The Cryptographic Layer of Biometric Authentication

Keng-Yu Chen

Supervisor: Serge Vaudenay
LASEC, EPFL

April 17, 2025

Abstract

In this project, we focus on the cryptographic layer for biometric authentication. The layer is added on the top of authentication scheme 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 our 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 false positive rate that is not negligible. 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 Work 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 [JP09; 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.

Explanation [A brief explanation of this work]

Contribution In this project, we present the following contributions:

- We provide a new general framework for analyzing a non-interactive biometric authentication scheme. Our framework formalizes a biometric authentication scheme by splitting it 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.

- We list two security games to model two security notions that we consider relevant to a biometric authentication scheme: the unforgeability (UF) game and the 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 successful authentication, which is similar to the unforgeability notion in [MR14] and the malicious adversary in [EM23]. However, we consider several options for the adversary to add more 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. Previous works [MR14; Lee+18; Che+21; EM23] consider a similar security notion by considering an adversary who has enrollment and probe messages. The security follows if the adversary cannot learn any information about the biometrics. Compared to them, our model provides the adversary with oracles to users' biometris and ask it to tell which one is used in the authentication process. This captures the server's ability of identifying the users and takes biometric distributions into consideration.

- We analyze the UF and IND security of existing instantiations of biometric authentication schemes from previous works. Our results demonstrate necessary and sufficient conditions and provide transformations for these instantiations to achieve our desired security.

Structure of the Paper In Section 2, we formally define a biometric authentication scheme, including the biometric layer and cryptographic layer. In Section 3, we introduce the unforgeability (UF) game and the indistinguishability (IND) game. In Section 4 and 5, we recall two instantiations using function-hiding inner-product functional encryption and relational hash, repsectively, and provide analyses of the UF and IND security of them.

Notation 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.
- For two equal-length vectors $\mathbf{x} = (x_1, \cdots, x_n)$ and $\mathbf{y} = (y_1, \cdots, y_n)$, $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{n} x_i \cdot y_i$ is the canonical inner product of $\mathbf{x}$ and $\mathbf{y}$.
- A PPT algorithm denotes a probabilistic polynomial time algorithm. Unless otherwise specified, all algorithms run in PPT.

2 Formalization

2.1 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 1 (Biometric Authentication Scheme)**.** A *biometric authentication scheme* 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. In practice, $\mathsf{getEnroll}$ can collect several biometric samples from a user's biometric distribution $\mathcal{B}$ to create a more accurate template.
- $\mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}}}() \to \mathbf{b}'$: 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. In practice, $\mathsf{getProbe}$ often directly outputs the answer from $\mathcal{O}_{\mathcal{B}}$.
- $\mathsf{BioCompare}(\mathbf{b}, \mathbf{b}') \to s$: Given a biometric template $\mathbf{b}$ from $\mathsf{getEnroll}$ and another template $\mathbf{b}'$ from $\mathsf{getProbe}$, 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 2.2

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

Definition 2 (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.

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

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

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

Definition 3 (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 FP to be negligible. However, due to the inherent nature of biometrics, there might be a non-zero false negative rate $1 - \mathsf{TP} > 0$ and a FP that is not negligible. Our security model and analysis also take these possibilities into consideration.

2.2 Cryptographic Layer

In this work, we add a cryptographic layer on top of a biometric authentication scheme to protect privacy of users.

Definition 4 (Cryptographic Layer)**.** The *cryptographic layer* of a biometric authentication scheme associated with a family $\mathbb{B}$ of biometric distributions is composed of the following algorithms.

- $\mathsf{Setup}(1^\lambda) \to \mathsf{esk}, \mathsf{psk}, \mathsf{csk}$: 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: A cryptographic layer 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

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

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.

We provide two instantiations of a biometric authentication scheme with the cryptographic layer in Sections 4.1 and 5.1.

3 Security Games

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

3.1 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: $\mathcal{B} \leftarrow_\$ \mathbb{B}, \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}$
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: $\tilde{\mathbf{z}} \leftarrow \mathcal{A}(\mathsf{option})$
6: **if** $\tilde{\mathbf{z}}$ is equal to any output of $\mathcal{O}_{\mathsf{Probe}}$ **then**
7:     **return** 0
8: **end if**
9: $s \leftarrow \mathsf{Compare}(\mathsf{csk}, \mathbf{c_x}, \tilde{\mathbf{z}})$
10: **return** $\mathsf{Verify}(s)$

The auxiliary information option can be nothing or include $\mathsf{esk}, \mathsf{psk}, \mathsf{csk}, \mathbf{c_x}$ 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 $\mathcal{O}_{\mathsf{Probe}}$.
- $\mathcal{O}_{\log}(\mathsf{csk}, \mathbf{c_x}, \cdot)$: 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 $\mathcal{O}_{\mathcal{B}}$. 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 $\mathcal{O}'_{\mathsf{Probe}}(\mathsf{psk})$.

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.

Choice of option Consider a UF adversary $\mathcal{A}$ in Algorithm 2. The option includes either psk or $\mathcal{O}_{\mathsf{Probe}}$. The advantage of $\mathcal{A}$ is

$$\begin{aligned}
&\Pr[\mathsf{UF}_{\Pi,\mathbb{B},\mathsf{option}}(\mathcal{A}) \to 1] \\
=\ &\Pr[\mathsf{Verify}(\mathsf{Compare}(\mathsf{csk}, \mathbf{c_x}, \mathbf{c_y})) = 1] \\
=\ &\Pr\left[\begin{array}{l} \mathcal{B} \leftarrow_{\$} \mathbb{B}, \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B} \\ \mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}}}() \\ \mathcal{B}' \leftarrow_{\$} \mathbb{B} \\ \mathbf{b}' \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}'}}() \end{array} : \mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}, \mathbf{b}')) = 1\right]. \\
=\ &\mathsf{FP}.
\end{aligned}$$

If psk and $\mathcal{O}_{\mathcal{B}}$ are given at the same time, the adversary can even win with a probability TP by returning a probe from the distribution $\mathcal{B}$. To prevent such trivial attacks, we add the following requirements:

- option only includes psk when FP is negligible.
- The adversary is not allowed to return what $\mathcal{O}_{\mathsf{Probe}}$ returns.
- option cannot include both psk and $\mathcal{O}_{\mathcal{B}}$.

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** $\mathbf{c_y}$

UF Security with Digital Signature We note that we can achieve UF security by a similar approach in [EM23] with a digital signature scheme. Given any authentication scheme Π and an sEUF-CMA secure 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 $\mathsf{esk}' \leftarrow \mathsf{esk}$, $\mathsf{psk}' \leftarrow (\mathsf{psk}, \mathsf{sk_{Sig}})$, $\mathsf{csk}' \leftarrow \mathsf{csk}$.
- Enroll': The same as Enroll.
- $\mathsf{Probe}'(\mathsf{psk}', \mathbf{b}')$: Run $\mathbf{c_y} \leftarrow \mathsf{Probe}(\mathsf{psk}, \mathbf{b}')$ and $\sigma \leftarrow \mathsf{Sig.Sign}(\mathsf{sk_{Sig}}, \mathbf{c_y})$. Output $\mathbf{c_y}' \leftarrow (\mathbf{c_y}, \sigma)$.
- $\mathsf{Compare}'(\mathsf{csk}, \mathbf{c_x}, \mathbf{c_y}')$: If $\mathsf{Sig.Verify}(\mathsf{pk_{Sig}}, \mathbf{c_y}, \sigma) = 1$, output $\mathsf{Compare}(\mathsf{csk}, \mathbf{c_x}, \mathbf{c_y})$; otherwise, output $\perp$.

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

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

Note that Theorem 1 also holds for the following trivial authentication scheme for any biometric layer.

