf{c_x}, \mathbf{c_y}^{(1)}), \cdots, \mathsf{Compare}(\mathsf{csk}, \mathbf{c_x}, \mathbf{c_y}^{(t)})\big) = \mathcal{D}_{\mathcal{B}^{(b)}}(t)$$

where b is the challenge bit.

Theorem 9. *For any distribution family $\mathbb{B}$ satisfying Assumption 2 and having a true positive rate $\mathsf{TP} > \frac{1}{\mathsf{poly}}$, if FE is fh-IND, then Π is IND.*

Proof. Given an adversary $\mathcal{A}$ in the IND game, consider the reduction adversary $\mathcal{R}$ in Algorithm 12 which plays the $\mathsf{fh\text{-}IND}$ game by running $\mathcal{A}$.

Algorithm 12 $\mathcal{R}^{\mathcal{O}_{\mathsf{KeyGen}}, \mathcal{O}_{\mathsf{Enc}}}(\mathsf{pp})$

$\mathcal{B}^{(0)} \leftarrow_{\$} \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}^{(0)}$
$\mathcal{B}^{(1)} \leftarrow_{\$} \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}^{(1)}$
$\mathbf{b}^{(0)} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}^{(0)}}}(), \mathbf{x}^{(0)} \leftarrow (b_1^{(0)}, \cdots, b_k^{(0)}, 1, \|\mathbf{b}^{(0)}\|^2)$
$\mathbf{b}^{(1)} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}^{(1)}}}(), \mathbf{x}^{(1)} \leftarrow (b_1^{(1)}, \cdots, b_k^{(1)}, 1, \|\mathbf{b}^{(1)}\|^2)$
$\mathbf{c_x} \leftarrow \mathcal{O}_{\mathsf{KeyGen}}(\mathbf{x}^{(0)}, \mathbf{x}^{(1)})$
for $i = 1$ to t **do**
  $\mathbf{b'}^{(0)} \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}^{(0)}}}()$
  $\mathbf{y}^{(0)} \leftarrow (-2{b'_1}^{(0)}, \cdots, -2{b'_k}^{(0)}, \|\mathbf{b'}^{(0)}\|^2, 1)$
  repeat
    $\mathbf{b'}^{(1)} \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}^{(1)}}}()$
    $\mathbf{y}^{(1)} \leftarrow (-2{b'_1}^{(1)}, \cdots, -2{b'_k}^{(1)}, \|\mathbf{b'}^{(1)}\|^2, 1)$
  until $\mathbf{x}^{(0)}{\mathbf{y}^{(0)}}^T = \mathbf{x}^{(1)}{\mathbf{y}^{(1)}}^T$
  $\mathbf{c_y}^{(i)} \leftarrow \mathcal{O}_{\mathsf{Enc}}(\mathbf{y}^{(0)}, \mathbf{y}^{(1)})$
end for
$\tilde{b} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{B}^{(0)}}, \mathcal{O}_{\mathcal{B}^{(1)}}}(\mathsf{pp}, \mathbf{c_x}, \{\mathbf{c_y}^{(i)}\}_{i=1}^t)$
return $\tilde{b}$

Note that $(\mathbf{x}^{(0)}, \mathbf{x}^{(1)})$ is the only query of $\mathcal{R}$ to $\mathcal{O}_{\mathsf{KeyGen}}$, and for any query $(\mathbf{y}^{(0)}, \mathbf{y}^{(1)})$ to $\mathcal{O}_{\mathsf{Enc}}$, it satisfies $\mathbf{x}^{(0)}{\mathbf{y}^{(0)}}^T = \mathbf{x}^{(1)}{\mathbf{y}^{(1)}}^T$. Hence, $\mathcal{R}$ is an admissible adversary.