- $\mathsf{Setup}(1^\lambda)$: $\mathsf{esk} = \mathsf{psk} = \mathsf{csk}$ are all empty strings.
- $\mathsf{Enroll}(\mathsf{esk}, \mathbf{b}) \to \mathbf{b}$.
- $\mathsf{Probe}(\mathsf{psk}, \mathbf{b}') \to \mathbf{b}'$.
- $\mathsf{Compare}(\mathsf{csk}, \mathbf{c}, \mathbf{c}') = \mathsf{BioCompare}(\mathbf{c}, \mathbf{c}')$.

3.2 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 $\mathcal{B}^{(1)}$ and option that depends on our threat model. It tries to guess from either $\mathcal{B}^{(0)}$ or $\mathcal{B}^{(1)}$ the enrollment or probe messages are generated. The whole game $\mathsf{IND}_{\Pi,\mathbb{B},\mathsf{option}}$ is defined in Algorithm 3.

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

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

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: $\tilde{b} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{B}^{(0)}}, \mathcal{O}_{\mathcal{B}^{(1)}}}(\mathsf{option})$
8: **return** $1_{\tilde{b}=b}$

- $\mathcal{O}_{\mathbf{c_y}}$: It first samples a biometric sample $\mathbf{b}' \leftarrow_\$ \mathsf{getProbe}^{\mathcal{B}^{(b)}}()$ and outputs $\mathbf{c_y} \leftarrow \mathsf{Probe}(\mathsf{psk}, \mathbf{b}')$.
- $\mathcal{O}_{\mathsf{Enroll}}(\mathsf{esk}, \cdot)$: 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 $\mathsf{Probe}(\mathsf{psk}, \mathbf{b}')$.
- $\mathcal{O}_{\mathsf{CompVrfy}}(\mathsf{csk}, \cdot, \cdot)$: On input $\mathbf{c}$ and $\mathbf{c}'$, it first computes $s \leftarrow \mathsf{Compare}(\mathsf{csk}, \mathbf{c}, \mathbf{c}')$ and outputs $\mathsf{Verify}(s)$.

Note that at least one of $\mathbf{c_x}$ and $\mathcal{O}_{\mathbf{c_y}}$ should be given; otherwise, IND security is trivial.

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

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

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

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

Trivial Attacks We note that if $\mathsf{TP} - \mathsf{FP} > \frac{1}{\mathsf{poly}}$, there are trivial attacks.

Theorem 2. *Given any distribution family $\mathbb{B}$ that $\mathsf{TP} - \mathsf{FP} > \frac{1}{\mathsf{poly}}$, Π is not option-IND secure for*

- $\mathsf{option} = \{\mathbf{c_x},$ *either* csk *or* $\mathcal{O}_{\mathsf{CompVrfy}},$ *either* psk *or* $\mathcal{O}_{\mathsf{Probe}}\}$
- $\mathsf{option} = \{\mathcal{O}_{\mathbf{c_y}},$ *either* csk *or* $\mathcal{O}_{\mathsf{CompVrfy}},$ *either* esk *or* $\mathcal{O}_{\mathsf{Enroll}}\}$

Proof. Let option be the first case that includes $\mathbf{c_x}$. The second case when $\mathcal{O}_{\mathbf{c_y}}$ is given can be analyzed analogously. Consider the adversary $\mathcal{A}$ in the $\mathsf{IND}_{\mathsf{option}}$ game in Algorithm 4. 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,

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

□

Algorithm 4 $\mathcal{A}^{\mathcal{O}_{\mathcal{B}^{(0)}},\mathcal{O}_{\mathcal{B}^{(1)}}}(\mathsf{psk},\mathsf{csk},\mathbf{c_x})$ or $\mathcal{A}^{\mathcal{O}_{\mathcal{B}^{(0)}},\mathcal{O}_{\mathcal{B}^{(1)}},\mathcal{O}_{\mathsf{Probe}},\mathcal{O}_{\mathsf{CompVrfy}}}(\mathbf{c_x})$

1: $\mathbf{b}^{(0)} \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}^{(0)}}}()$
2: $\mathbf{c_y}^{(0)} \leftarrow \mathsf{Probe}(\mathsf{psk},\mathbf{b}^{(0)})$ ▷ or $\mathbf{c_y}^{(0)} \leftarrow \mathcal{O}_{\mathsf{Probe}}(\mathbf{b}^{(0)})$
3: $r \leftarrow \mathsf{Verify}(\mathsf{Compare}(\mathsf{csk},\mathbf{c_x},\mathbf{c_y}^{(0)}))$ ▷ or $r \leftarrow \mathcal{O}_{\mathsf{CompVrfy}}(\mathbf{c_x},\mathbf{c_y}^{(0)})$
4: **return** $1-r$

Necessity of IND Security Recall the trivial authentication scheme for any biometric layer we introduced in Section 3.1. By Theorem 1, we can obtain an authentication scheme Π' that is option-UF secure 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 option-IND secure for an option that includes either $\mathbf{c_x}$ or $\mathcal{O}_{\mathbf{c_y}}$ when $\mathsf{TP}-\mathsf{FP} > \frac{1}{\mathsf{poly}}$.

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

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

where $\mathbf{b}^*_\mathcal{B} \in \{0,1\}^k$ 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 $\mathsf{BioCompare}(\mathbf{b},\mathbf{b}') \to 1_{\mathsf{HD}(\mathbf{b},\mathbf{b}')\le\tau}$. Then

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

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 $\mathbf{b} \oplus \mathbf{r}$.
- $\mathsf{Probe}(\mathsf{psk},\mathbf{b}')$: Output $\mathbf{b}' \oplus \mathbf{r}$.
- $\mathsf{Compare}(\mathsf{csk},\mathbf{c_x},\mathbf{c_y})$: If $\mathsf{HD}(\mathbf{c_x},\mathbf{c_y}) \le \tau$, return 1; otherwise, return 0.

The correctness of Π holds by

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

Theorem 3. *Let* $\mathsf{option} = \{\mathsf{csk},\mathbf{c_x},\mathcal{O}_{\mathbf{c_y}}\}$. *The authentication scheme* Π *is* option-*IND secure.*

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. Given any adversary, assume that the number of its queries to $\mathcal{O}_{\mathbf{c_y}}$ is bounded by t. For any $\mathbf{v}, \mathbf{v}^{(1)}, \cdots, \mathbf{v}^{(t)}$,

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

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

□

IND Security with Public-Key Encryption We note that we can achieve IND security with a public-key encryption scheme. We first recall the *multi-challenge IND-CPA* security for a public-key encryption scheme.

Definition 5 (Multi-challenge IND-CPA)**.** A public-key encryption scheme $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ is multi-challenge IND-CPA secure if for any adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in the game $\mathsf{MC-IND-CPA_{PKE}}$ is

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

Algorithm 5 $\mathsf{MC\text{-}IND\text{-}CPA_{PKE}}(\mathcal{A})$

```
1: b ←$ {0,1}
2: (sk_PKE, pk_PKE) ← PKE.KeyGen(1^λ)
3: b̃ ← A^{O_MC-Enc}(pk_PKE)
4: return 1_{b̃=b}
```

- $\mathcal{O}_{\mathsf{MC\text{-}Enc}}(\cdot,\cdot)$: On input $(\mathsf{m}_0, \mathsf{m}_1)$, it outputs $\mathsf{PKE.Enc}(\mathsf{pk_{PKE}}, \mathsf{m}_b)$

Given any authentication scheme Π and a multi-challenge IND-CPA secure public-key encryption scheme $\mathsf{PKE} = (\mathsf{PKE.KeyGen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$, consider the following scheme Π'.

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

- $\mathsf{Enroll}'(\mathsf{esk}', \mathbf{b})$: Run $\mathbf{c_x} \leftarrow \mathsf{Enroll}(\mathsf{esk}, \mathbf{b})$ and $\mathsf{ct}_\mathbf{x} \leftarrow \mathsf{PKE.Enc}(\mathsf{pk_{PKE}}, \mathbf{c_x})$. Output $\mathbf{c_x}' \leftarrow \mathsf{ct}_\mathbf{x}$.

- $\mathsf{Probe}'(\mathsf{psk}', \mathbf{b}')$: Run $\mathbf{c_y} \leftarrow \mathsf{Probe}(\mathsf{psk}, \mathbf{b}')$ and $\mathsf{ct}_\mathbf{y} \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}_\mathsf{PKE}, \mathbf{c_y})$. Output $\mathbf{c_y}' \leftarrow \mathsf{ct}_\mathbf{y}$.
- $\mathsf{Compare}'(\mathsf{csk}', \mathbf{c_x}', \mathbf{c_y}')$: First decrypt $\mathbf{c_x} \leftarrow \mathsf{PKE.Dec}(\mathsf{sk}_\mathsf{PKE}, \mathbf{c_x}')$ and $\mathbf{c_y} \leftarrow \mathsf{PKE.Dec}(\mathsf{sk}_\mathsf{PKE}, \mathbf{c_y}')$. Output $\mathsf{Compare}(\mathsf{csk}, \mathbf{c_x}, \mathbf{c_y})$.

Theorem 4. *Let* $\mathsf{option} = \{\mathsf{esk}, \mathsf{psk}, \mathbf{c_x}, \mathcal{O}_{\mathbf{c_y}}\}$. *For any authentication scheme* Π, Π' *is* option*-IND secure.*

Proof. Given an adversary $\mathcal{A}$ in the $\mathsf{IND}_\mathsf{option}$ game, consider the reduction adversary $\mathcal{R}$ in Algrothm 6 which plays the $\mathsf{MC\text{-}IND\text{-}CPA}_\mathsf{PKE}$ game by running $\mathcal{A}$. $\mathcal{R}$ simulates $\mathcal{O}_{\mathbf{c_y}}$ by the following steps.

1. Sample $\mathbf{b}'^{(0)} \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}^{(0)}}}()$ and $\mathbf{b}'^{(1)} \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}^{(1)}}}()$.
2. Run $\mathbf{c_y}^{(0)} \leftarrow \mathsf{Probe}(\mathsf{psk}, \mathbf{b}'^{(0)})$ and $\mathbf{c_y}^{(1)} \leftarrow \mathsf{Probe}(\mathsf{psk}, \mathbf{b}'^{(1)})$
3. Output $\mathcal{O}_{\mathsf{MC\text{-}Enc}}(\mathbf{c_y}^{(0)}, \mathbf{c_y}^{(1)})$.

Algorithm 6 $\mathcal{R}^{\mathcal{O}_{\mathsf{MC\text{-}Enc}}}(\mathsf{pk}_\mathsf{PKE})$

1: $\mathcal{B}^{(0)} \leftarrow_\$ \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}^{(0)}$
2: $\mathcal{B}^{(1)} \leftarrow_\$ \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}^{(1)}$
3: $\mathsf{esk}, \mathsf{psk}, \mathsf{csk} \leftarrow \mathsf{Setup}(1^\lambda)$
4: $\mathbf{b}^{(0)} \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}^{(0)}}}(), \mathbf{c_x}^{(0)} \leftarrow \mathsf{Enroll}(\mathsf{esk}, \mathbf{b}^{(0)})$
5: $\mathbf{b}^{(1)} \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}^{(1)}}}(), \mathbf{c_x}^{(1)} \leftarrow \mathsf{Enroll}(\mathsf{esk}, \mathbf{b}^{(1)})$
6: $\mathbf{c_x}' \leftarrow \mathcal{O}_{\mathsf{MC\text{-}Enc}}(\mathbf{c_x}^{(0)}, \mathbf{c_x}^{(1)})$
7: $\mathsf{esk}' \leftarrow (\mathsf{esk}, \mathsf{pk}_\mathsf{PKE}), \mathsf{psk}' \leftarrow (\mathsf{psk}, \mathsf{pk}_\mathsf{PKE})$
8: $\tilde{b} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{B}^{(0)}}, \mathcal{O}_{\mathcal{B}^{(1)}}, \mathcal{O}_{\mathbf{c_y}}}(\mathsf{esk}', \mathsf{psk}', \mathbf{c_x}')$
9: **return** $\tilde{b}$

Since $\mathcal{R}$ simulates an $\mathsf{IND}_\mathsf{option}$ game for $\mathcal{A}$, the advantage of $\mathcal{A}$ is

$$\mathbf{Adv}^{\mathsf{IND}}_{\Pi,\mathbb{B},\mathcal{A},\mathsf{option}} = \mathbf{Adv}^{\mathsf{MC\text{-}IND\text{-}CPA}}_{\mathsf{PKE},\mathcal{R}} = \mathsf{negl}.$$

□

4 Security Analysis: fh-IPFE-based Instantiation

Definition 6 (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.

- $\mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{x}) \to \mathsf{sk}_{\mathbf{x}}$: It generates the functional decryption key $\mathsf{sk}\mathbf{x}$ for an input vector $\mathbf{x} \in \mathbb{F}^k$.
- $\mathsf{FE.Enc}(\mathsf{msk}, \mathsf{pp}, \mathbf{y}) \to \mathsf{ct}_{\mathbf{y}}$: It encrypts the input vector $\mathbf{y} \in \mathbb{F}^k$ to the ciphertext $\mathsf{ct}_{\mathbf{y}}$.
- $\mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}_{\mathbf{x}}, \mathsf{ct}_{\mathbf{y}}) \to z$: It outputs a value $z \in \mathbb{F}$ or an error symbol $\perp$.

Correctness An fh-IPFE scheme FE is *correct* if for all non-zero $\mathbf{x}, \mathbf{y} \in \mathbb{F}^k \setminus \{\mathbf{0}\}$, let $(\mathsf{msk}, \mathsf{pp}) \leftarrow \mathsf{FE.Setup}(1^\lambda)$, we have

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

4.1 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. 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 $\mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{x}) \to \mathsf{sk}_{\mathbf{x}}$ and outputs $\mathbf{c}_{\mathbf{x}} \leftarrow \mathsf{sk}_{\mathbf{x}}$.
- $\mathsf{Probe}(\mathsf{psk}, \mathbf{b}')$: 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 $\mathsf{FE.Enc}(\mathsf{msk}, \mathsf{pp}, \mathbf{y}) \to \mathsf{ct}_{\mathbf{y}}$ and outputs $\mathbf{c}_{\mathbf{y}} \leftarrow \mathsf{ct}_{\mathbf{y}}$.
- $\mathsf{Compare}(\mathsf{csk}, \mathbf{c}_{\mathbf{x}}, \mathbf{c}_{\mathbf{y}})$: It calls $\mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c}_{\mathbf{x}}, \mathbf{c}_{\mathbf{y}}) \to s$ and outputs the value s.

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

$$s = \mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c}_{\mathbf{x}}, \mathbf{c}_{\mathbf{y}}) = \langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^{k} -2b_i b'_i + \|\mathbf{b}\|^2 + \|\mathbf{b}'\|^2 = \|\mathbf{b} - \mathbf{b}'\|^2.$$

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 $\langle \mathbf{x}, \mathbf{y}' \rangle$ 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 $\mathbf{c_{y^*}}$ such that $\langle \mathbf{x}, \mathbf{y}^* \rangle \leq \tau$ with a good probability to impersonate the user by sampling many times offline.

We note that the construction in [EM23] is applying Theorem 1 on this instantiation. The user holds esk and psk while the server holds csk, the public parameter of the functional encryption scheme.

Let Π be an authentication scheme instantiated by an fh-IPFE scheme FE for a field $\mathbb{F} = \mathbb{Z}_q$. In the following, we discuss the UF and IND security of Π in this section. For this, we first define two security notions of FE[1].

4.2 fh-IND Security of FE

Given an fh-IPFE scheme FE, we define the $\mathsf{fh\text{-}IND}$ game [Kim+16] in Algorithm 7.