The probability that Line 12 is satisfied is

$$\begin{aligned}
\Pr[\mathcal{D}_{\mathcal{B}^{(0)}}(1) = \mathcal{D}_{\mathcal{B}^{(1)}}(1)] &\geq \sum_{i=0}^{\tau} \Pr[\mathcal{D}_{\mathcal{B}^{(0)}}(1) = i]^2 \qquad \text{(Assumption 2)} \\
&\geq \frac{1}{\tau+1} \cdot \left(\sum_{i=0}^{\tau} \Pr[\mathcal{D}_{\mathcal{B}^{(0)}}(1) = i] \right)^2 \\
&= \frac{1}{\tau+1} \cdot \left(\Pr\left[\begin{matrix} \mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}^{(0)}}}() \\ \mathbf{b}' \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}^{(0)}}}() \end{matrix} : \|\mathbf{b} - \mathbf{b}'\| \leq \tau \right] \right)^2 \\
&= \frac{\mathsf{TP}(\mathcal{B}^{(0)})^2}{\tau+1} = \frac{\mathsf{TP}^2}{\tau+1} \qquad \text{(Assumption 2)}
\end{aligned}$$

The expected number of repetitions is bounded above by $\frac{\tau+1}{\mathsf{TP}^2}$. Moreover, the probability that it is satisfied within T repetitions is at least

$$1 - (1 - \frac{\mathsf{TP}^2}{\tau+1})^T \geq 1 - e^{-T \cdot \frac{\mathsf{TP}^2}{\tau+1}}$$

We can reach a $1 - \mathsf{negl}.$ probability that the loop will end within T times by setting a polynomial-size T.

Now, we show that $\mathcal{R}$ perfectly simulate an IND game for $\mathcal{A}$. If the challenge bit b of the fh-IND game is 0, $\mathbf{c_x}$ and ${\mathbf{c_y}}^{(i)}$ for all $i \in [t]$ are generated from $\mathcal{B}^{(0)}$ and have the same distributions as the inputs for an adversary in IND game. If the challenge bit b is 1, we show that distributions of $\mathbf{c_x}$, $\{{\mathbf{c_y}}^{(i)}\}_{i=1}^t$ also follow the same distribution given Assumption 2.

Let $b' \in \{0,1\}$, define distributions

$$\begin{aligned}
\mathbf{X}^{(b')} &= \{\mathbf{b}^{(b')} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}^{(b')}}}() : \mathbf{x}^{(b')} \leftarrow (b_1^{(b')}, \cdots, b_k^{(b')}, 1, \|\mathbf{b}^{(b')}\|^2)\} \\
\mathbf{Y}_i^{(b')} &= \{\mathbf{b}^{(b')} \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}^{(b')}}}() : \mathbf{y}^{(b')} \leftarrow (-2b_1^{(b')}, \cdots, -2b_k^{(b')}, \|\mathbf{b}^{(b')}\|^2, 1)\} \\
\{\mathbf{Y}_i^{(b')}\}_{i \in [t]} &= (\mathbf{Y}_1^{(b')}, \cdots, \mathbf{Y}_t^{(b')}) \qquad (t \text{ identical and independent distributions})
\end{aligned}$$

Let $\mathbf{Y}'_i$ be the distribution of $\mathbf{y}^{(1)}$ derived after the loop in Line 12 in the i-th iteration. For any $\{d_i\}_{i=1}^t, d_i > 0$,

$$\begin{aligned}
\Pr\left[\bigwedge_{i=1}^{t} \mathbf{X}^{(0)} {\mathbf{Y}_i^{(0)}}^T = d_i^2 \right] &= \Pr\left[\mathcal{D}_{\mathcal{B}^{(0)}}(t) = (d_1, \cdots, d_t) \right] \\
&= \Pr\left[\mathcal{D}_{\mathcal{B}^{(1)}}(t) = (d_1, \cdots, d_t) \right] = \Pr\left[\bigwedge_{i=1}^{t} \mathbf{X}^{(1)} {\mathbf{Y}_i^{(1)}}^T = d_i^2 \right]
\end{aligned}$$

Hence, for any $\mathbf{x}$ and $\{\mathbf{y}_i\}_{i=1}^t$,

$$\begin{aligned}
&\Pr[\mathbf{X}^{(1)} = \mathbf{x}, \mathbf{Y}'_1 = \mathbf{y}_1, \cdots, \mathbf{Y}'_t = \mathbf{y}_t] \\
&= \sum_{d_1,\cdots,d_t} \left(\Pr\left[\mathbf{X}^{(1)} = \mathbf{x}, \mathbf{Y}_1^{(1)} = \mathbf{y}_1, \cdots, \mathbf{Y}_t^{(1)} = \mathbf{y}_t \mid \bigwedge_{i=1}^t \mathbf{X}^{(1)} {\mathbf{Y}_i^{(1)}}^T = d_i^2 \right] \right. \\
&\qquad\qquad \left. \times \Pr\left[\bigwedge_{i=1}^t \mathbf{X}^{(0)} {\mathbf{Y}_i^{(0)}}^T = d_i^2 \right] \right) \\
&= \sum_{d_1,\cdots,d_t} \left(\Pr\left[\mathbf{X}^{(1)} = \mathbf{x}, \mathbf{Y}_1^{(1)} = \mathbf{y}_1, \cdots, \mathbf{Y}_t^{(1)} = \mathbf{y}_t \mid \bigwedge_{i=1}^t \mathbf{X}^{(1)} {\mathbf{Y}_i^{(1)}}^T = d_i^2 \right] \right. \\
&\qquad\qquad \left. \times \Pr\left[\bigwedge_{i=1}^t \mathbf{X}^{(1)} {\mathbf{Y}_i^{(1)}}^T = d_i^2 \right] \right) \\
&= \Pr[\mathbf{X}^{(1)} = \mathbf{x}, \mathbf{Y}_1^{(1)} = \mathbf{y}_1, \cdots, \mathbf{Y}_t^{(1)} = \mathbf{y}_t]
\end{aligned}$$

which implies $\mathcal{R}$ also perfectly simulate an IND game for $\mathcal{A}$ when the challenge bit $b = 1$.

In conclusion,

$$\mathbf{Adv}_{\mathsf{FE},\mathcal{R}}^{\mathsf{fh\text{-}IND}} = \mathbf{Adv}_{\Pi,\mathbb{B},\mathcal{A}}^{\mathsf{IND}} = \mathsf{negl}.$$

which holds for all adversaries $\mathcal{A}$ in the IND game. This implies the IND security of Π.

□

6 Security Analysis: Relational Hash-based Instantiation

Let Π be an authentication scheme instantiated by a relational hash scheme RH as in Section 3.2.2. We discuss the UF and IND security of Π in this section. Note that in this instantiation, $\mathsf{esk}, \mathsf{psk}, \mathsf{csk}$ are all public hash keys pk of FE and assumed to be given to all adversaries.

Given a relational scheme RH for a relation $R \subseteq X \times Y$, we first define the unforgeability [MR14] of RH.

Definition 10 (Unforgeability)**.** A relational hash scheme RH is called *unforgeable* for the distribution $\mathcal{X}$ if for any adversary $\mathcal{A}$, the following probability is negligible.

$$\Pr\left[\begin{array}{l} \mathbf{x} \leftarrow_\$ \mathcal{X} \\ \mathsf{pk} \leftarrow \mathsf{RH.KeyGen}(1^\lambda) \\ \mathbf{h_x} \leftarrow \mathsf{RH.Hash}_1(\mathsf{pk}, \mathbf{x}) \\ \tilde{\mathbf{z}} \leftarrow \mathcal{A}(\mathsf{pk}, \mathbf{h_x}) \end{array} : \mathsf{RH.Verify}(\mathsf{pk}, \mathbf{h_x}, \tilde{\mathbf{z}}) = 1 \right] = \mathsf{negl}.$$

6.1 UF Security of Π

We first consider option that includes $\mathbf{c_x}$.

Theorem 10. *Let* $\mathsf{option} = \{\mathbf{c_x}, \mathsf{esk}, \mathsf{psk}, \mathsf{csk}\}$*. If* RH *is unforgeable for the distribution*

$$\mathcal{X} = \{\mathcal{B} \leftarrow_\$ \mathbb{B} : \mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}() \mid \mathbb{B}\},$$

then Π *is* option*-UF.*

In [MR14], the authors construct an RH that is unforgeable for the uniform distribution over $\{0,1\}^k$, under the hardness of some computational problems. Note that we need to provide knowledge of $\mathbb{B}$ in the distribution $\mathcal{X}$.

Proof. Recall that the distribution of $\mathbf{c_x}$ in the UF game of the instantiation of Section 3.2.2 is

$$\left\{\begin{matrix} \mathcal{B} \leftarrow_\$ \mathbb{B} \\ \mathsf{pk} \leftarrow \mathsf{RH.KeyGen}(1^\lambda) \\ \mathbf{x} = \mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}() \end{matrix} : \mathbf{c_x} \leftarrow \mathsf{RH.Hash}_1(\mathsf{pk}, \mathbf{x})\right\}$$

Also recall that $\mathsf{Verify}(\mathsf{Compare}(\mathsf{csk}, \mathbf{c_x}, \tilde{\mathbf{z}})) = \mathsf{RH.Verify}(\mathsf{pk}, \mathbf{c_x}, \tilde{\mathbf{z}})$. The option-UF security is thus guaranteed by the unforgeability of RH.