Algorithm 7 $\mathsf{fh\text{-}IND}_{\mathsf{FE}}(\mathcal{A})$

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

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

Definition 7 (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 $\forall i \in [Q_K], \forall j \in [Q_E]$, we have

$$\left\langle \mathbf{x}_i^{(0)}, \mathbf{y}_j^{(0)} \right\rangle = \left\langle \mathbf{x}_i^{(1)}, \mathbf{y}_j^{(1)} \right\rangle$$

Definition 8 (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 7 is

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

[1]In security definition, the vectors lie in $\mathbb{Z}_q^k$, but we consider $\mathbb{Z}_q^{k+2}$ when discussing the instantiation Π.

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 secure. However, fh-IND security may not be sufficient for the UF security of the instantiation in Section 4.1.

Theorem 5. *An instantiation Π using the construction in [Kim+16] is not* option*-UF secure 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.

□

4.3 RUF Security of FE

We also define the $\mathsf{RUF}^{\mathcal{O}}_{\mathsf{FE}}$ game in Algorithm 8.

Algorithm 8 $\mathsf{RUF}^{\mathcal{O}}_{\mathsf{FE}}(\mathcal{A})$

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

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

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

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

4.3.1 Achievability of RUF Security

We note that RUF security is a new security notion of fh-IPFE but can be achieved with a digital signature scheme. Let $\mathsf{Sig} = (\mathsf{Sig.KeyGen}, \mathsf{Sig.Sign}, \mathsf{Sig.Verify})$ be an sEUF-CMA signature scheme. By adding Sig, an fh-IPFE scheme FE can be upgraded to an RUF scheme FE'.

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

Theorem 6. *For any fh-IPFE FE, FE' is an RUF secure fh-IPFE.*

Proof. Given an adversary $\mathcal{A}$ in the $\mathsf{RUF}_{\mathsf{FE}'}^{\mathcal{O}'_{\mathsf{KeyGen}}, \mathcal{O}'_{\mathsf{Enc}}}$ game, consider the reduction adversary $\mathcal{R}$ in Algorithm 9 which plays the $\mathsf{sEUF\text{-}CMA}$ game of Sig. $\mathcal{R}$ is given a verification public key $\mathsf{pk}_{\mathsf{Sig}}$ and a signing oracle $\mathcal{O}_{\mathsf{Sig}}$ and returns a forged message-signature pair that is not equal to any previous answer of $\mathcal{O}_{\mathsf{Sig}}$. To run $\mathcal{A}$, $\mathcal{R}$ simulates each oracle in the following way.

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

Algorithm 9 $\mathcal{R}^{\mathcal{O}_{\mathsf{Sign}}}(\mathsf{pk}_{\mathsf{Sig}})$

```
1: r ← F^k
2: msk, pp ← FE.Setup(1^λ)
3: sk_r ← FE.KeyGen(msk, pp, r)
4: pp' ← (pp, pk_Sig)
5: z̃ ← A^{O'_KeyGen, O'_Enc}(pp', sk_r)
6: Parse (ct_z, σ') ← z̃
7: return (ct_z, σ')
```

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

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

$\square$

4.4 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 7. *Let $\mathsf{option} = \{\mathsf{csk}, \mathbf{c}_{\mathbf{x}}, \mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Enroll}}\}$. For any distribution family $\mathbb{B}$, if FE is fh-IND secure and $\mathcal{O}'_{\mathsf{KeyGen}}$-RUF secure, then Π is option-UF secure.*

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 $\mathsf{fh\text{-}IND}$ 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 $\mathsf{fh\text{-}IND}$ game. Note that since $\mathcal{R}$ never calls $\mathcal{O}_{\mathsf{Enc}}$, it 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: $\mathsf{sk} \leftarrow \mathcal{O}_{\mathsf{KeyGen}}(\mathbf{x}, \mathbf{r})$
6: $\tilde{\mathbf{z}} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Enroll}}}(\mathsf{pp}, \mathsf{sk})$
7: $s \leftarrow \mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}, \tilde{\mathbf{z}})$
8: **if** $\mathsf{Verify}(s) = 1$ **then**
9:     **return** $\tilde{b} = 0$
10: **else**
11:     **return** $\tilde{b} \leftarrow_{\$} \{0, 1\}$
12: **end if**

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 11 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.

Algorithm 11 $\mathcal{A}'^{\mathcal{O}'_{\mathsf{KeyGen}}}(\mathsf{pp}, \mathsf{sk}_{\mathbf{r}})$

$\mathcal{B} \leftarrow_{\$} \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}$
$\tilde{\mathbf{z}} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Enroll}}}(\mathsf{pp}, \mathsf{sk}_{\mathbf{r}})$
return $\tilde{\mathbf{z}}$

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 $\mathsf{Verify}(s) = 1$ in Line 8 is smaller than $\Pr[s \neq \perp] = \Pr[\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{KeyGen}}}(\mathcal{A}') \to 1]$

In conclusion,

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

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 $\mathbf{Adv}_{\mathsf{FE},\mathcal{A}'}^{\mathsf{RUF},\mathcal{O}'_{\mathsf{KeyGen}}} = \Pr[\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{KeyGen}}}(\mathcal{A}') \to 1]$ are negligilbe,

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

□

For option that includes $\mathcal{O}_{\mathsf{Probe}}$, we first note that for any $d \in \mathbb{Z}_q$ and any non-zero vector $\mathbf{r} \in \mathbb{Z}_q^{k+2} \setminus \{\mathbf{0}\}$, there exists a vector $\mathbf{y} \in \mathbb{Z}_q^{k+2}$ such that $\langle \mathbf{r}, \mathbf{y} \rangle = d$.

Theorem 8. *Let $\mathsf{option} = \{\mathsf{csk}, \mathbf{c}_{\mathbf{x}}, \mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Probe}}\}$. For any distribution family $\mathbb{B}$, if FE is fh-IND secure and $\mathcal{O}'_{\mathsf{Enc}}$-RUF secure, then Π is option-UF secure.*

Proof. [There exists a flaw in this proof. Line 7 in Algorithm 13 is not feasible for current fh-IPFE constructions. I am now trying to solve this problem. Details are in Appendix E]

Given an adversary $\mathcal{A}$ in the $\mathsf{UF}_{\mathsf{option}}$ game, consider the reduction adversary $\mathcal{R}$ in Algorithm 12 which plays the $\mathsf{fh\text{-}IND}$ 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 $d \leftarrow \langle \mathbf{x}, \mathbf{y}' \rangle$ and finds a vector $\mathbf{y}''$ such that $\langle \mathbf{r}, \mathbf{y}'' \rangle = d$. Finally, it calls $\mathcal{O}_{\mathsf{Enc}}(\mathbf{y}', \mathbf{y}'')$, which is given by the $\mathsf{fh\text{-}IND}$ 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 $\langle \mathbf{x}, \mathbf{y}' \rangle = \langle \mathbf{r}, \mathbf{y}'' \rangle$. Hence, $\mathcal{R}$ 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 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 13 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 $d \leftarrow \langle \mathbf{x}^{(*)}, \mathbf{y}' \rangle$ and finds a vector $\mathbf{y}''$ such that $\langle \mathbf{r}, \mathbf{y}'' \rangle = d$. 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.

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

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: $\mathsf{sk} \leftarrow \mathcal{O}_{\mathsf{KeyGen}}(\mathbf{x}, \mathbf{r})$
6: $\tilde{\mathbf{z}} \leftarrow \mathcal{A}^{\mathcal{O}_\mathcal{B},\mathcal{O}_{\mathsf{Probe}}}(\mathsf{pp}, \mathsf{sk})$
7: **if** $\tilde{\mathbf{z}}$ is equal to any output of $\mathcal{O}_{\mathsf{Probe}}$ **then**
8:     **return** $\perp$
9: **end if**
10: $s \leftarrow \mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}, \tilde{\mathbf{z}})$
11: **if** $\mathsf{Verify}(s) = 1$ **then**
12:     **return** $\tilde{b} = 0$
13: **else**
14:     **return** $\tilde{b} \leftarrow_\$ \{0, 1\}$
15: **end if**

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

Let $\mathcal{A}'$ play a tweaked $\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{Enc}}}$ 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 $\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{Enc}}}$ game and the tweaked one.