□

Remark As we mentioned in Section 4.1, an adversary with psk can enjoy a winning rate of the false positive rate FP of $\mathbb{B}$. Theorem 10 thus implies that if FP is not negligible, there does not exist an RH that is unforgeable for the distribution $\{\mathcal{B} \leftarrow_\$ \mathbb{B} : \mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}() \mid \mathbb{B}\}$.

Note that since $\mathsf{esk}, \mathsf{psk}$, and csk are all public in this instantiation, it is meaningless to discuss $\mathcal{O}_{\mathsf{Enroll}}, \mathcal{O}_{\mathsf{Probe}}$, or $\mathcal{O}_{\mathsf{log}}$. In addition, for option that includes $\mathcal{O}_\mathcal{B}$ or $\mathcal{O}'_{\mathsf{Probe}}$, as discussed in Section 4.1, we cannot achieve option-UF security since psk is public in this instantiation.

For option that includes $\mathcal{O}'_{\mathsf{Enroll}}$, we notice that for the RH construction in [MR14], there exists an invalid pk' such that $\mathsf{RH.Hash}_1(\mathsf{pk}', \mathbf{x})$ directly leaks $\mathbf{x}$. By returning $\mathsf{RH.Hash}_2(\mathsf{pk}, \mathbf{x})$, one can break the $\mathsf{UF}_{\mathsf{option}}$ game with probability 1.

6.2 IND Security of Π

For the IND security, we have a negative result for Π.

Theorem 11. *For any distribution family* $\mathbb{B}$ *that* $\mathsf{TP} - \mathsf{FP} > \frac{1}{\mathsf{poly}}$*, and for any relational hash scheme* RH*,* Π *is not IND for any* $t \geq 0$*.*

Proof. Consider the adversary $\mathcal{A}$ in Algorithm 13. When the challenge bit $b = 0$, the probability that $\mathcal{A}$ wins is TP. When the challenge bit $b = 1$, the probability that $\mathcal{A}$ wins is $1 - \mathsf{FP}$. Now,

$$\mathbf{Adv}^{\mathsf{IND}}_{\Pi,\mathbb{B},\mathcal{A}} = \left|\Pr[\mathsf{IND}_\Pi(\mathcal{A}) \to 1] - \frac{1}{2}\right| = \left|\frac{1}{2}(\mathsf{TP} + 1 - \mathsf{FP}) - \frac{1}{2}\right| > \frac{1}{\mathsf{poly}}.$$

□

Algorithm 13 $\mathcal{A}^{\mathcal{O}_{\mathcal{B}^{(0)}},\mathcal{O}_{\mathcal{B}^{(1)}}}(\mathsf{csk}=\mathsf{pk},\mathbf{c_x},\{\mathbf{c_y}^{(i)}\}_{i=1}^{t})$

1: $\mathbf{y}^{(0)}=\mathbf{b}^{(0)}\leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}^{(0)}}}()$
2: $\mathbf{h_y}^{(0)}\leftarrow \mathsf{RH.Hash}_2(\mathsf{pk},\mathbf{y}^{(0)})$
3: **if** $\mathsf{RH.Verify}(\mathsf{pk},\mathbf{c_x},\mathbf{h_y}^{(0)})=1$ **then**
4: **return** 0
5: **else**
6: **return** 1
7: **end if**

We note that this insecurity result holds whenever psk is public. When esk is public, one can also use $\mathbf{c_y}^{(i)}$ to verify from which distribution the chalenge ciphertexts are generated. We write this observation formally in the following theorem.

Theorem 12. *Given any distribution family $\mathbb{B}$ that $\mathsf{TP}-\mathsf{FP}>\frac{1}{\mathsf{poly}}$. If psk is public, Π is not IND for any $t\geq 0$. If esk is public, Π is not IND for any $t\geq 1$.*

6.2.1 IND Security for a Particular Biometric Layer

Recall that in Section 4.2, we introduce as an example a particular biometric layer:

$$\mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}}}()\rightarrow \mathbf{b}^*+\mathcal{E}_{\mathsf{Enroll}} \quad\text{and}\quad \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}}}()\rightarrow \mathbf{b}^*+\mathcal{E}_{\mathsf{Probe}}$$

where $\mathbf{b}^*\in\{0,1\}^k$ is a fixed vector only dependent on $\mathcal{B}$, and $\mathcal{E}_{\mathsf{Enroll}},\mathcal{E}_{\mathsf{Probe}}\subseteq\{0,1\}^k$ are some *error distributions* independent of $\mathcal{B}$. With the same relational hash RH in Section 3.2.2, we can instantiate another authentication scheme using RH.

- $\mathsf{Setup}(1^\lambda)$: It runs $\mathsf{RH.KeyGen}(1^\lambda)\rightarrow\mathsf{pk}$ and samples $\mathbf{r}\leftarrow_{\$}\{0,1\}^k$. Then it outputs $\mathsf{esk}\leftarrow(\mathsf{pk},\mathbf{r})$, $(\mathsf{psk}\leftarrow\mathsf{pk},\mathbf{r})$, and $\mathsf{csk}\leftarrow\mathsf{pk}$.
- $\mathsf{Enroll}(\mathsf{esk},\mathbf{b})$: Let $\mathbf{x}\leftarrow\mathbf{b}$. It calls $\mathsf{RH.Hash}_1(\mathsf{pk},\mathbf{x}+\mathbf{r})\rightarrow\mathbf{h_x}$ and outputs $\mathbf{c_x}\leftarrow\mathbf{h_x}$.
- $\mathsf{Probe}(\mathsf{psk},\mathbf{b}')$: Let $\mathbf{y}\leftarrow\mathbf{b}$. It calls $\mathsf{RH.Hash}_2(\mathsf{pk},\mathbf{y}+\mathbf{r})\rightarrow\mathbf{h_y}$ and outputs $\mathbf{c_y}\leftarrow\mathbf{h_y}$.
- $\mathsf{Compare}(\mathsf{csk},\mathbf{c_x},\mathbf{c_y})$: It calls $\mathsf{RH.Verify}(\mathsf{pk},\mathbf{h_x},\mathbf{h_y})\rightarrow s$ and outputs the value s.

Correctness holds because

$$\mathsf{Compare}(\mathsf{csk},\mathbf{c_x},\mathbf{c_y})=1\Leftrightarrow\mathsf{HD}(\mathbf{x}+\mathbf{r},\mathbf{y}+\mathbf{r})\leq\tau\Leftrightarrow\mathsf{HD}(\mathbf{x},\mathbf{y})\leq\tau=\mathsf{BioCompare}(\mathbf{b},\mathbf{b}').$$

With the same argument in Theorem 2, one can prove that this new scheme is now IND, albeit at the cost of requiring esk and psk to remain secret.

A Construction in [Kim+16]

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of order a prime number q with generators g_1 and g_2, respectively. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a mapping to a target group $\mathbb{G}_T$ also of order q.

Definition 11 (Bilinear asymmetric group [Kim+16])**.** A tuple $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e)$ is a *bilinear asymmetric group* if the following hold.

- Group operations in $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ and mapping e are efficiently computable.
- e is bilinear. That is, for $x, y \in \mathbb{Z}_q$, $e(g_1^x, g_2^y) = e(g_1, g_2)^{xy}$.
- e is non-degenerate. That is, $e(g_1, g_2) \neq 1$, the identity element of $\mathbb{G}_T$.

For a vector $\mathbf{v} = (v_1, v_2, \cdots, v_n) \in \mathbb{Z}_q^n$ and a group element g in group of order q, we write $g^{\mathbf{v}}$ to denote the vector of group elements $(g^{v_1}, g^{v_2}, \cdots, g^{v_n})$. Moreover, for $k \in \mathbb{Z}_q$ and $\mathbf{v}, \mathbf{w} \in \mathbb{Z}_q^n$, we write $(g^{\mathbf{v}})^k = g^{k \cdot \mathbf{v}}$ and $g^{\mathbf{v}} \cdot g^{\mathbf{w}} = g^{\mathbf{v}+\mathbf{w}}$. Finally, the pairing operation is extended to vectors.

$$e(g_1^{\mathbf{v}}, g_2{}^{\mathbf{w}}) = \prod_{i \in [n]} e(g_1^{v_i}, g_2^{w_i}) = e(g_1, g_2)^{\mathbf{v}\mathbf{w}^T}.$$

We now recall the fh-IPFE construction FE in [Kim+16].