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

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

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

Algorithm 13 $\mathcal{A}'^{\mathcal{O}'_{\mathsf{Enc}}}(\mathsf{pp}, \mathsf{sk}_{\mathbf{r}})$

$\mathcal{B} \leftarrow_{\$} \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}$
$\mathbf{b}^{(*)} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}}}()$
$\mathbf{x}^{(*)} \leftarrow (b_1^{(*)}, \cdots, b_k^{(*)}, 1, \|\mathbf{b}^{(*)}\|^2)$
Sample $k+2$ linearly independent vectors $\{\mathbf{e}^{(i)}\}_{i=1}^{k+2}$.
for $i = 1$ to $k+2$ **do**
  $\mathsf{ct}^{(i)} \leftarrow \mathcal{O}'_{\mathsf{Enc}}(\mathbf{e}^{(i)})$.
  $d_i \leftarrow \mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}_{\mathbf{r}}, \mathsf{ct}^{(i)})$.
end for
Find the vector $\mathbf{r}$ by solving the linear system $\{\langle \mathbf{r}, \mathbf{e}^{(i)} \rangle = d_i\}_{i=1}^{k+2}$.
if $\mathbf{r} = \mathbf{0}$ **then**
  return $\perp$
end if
$\tilde{\mathbf{z}} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Probe}}}(\mathsf{pp}, \mathsf{sk}_{\mathbf{r}})$
return $\tilde{\mathbf{z}}$

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

Note that each $\mathsf{ct}^{(i)}$ is an encryption of of some uniform non-zero 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, $\Pr[\tilde{\mathbf{z}} = \mathsf{ct}^{(i)}] \leq \frac{1}{q^{k+2}-1}$ and

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

Combining both results from Equation 1 and 2, we derive

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

Finally, similar to the proof of Theorem 7, we derive

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

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

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

□

Unfortunately, for the instantiation in Section 4.1, we cannot achieve UF security when the adversary has $\textsf{psk}$, even if the false positive rate is negligible. The adversary can simply compute $\mathbf{c} \leftarrow \textsf{Probe}(\textsf{psk}, \mathbf{0})$ and return $\mathbf{c}$. The same results also hold for an $\textsf{option}$ that includes $\textsf{esk}$ since both $\textsf{psk}$ and $\textsf{esk}$ are equal to $\textsf{msk}$ and allow the adversary to run $\textsf{FE.Enc}(\textsf{msk}, \textsf{pp}, \mathbf{v})$ for any vector $\mathbf{v}$. We state this result formally in the following theorem.

Theorem 9. *Let $\textsf{option}$ include $\textsf{esk}$ or $\textsf{psk}$. For any distribution family $\mathbb{B}$ and functional encryption $\textsf{FE}$, Π is not $\textsf{option}$-UF secure.*

4.5 UF security with [Kim+16]

In this section, we show that we can achieve a concrete UF secure scheme by the fh-IPFE scheme given in [Kim+16]. In a high-level overview, two main reasons make the scheme not UF secure:

- The construction [Kim+16] allows anyone to generate a ciphertext that corresponds to a zero vector, which is described in Theorem 5. This makes the fh-IPFE scheme not even $\emptyset$-RUF secure.
- With the master secret key $\textsf{msk}$, one can reconstruct the vector $\mathbf{x}$ that a secret key $\textsf{sk}_{\mathbf{x}}$ corresponds to or even submit an encryption $\textsf{ct}_{\mathbf{y}}$ of a small vector $\mathbf{y}$. This is a drawback of all the schemes instantiated by fh-IPFE in the way described in Section 4.1. Therefore, we need to instantiate the authentication scheme using a different way.

  [I haven't found a solution for this problem yet, but I am now trying a few possible ways.]

We first upgrade the scheme in [Kim+16], which is not $\emptyset$-RUF secure due to Theorem 5, to $\mathcal{O}'_{\textsf{KeyGen}}$-RUF secure. Firstly, for an fh-IPFE scheme $\textsf{FE}$ over the field $\mathbb{Z}_q$, consider the following transformation $\textsf{FE}'$:

- $\textsf{FE}'.\textsf{Setup}$: The same as $\textsf{FE.Setup}$.
- $\textsf{FE}'.\textsf{KeyGen}(\textsf{pp}, \textsf{msk}, \mathbf{x})$: Sample $\sigma \leftarrow_{\$} \mathbb{Z}_q$. Return $\textsf{sk}_{\mathbf{x}\|\sigma} \leftarrow \textsf{FE.KeyGen}(\textsf{msk}, \textsf{pp}, (\mathbf{x}\|\sigma))$ and σ, where $\mathbf{x}\|\sigma$ is appending σ to $\mathbf{x}$.
- $\textsf{FE}'.\textsf{Enc}(\textsf{pp}, \textsf{msk}, \mathbf{y})$: Return $\textsf{ct}_{\mathbf{y}\|1} \leftarrow \textsf{FE.Enc}(\textsf{msk}, \textsf{pp}, (\mathbf{y}\|1))$, where $\mathbf{y}\|1$ is appending the constant 1 to $\mathbf{y}$.
- $\textsf{FE}'.\textsf{Dec}(\textsf{pp}, \textsf{sk}_{\mathbf{x}\|\sigma}, \sigma, \mathbf{ct}_{\mathbf{y}\|1})$: Return $\textsf{FE.Dec}(\textsf{pp}, \textsf{sk}_{\mathbf{x}\|\sigma}, \mathbf{c}_{\mathbf{y}\|1}) - \sigma \bmod q$.

One can show that if FE is fh-IND secure, FE' is also fh-IND secure. Moreover, for the concrete construction [Kim+16], we prove that FE' is also $\mathcal{O}'_{\mathsf{KeyGen}}$-RUF secure.

Theorem 10. *Let* FE *be the scheme described in [Kim+16]. If* FE *is fh-IND secure, then* FE' *is* $\mathcal{O}'_{\mathsf{KeyGen}}$*-RUF secure.*

Recall that FE is proven fh-IND secure in the generic group model. In addition, the decryption of FE requires speciflying a polynomially-bounded set S. It searches $s \in S$ such that $D_1^s = D_2$, where D_1 and D_2 are elements in a group $\mathbb{G}_T$ of order q that is in exponential of λ. If no such s is found, the decryption returns $\perp$.

Proof. Given an adversary $\mathcal{A}$ in the $\mathsf{RUF}_{\mathsf{FE}'}^{\mathcal{O}'_{\mathsf{KeyGen}}}$ game, consider the reduction adversary $\mathcal{R}$ in Algorithm 14 which plays the $\mathsf{fh\text{-}IND}$ game of the scheme FE. $\mathcal{R}$ simulates $\mathcal{O}'_{\mathsf{KeyGen}}(\mathbf{x}')$ by sampling $\sigma \leftarrow\$ \mathbb{Z}_q$ and querying $\mathcal{O}_{\mathsf{KeyGen}}(\mathbf{x}'\|\sigma, \mathbf{x}'\|\sigma)$.