- $\mathsf{FE.Setup}(1^\lambda)$: Sample an asymmetric bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e)$ and choose generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$. Sample $\mathbf{B} \in \mathbb{GL}_n(\mathbb{Z}_q)$ and find $\mathbf{B}^* = \det(\mathbf{B}) \cdot (\mathbf{B}^{-1})^T$. Finally, output the public parameter $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, e)$ and the master secret key $\mathsf{msk} = (\mathsf{pp}, g_1, g_2, \mathbf{B}, \mathbf{B}^*)$.
- $\mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{x})$: Sample $\alpha \leftarrow_\$ \mathbb{Z}_q$ and output
$$f_{\mathbf{x}} = (K_1, K_2) = \left(g_1^{\alpha \cdot \det(\mathbf{B})}, g_1^{\alpha \cdot \mathbf{x} \cdot \mathbf{B}}\right)$$
- $\mathsf{FE.Enc}(\mathsf{msk}, \mathsf{pp}, \mathbf{y})$: Sample $\beta \leftarrow_\$ \mathbb{Z}_q$ and output
$$\mathbf{c_y} = (C_1, C_2) = \left(g_2^{\beta}, g_2^{\beta \cdot \mathbf{y} \cdot \mathbf{B}^*}\right)$$
- $\mathsf{FE.Dec}(\mathsf{pp}, f_{\mathbf{x}}, \mathbf{c_y}) \to z$: Parse $f_{\mathbf{x}} = (K_1, K_2)$ and $\mathbf{c_y} = (C_1, C_2)$ and compute
$$D_1 = e(K_1, C_1) \quad \text{and} \quad D_2 = e(K_2, C_2)$$
Solve the discrete logarithm to find z such that $D_1^z = D_2$ and output z. If it fails to find such z, output $\perp$.

Correctness We have

$$D_1 = e(K_1, C_1) = e(g_1, g_2)^{\alpha \cdot \beta \cdot \det(\mathbf{B})}$$

and

$$D_2 = e(K_2, C_2) = e(g_1, g_2)^{\alpha \cdot \beta \cdot \mathbf{x} \cdot \mathbf{B} \cdot (\mathbf{B}^*)^T \cdot \mathbf{y}^T \cdot} = e(g_1, g_2)^{\alpha \cdot \beta \cdot \det(\mathbf{B}) \cdot \mathbf{x}\mathbf{y}^T}.$$

Therefore, $(D_1)^{\mathbf{x}\mathbf{y}^T} = D_2$.

Remark In this construction, q is exponential to λ to achieve security, and decryption relies on some priori knowledge of possible ranges of the inner product $\mathbf{xy}^T$. For example, for the instantiation in Section 3.2.1, one can enumerate $z \in \{0, 1, \cdots, \tau\}$ and return $\perp$ when no valid $z \leq \tau$ such that $D_1^z = D_2$ is found.

B One-Way Game

Algorithm 14 $\mathsf{OW}_{\Pi,\mathbb{B}}(\mathcal{A})$

1: $\mathcal{B} \leftarrow_\$ \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}$
2: $\mathsf{esk}, \mathsf{psk}, \mathsf{csk} \leftarrow \mathsf{Setup}(1^\lambda)$
3: $\mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_\mathcal{B}}()$
4: $\mathbf{c_x} \leftarrow \mathsf{Enroll}(\mathsf{esk}, \mathbf{b})$
5: $\tilde{\mathbf{b}} \leftarrow \mathcal{A}(\mathsf{option})$
6: $s \leftarrow \mathsf{BioCompare}(\tilde{\mathbf{b}}, \mathbf{b})$
7: **return** $\mathsf{Verify}(s)$

option can be $\mathbf{c_x}, \mathsf{esk}, \mathsf{psk}, \mathsf{csk}$ or oracles $\mathcal{O}_{\mathsf{Enroll}}, \mathcal{O}_{\mathsf{Probe}}, \mathcal{O}'_{\mathsf{Enroll}}, \mathcal{O}'_{\mathsf{Probe}}$.

In the reusable fuzzy extractor paper [Boy04], if public strings related to personal information are generated many times, the sensitive information might leak. In our setting, it is a little similar to having multiple $\mathbf{c_x}^{(i)} \leftarrow \mathsf{Enroll}(\mathsf{esk}_i, \mathbf{b})$.