Algorithm 14 $\mathcal{R}^{\mathcal{O}_{\mathsf{KeyGen}}, \mathcal{O}_{\mathsf{Enc}}}(\mathsf{pp})$

$\mathbf{r}^{(0)}, \mathbf{r}^{(1)} \leftarrow\$ \mathbb{Z}_q^k$
$\sigma \leftarrow\$ \mathbb{Z}_q$
$\mathsf{sk} \leftarrow \mathcal{O}_{\mathsf{KeyGen}}(\mathbf{r}^{(0)}\|\sigma, \mathbf{r}^{(1)}\sigma)$
$\tilde{\mathbf{z}} \leftarrow \mathcal{A}^{\mathcal{O}'_{\mathsf{KeyGen}}}(\mathsf{pp}, \mathsf{sk}, \sigma)$
$\mathbf{r}' \leftarrow\$ \mathbb{Z}_q^{k+1}$
$\mathsf{sk}' \leftarrow \mathcal{O}_{\mathsf{KeyGen}}(\mathbf{r}^{(0)}\|\sigma, \mathbf{r}')$
$s' \leftarrow \mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}', \tilde{\mathbf{z}}) - \sigma \bmod q$
return $1_{s'=\perp}$

In Algorithm 14, let

- $\mathsf{sk} = (g_1^{\alpha_0}, g_1^{\alpha_1}, \cdots, g_1^{\alpha_{k+1}})$.
- $\mathsf{sk}' = (g_1^{\alpha'_0}, g_1^{\alpha'_1}, \cdots, g_1^{\alpha'_{k+1}})$.
- $s = \mathsf{FE}'.\mathsf{Dec}(\mathsf{pp}, \mathsf{sk}, \tilde{\mathbf{z}}) = \mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}, \tilde{\mathbf{z}}) - \sigma \bmod q$.

Since $\mathcal{R}$ simulates a $\mathsf{RUF}_{\mathsf{FE}'}^{\mathcal{O}'_{\mathsf{KeyGen}}}$ game for $\mathcal{A}$, regardless of the challenge bit, we have

$$\Pr[s \neq S] = \Pr[s \in S] = \Pr[\mathsf{RUF}_{\mathsf{FE}'}^{\mathcal{O}'_{\mathsf{KeyGen}}}(\mathcal{A}) \to 1]$$

Moreover, if $s \neq \perp$, let $\tilde{\mathbf{z}} = (g_2^{\beta_0}, g_2^{\beta_1}, \cdots g_2^{\beta_{k+1}})$, we have

$$\prod_{i=1}^{k+1} e(g_1^{\alpha_i}, g_2^{\beta_i}) = g_T^{\sum_{i=1}^{k+1} \alpha_i \beta_i} \implies \mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}, \tilde{\mathbf{z}}) = \begin{cases} \frac{\sum_{i=1}^{k+1} \alpha_i \beta_i}{\alpha_0 \beta_0} & \text{if } \beta_0 \neq 0 \\ 0 & \text{if } \beta_0 = 0 \end{cases}$$

Similarly, $\mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}', \tilde{\mathbf{z}}) = \frac{\sum_{i=1}^{k+1} \alpha'_i \beta_i}{\alpha'_0 \beta_0}$ or 0. If the challenge bit $b = 0$, since sk and sk' correspond to the same vector $\mathbf{r}^{(0)}\|\sigma$, we have

$$\frac{\alpha'_0}{\alpha_0} = \frac{\alpha'_1}{\alpha_1} = \cdots = \frac{\alpha'_{k+1}}{\alpha_{k+1}},$$

which implies $\mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}, \tilde{\mathbf{z}}) = \mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}', \tilde{\mathbf{z}})$ and $s = s'$.

Now, we analyze the advantage of $\mathcal{R}$. If the challenge bit $b = 0$, the probability that $s \neq \perp$ is

$$\begin{aligned}\Pr[s' \neq \perp \mid b = 0] &\geq \Pr[s' \in S \mid s \in S, b = 0] \cdot \Pr[s \in S \mid b = 0] \\ &= 1 \cdot \Pr[\mathsf{RUF}_{\mathsf{FE}'}^{\mathcal{O}'_{\mathsf{KeyGen}}}(\mathcal{A}) \to 1]\end{aligned}$$

If the challenge bit $b = 1$,

$$\begin{aligned}\Pr[s' \neq \perp \mid b = 1] &\leq \Pr[s' \in S \mid s \in S, b = 1] + \Pr[s \notin S \mid b = 1] \\ &\leq \Pr[\mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}', \tilde{\mathbf{z}}) - \sigma \bmod q \in S \mid s \in S, b = 1] \\ &\quad + 1 - \Pr[\mathsf{RUF}_{\mathsf{FE}'}^{\mathcal{O}'_{\mathsf{KeyGen}}}(\mathcal{A}) \to 1]\end{aligned}$$

Note that $\mathbf{r}', \mathsf{sk}'$ and thus all α'_i are independent of $\tilde{\mathbf{z}}$ and all β_i. Hence,

$$\begin{aligned}&\Pr[\mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}', \tilde{\mathbf{z}}) - \sigma \bmod q \in S \mid s \in S, b = 1] \\ &\leq \Pr\left[\frac{\sum_{i=1}^{k+1} \alpha'_i \beta_i}{\alpha'_0 \beta_0} - \sigma \bmod q \in S \mid s \in S, b = 1, \beta_0 \neq 0\right] + \Pr\left[0 - \sigma \bmod q \in S\right] \\ &\leq \Pr\left[\frac{\sum_{i=1}^{k+1} \alpha'_i \beta_i}{\alpha'_0 \beta_0} - \sigma \bmod q \in S \mid s \in S, b = 1, \beta_0 \neq 0\right] + \Pr\left[0 - \sigma \bmod q \in S\right] \\ &\leq \frac{|S|}{q} + \frac{|S|}{q}\end{aligned}$$

In conclusion,

$$\begin{aligned}\Pr[\mathsf{fh\text{-}IND}(\mathcal{R}) \to 1] &= \frac{1}{2} \cdot \Pr[s' \neq \perp \mid b = 0] + \frac{1}{2} \cdot \Pr[s' = \perp \mid b = 1] \\ &\geq \frac{1}{2} \cdot \Pr[\mathsf{RUF}_{\mathsf{FE}'}^{\mathcal{O}'_{\mathsf{KeyGen}}}(\mathcal{A}) \to 1] + \frac{1}{2} \cdot \left(1 - \frac{2|S|}{q} - (1 - \Pr[\mathsf{RUF}_{\mathsf{FE}'}^{\mathcal{O}'_{\mathsf{KeyGen}}}(\mathcal{A}) \to 1])\right) \\ &= \frac{1}{2} \cdot \Pr[\mathsf{RUF}_{\mathsf{FE}'}^{\mathcal{O}'_{\mathsf{KeyGen}}}(\mathcal{A}) \to 1] - \frac{|S|}{q} + \frac{1}{2} \cdot \Pr[\mathsf{RUF}_{\mathsf{FE}'}^{\mathcal{O}'_{\mathsf{KeyGen}}}(\mathcal{A}) \to 1]) \\ &= \Pr[\mathsf{RUF}_{\mathsf{FE}'}^{\mathcal{O}'_{\mathsf{KeyGen}}}(\mathcal{A}) \to 1] - \frac{|S|}{q}\end{aligned}$$

Since $\mathbf{Adv}_{\mathsf{FE},\mathcal{R}}^{\mathsf{fh\text{-}IND}} = \left|\Pr[\mathsf{fh\text{-}IND}(\mathcal{R}) \to 1] - \frac{1}{2}\right|$ and $\frac{|S|}{q}$ are negligible,

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

□

To instantiate the authentication scheme Π using the way described in Section 4.1 with FE', we assume that $s \in S$ for all integer $s \leq \tau^2$, the pre-defined threshold in the biometric layer. Along with Theorem 7, we have the following result.

Corollary 1. *Let* $\mathsf{option} = \{\mathsf{csk}, \mathbf{c}_{\mathbf{x}}, \mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Enroll}}\}$*, and let* FE *be the scheme described in [Kim+16]. If* FE *is fh-IND secyre, then for any distribution family* $\mathbb{B}$*, the authentication scheme instantiated by* FE' *is* option*-UF secure.*

4.6 IND Security of Π

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

Definition 10. 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 1. 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 option-IND security when option includes $\mathsf{csk}, \mathbf{c_x}$ and $\mathcal{O}_{\mathbf{c_y}}$ 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 11. *Let $\mathsf{option} = \{\mathsf{csk}, \mathbf{c_x}, \mathcal{O}_{\mathbf{c_y}}\}$. For a distribution family $\mathbb{B}$ satisfying Assumption 1 and having a true positive rate $\mathsf{TP} > \frac{1}{\mathsf{poly}}$, if FE is fh-IND secure, then Π is option-IND secure.*

Proof. Given an adversary $\mathcal{A}$ in the $\mathsf{IND}_{\mathsf{option}}$ game, consider the reduction adversary $\mathcal{R}$ in Algorithm 15 which plays the $\mathsf{fh}\text{-}\mathsf{IND}$ game by running $\mathcal{A}$. $\mathcal{R}$ simulates $\mathcal{O}_{\mathbf{c_y}}$ by the following steps.

1. Sample $\mathbf{b'}^{(0)} \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}^{(0)}}}()$ and let $\mathbf{y}^{(0)} \leftarrow (-2{b'_1}^{(0)}, \cdots, -2{b'_k}^{(0)}, \|\mathbf{b'}^{(0)}\|^2, 1)$
2. Repeat sampling $\mathbf{b'}^{(1)} \leftarrow \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}^{(1)}}}()$ and let $\mathbf{y}^{(1)} \leftarrow (-2{b'_1}^{(1)}, \cdots, -2{b'_k}^{(1)}, \|\mathbf{b'}^{(1)}\|^2, 1)$ until $\langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle = \langle \mathbf{x}^{(1)}, \mathbf{y}^{(1)} \rangle$.
3. Return $\mathsf{ct}_{\mathbf{y}}^{(i)} \leftarrow \mathcal{O}_{\mathsf{Enc}}(\mathbf{y}^{(0)}, \mathbf{y}^{(1)})$.

Algorithm 15 $\mathcal{R}^{\mathcal{O}_{\mathsf{KeyGen}}, \mathcal{O}_{\mathsf{Enc}}}(\mathsf{pp})$

1: $\mathcal{B}^{(0)} \leftarrow_\$ \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}^{(0)}$
2: $\mathcal{B}^{(1)} \leftarrow_\$ \mathbb{B}, \quad \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}^{(1)}$
3: $\mathbf{b}^{(0)} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}^{(0)}}}(), \mathbf{x}^{(0)} \leftarrow (b_1^{(0)}, \cdots, b_k^{(0)}, 1, \|\mathbf{b}^{(0)}\|^2)$
4: $\mathbf{b}^{(1)} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}^{(1)}}}(), \mathbf{x}^{(1)} \leftarrow (b_1^{(1)}, \cdots, b_k^{(1)}, 1, \|\mathbf{b}^{(1)}\|^2)$
5: $\mathsf{sk} \leftarrow \mathcal{O}_{\mathsf{KeyGen}}(\mathbf{x}^{(0)}, \mathbf{x}^{(1)})$
6: $\tilde{b} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{B}^{(0)}}, \mathcal{O}_{\mathcal{B}^{(1)}}, \mathcal{O}_{\mathbf{c_y}}}(\mathsf{pp}, \mathsf{sk})$
7: **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 $\langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle = \langle \mathbf{x}^{(1)}, \mathbf{y}^{(1)} \rangle$. Hence, $\mathcal{R}$ is an admissible adversary.

We first show that the simulation of oracle $\mathcal{O}_{\mathbf{c_y}}$ is efficient. The probability that $\langle \mathbf{x}^{(0)}, \mathbf{y}^{(0)} \rangle = \langle \mathbf{x}^{(1)}, \mathbf{y}^{(1)} \rangle$ is satisfied is

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

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 $\mathsf{IND}_{\mathsf{option}}$ game for $\mathcal{A}$. Assume that $\mathcal{A}$ makes t queries to $\mathcal{O}_{\mathbf{c_y}}$ and receives probe messages $\{\mathbf{c_y}^{(i)}\}_{i=1}^{t} = \{\mathsf{ct}_{\mathbf{y}}^{(i)}\}_{i=1}^{t}$. If the challenge bit b of the fh-IND game is 0, $\mathbf{c_x} = \mathsf{sk}$ 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 1.

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

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

Note that for any $\{d_i\}_{i=1}^{t}, d_i > 0$,

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

Now, let $\mathbf{Y}'_i$ be the distribution of $\mathbf{y}^{(1)}$ derived in the i-th query to $\mathcal{O}_{\mathbf{c_y}}$. For any

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

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

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

In conclusion,

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

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

□

5 Security Analysis: Relational Hash-based Instantiation

Definition 11 (Relational Hash (adapted from [MR14]))**.** Let R be a relation over sets X, Y, and Z. A *relational hash* scheme RH for R 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.
- $\mathsf{RH.Hash}_1(\mathsf{pk}, \mathbf{x}) \to \mathbf{h_x}$: Given a hash key pk and $\mathbf{x} \in X$, it outputs a hash $\mathbf{h_x}$.
- $\mathsf{RH.Hash}_2(\mathsf{pk}, \mathbf{y}) \to \mathbf{h_y}$: Given a hash key pk and $\mathbf{y} \in Y$, it outputs a hash $\mathbf{h_y}$.
- $\mathsf{RH.Verify}(\mathsf{pk}, \mathbf{h_x}, \mathbf{h_y}, \mathbf{z}) \to r \in \{0,1\}$: Given a hash key pk, two hashes $\mathbf{h_x}$ and $\mathbf{h_y}$, and $\mathbf{z} \in Z$, it verifies whether the relation among $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$ holds.

Correctness A relational hash scheme RH is *correct* if $\forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in X \times Y \times Z$,

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

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

5.1 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 for the relation R^τ of Hamming distance proximity parametrized by a constant τ.

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

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),

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

The idea behing this construction is that users hold a personal device which runs Setup, Enroll, and Probe using templates coming from a biometric sensor. The message $\mathbf{c_x}$ along with the comparison key $\mathsf{csk} = \mathsf{pk}$, which is assumed to be public, are sent to a server in order to enroll the user, and $\mathbf{c_y}$ is sent on authentication.

Now, let Π be an authentication scheme instantiated by a relational hash scheme RH. We discuss the UF and IND security of Π in the following subsections.

5.2 UF Security of Π

We first recall the unforgeability [MR14] of a relational hash scheme.

Definition 12 (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.

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

In this work, since we assume the existence of a family $\mathbb{B}$ of biometric distributions and interfaces to interact with it, we extend the notion to *unforgeable for any adversary who has access to* $\mathbb{B}$.

Theorem 12. *Let $\mathsf{option} = \{\mathsf{esk}, \mathsf{psk}, \mathsf{csk}, \mathbf{c_x}\}$. If RH is unforgeable for the distribution*

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

for any adversary who has access to $\mathbb{B}$, then Π is option-UF secure.

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.

Proof. Recall that the distribution of $\mathbf{c_x}$ in the UF game of the instantiation of Section 5.1 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 3.1, an adversary with psk can enjoy a winning rate of the false positive rate FP of $\mathbb{B}$. Theorem 12 thus implies that if FP is not negligible, there does not exist an RH that is unforgeable for the distribution $\{\mathcal{B} \leftarrow_{\$} \mathbb{B} : \mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}}}()\}$ for any adversary who has access to $\mathbb{B}$.

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 3.1, 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.

5.3 IND Security of Π

For the IND security, since $\mathsf{esk}, \mathsf{psk}$ and csk are assumed to be public in this instantiation and should given in option, by applying Theorem 2, Π is not option-IND secure for any option that includes $\mathbf{c_x}$ or $\mathcal{O}_{\mathbf{c_y}}$.