To model reusability, I think we can consider either providing $\mathbf{c_x}^{(i)} \leftarrow \mathsf{Enroll}(\mathsf{esk}_i, \mathbf{b})$ for different esk_i, or just providng $\mathcal{O}'_{\mathsf{Enroll}}$.

References

[Boy04] Xavier Boyen. "Reusable cryptographic fuzzy extractors". In: *Proceedings of the 11th ACM Conference on Computer and Communications Security*. CCS '04. Washington DC, USA: Association for Computing Machinery, 2004, pp. 82–91. ISBN: 1581139616. DOI: 10.1145/1030083.1030096. URL: https://doi.org/10.1145/1030083.1030096.

[BCP12] Julien Bringer, Herve Chabanne, and Alain Patey. *SHADE: Secure HAmming DistancE computation from oblivious transfer*. Cryptology ePrint Archive, Paper 2012/586. 2012. URL: https://eprint.iacr.org/2012/586.

[Yas+13] Masaya Yasuda et al. "Packed Homomorphic Encryption Based on Ideal Lattices and Its Application to Biometrics". In: *Security Engineering and Intelligence Informatics*. Ed. by Alfredo Cuzzocrea et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 55–74. ISBN: 978-3-642-40588-4.

[MR14] Avradip Mandal and Arnab Roy. *Relational Hash*. Cryptology ePrint Archive, Paper 2014/394. 2014. URL: https://eprint.iacr.org/2014/394.

[DDM15] Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay. *Functional Encryption for Inner Product with Full Function Privacy*. Cryptology ePrint Archive, Paper 2015/1255. 2015. URL: https://eprint.iacr.org/2015/1255.

[Kim+16] Sam Kim et al. *Function-Hiding Inner Product Encryption is Practical*. Cryptology ePrint Archive, Paper 2016/440. 2016. URL: https://eprint.iacr.org/2016/440.

[TAO16] Junichi Tomida, Masayuki Abe, and Tatsuaki Okamoto. "Efficient Functional Encryption for Inner-Product Values with Full-Hiding Security". In: *Information Security*. Ed. by Matt Bishop and Anderson C A Nascimento. Cham: Springer International Publishing, 2016, pp. 408–425. ISBN: 978-3-319-45871-7.

[Li+17] Nan Li et al. "Fuzzy Extractors for Biometric Identification". In: *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. 2017, pp. 667–677. DOI: 10.1109/ICDCS.2017.107.

[Lee+18] Joohee Lee et al. *Instant Privacy-Preserving Biometric Authentication for Hamming Distance*. Cryptology ePrint Archive, Paper 2018/1214. 2018. URL: https://eprint.iacr.org/2018/1214.

[Che+21] Jung Hee Cheon et al. "Lattice-Based Secure Biometric Authentication for Hamming Distance". In: *Information Security and Privacy*. Ed. by Joonsang Baek and Sushmita Ruj. Cham: Springer International Publishing, 2021, pp. 653–672. ISBN: 978-3-030-90567-5.

[PM21] Gaëtan Pradel and Chris Mitchell. *Privacy-Preserving Biometric Matching Using Homomorphic Encryption*. 2021. arXiv: 2111.12372 [cs.CR]. URL: https://arxiv.org/abs/2111.12372.

[Cac+22] Chloe Cachet et al. "Proximity Searchable Encryption for the Iris Biometric". In: *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*. ASIA CCS '22. Nagasaki, Japan: Association for Computing Machinery, 2022, pp. 1004–1018. ISBN: 9781450391405. DOI: 10.1145/3488932.3497754. URL: https://doi.org/10.1145/3488932.3497754.

[PP22] Paola de Perthuis and David Pointcheval. *Two-Client Inner-Product Functional Encryption, with an Application to Money-Laundering Detection*. Cryptology ePrint Archive, Paper 2022/441. 2022. DOI: 10.1145/3548606.3559374. URL: https://eprint.iacr.org/2022/441.

[EM23] Johannes Ernst and Aikaterini Mitrokotsa. *A Framework for UC Secure Privacy Preserving Biometric Authentication using Efficient Functional Encryption*. Cryptology ePrint Archive, Paper 2023/481. 2023. URL: https://eprint.iacr.org/2023/481.