Theorem 13. *Let $\mathsf{option} = \{\mathsf{esk}, \mathsf{psk}, \mathsf{csk}, \mathbf{c_x}\}$ or $\{\mathsf{esk}, \mathsf{psk}, \mathsf{csk}, \mathcal{O}_{\mathbf{c_y}}\}$. 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 option-IND secure.*

5.3.1 IND Security for a Particular Biometric Layer

Recall that in Section 3.2, we introduce as an example a particular biometric layer:

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

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 $\mathcal{B}$. With the same relational hash RH in Section 5.1, we can instantiate another authentication scheme using RH.

- $\mathsf{Setup}(1^\lambda)$: It runs $\mathsf{RH.KeyGen}(1^\lambda) \to \mathsf{pk}$ 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 $\mathsf{RH.Hash}_1(\mathsf{pk}, \mathbf{x} + \mathbf{r}) \to \mathbf{h}_\mathbf{x}$ and outputs $\mathbf{c}_\mathbf{x} \leftarrow \mathbf{h}_\mathbf{x}$.
- $\mathsf{Probe}(\mathsf{psk}, \mathbf{b}')$: Let $\mathbf{y} \leftarrow \mathbf{b}$. It calls $\mathsf{RH.Hash}_2(\mathsf{pk}, \mathbf{y} + \mathbf{r}) \to \mathbf{h}_\mathbf{y}$ and outputs $\mathbf{c}_\mathbf{y} \leftarrow \mathbf{h}_\mathbf{y}$.
- $\mathsf{Compare}(\mathsf{csk}, \mathbf{c}_\mathbf{x}, \mathbf{c}_\mathbf{y})$: It calls $\mathsf{RH.Verify}(\mathsf{pk}, \mathbf{h}_\mathbf{x}, \mathbf{h}_\mathbf{y}) \to s$ and outputs the value s.

Correctness holds because

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

With the same argument in Theorem 3, one can prove that this new scheme is now $\{\mathsf{csk}, \mathbf{c}_\mathbf{x}, \mathcal{O}_{\mathbf{c}_\mathbf{y}}\}$-IND secure, 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 13 (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.

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

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
$$\mathsf{sk}_{\mathbf{x}} = (K_1, K_2) = \left(g_1^{\alpha \cdot \det(\mathbf{B})}, g_1^{\alpha \cdot \mathbf{x} \cdot \mathbf{B}} \right)$$
- $\mathsf{FE.Enc}(\mathsf{msk}, \mathsf{pp}, \mathbf{y})$: Sample $\beta \leftarrow_\$ \mathbb{Z}_q$ and output
$$\mathsf{ct}_{\mathbf{y}} = (C_1, C_2) = \left(g_2^{\beta}, g_2^{\beta \cdot \mathbf{y} \cdot \mathbf{B}^*} \right)$$
- $\mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}_{\mathbf{x}}, \mathsf{ct}_{\mathbf{y}}) \to z$: Parse $\mathsf{sk}_{\mathbf{x}} = (K_1, K_2)$ and $\mathsf{ct}_{\mathbf{y}} = (C_1, C_2)$ 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, $(D_1)^{\langle \mathbf{x}, \mathbf{y} \rangle} = D_2$.

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 $\langle \mathbf{x}, \mathbf{y} \rangle$. For example, for the instantiation in Section 4.1, 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 γ-RUF Security of FE

Let $\mathbb{F} = \mathbb{Z}_q$. We can extend the definition of the RUF security in Section 4.3 with an integer parameter γ.

Algorithm 16 $\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O},\gamma}(\mathcal{A})$

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

Here, the game runs $1_{s\leq\gamma}$ by first viewing the field element $s \in \mathbb{Z}_q$ as a positive integer in $\{0, 1, \cdots, q-1\}$ and comparing it with γ.

The oracle $\mathcal{O}$ can be nothing or include $\mathcal{O}'_{\mathsf{KeyGen}}(\cdot)$ and $\mathcal{O}'_{\mathsf{Enc}}(\cdot)$ based on the threat model as in Section 4.3.

Definition 14 (γ-RUF Security)**.** An fh-IPFE scheme FE is called $\{\mathcal{O}, \gamma\}$-RUF secure if for any adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in the $\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O},\gamma}$ game in Algorithm 16 is

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

Note that if FE is $\mathcal{O}$-RUF secure, it is $\{\mathcal{O}, \gamma\}$-RUF secure for any integer γ.

With the extension with γ, we can rewrite our results in Section 4.

Theorem 14 (Theorem 7)**.** *Let $\mathsf{option} = \{\mathsf{csk}, \mathbf{c}_\mathbf{x}, \mathcal{O}_\mathcal{B}, \mathcal{O}_{\mathsf{Enroll}}\}$. For any distribution family $\mathbb{B}$, if FE is fh-IND secure and $\{\mathcal{O}'_{\mathsf{KeyGen}}, \gamma\}$-RUF secure for a $\gamma \geq \tau^2$, then Π is option-UF secure.*

Theorem 15 (Theorem 8)**.** *Let $\mathsf{option} = \{\mathsf{csk}, \mathbf{c}_\mathbf{x}, \mathcal{O}_\mathcal{B}, \mathcal{O}_{\mathsf{Probe}}\}$. For any distribution family $\mathbb{B}$, if FE is fh-IND secure and $\{\mathcal{O}'_{\mathsf{Enc}}, \gamma\}$-RUF secure for a $\gamma \geq \tau^2$, then Π is option-UF secure.*

B.1 Achievability of γ-RUF Security

Assumption 2. Assume that $\mathsf{FE.Dec}(\mathsf{pp}, \mathbf{c}, \mathbf{z})$ only returns when $\mathbf{z}$ corresponds to a *non-zero* vector $\mathbf{v} \in \mathbb{F}^k$. More precisely, assume that for any $\mathbf{z}$, there can only be two possibilities.

- Either there exists a vector $\mathbf{v} \in \mathbb{F}^k \setminus \{\mathbf{0}\}$ such that for any $\mathbf{x} \in \mathbb{F}^k, \mathsf{sk}_\mathbf{x} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{x})$,

$$\mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}_\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{v} \rangle.$$

- Or for any $\mathbf{x} \in \mathbb{F}^k$ and $\mathsf{sk}_{\mathbf{x}} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{x})$, $\mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}_{\mathbf{x}}, \mathbf{z}) \to \perp$.

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

Theorem 16. *Given Assumption 2. If FE is fh-IND secure, then FE is $\{\mathcal{O}'_{\mathsf{KeyGen}}, \gamma\}$-RUF secure for any $\gamma \leq (1 - \frac{1}{\mathsf{poly}}) \cdot \|\mathbb{F}\|$.*

Proof. Given an adversary $\mathcal{A}$ in the $\mathsf{RUF}_{\mathsf{FE}}^{\mathcal{O}'_{\mathsf{KeyGen}}, \gamma}$ game for a $\gamma \leq (1 - \frac{1}{P(\lambda)}) \cdot \|\mathbb{F}\|$, where $P(\lambda)$ is any polynomial. Let t be an integer, consider the reduction adversary $\mathcal{R}$ in Algorithm 17 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 2, let $\tilde{\mathbf{z}}$ correspond to a non-zero vector $\tilde{\mathbf{v}}$.

Algorithm 17 $\mathcal{R}^{\mathcal{O}_{\mathsf{KeyGen}}, \mathcal{O}_{\mathsf{Enc}}}(\mathsf{pp})$

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

If the challenge bit $b = 0$, then by Assumption 2, 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

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

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

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

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

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

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

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

$\square$

C Summary of UF and IND Security

	UF	IND
fh-IND	✗ (Thm 5)	$\{\mathsf{csk}, \mathbf{c_x}, \mathcal{O}_{\mathbf{c_y}}\}$ (Thm 11)
fh-IND, $\mathcal{O}'_{\mathsf{KeyGen}}$-RUF	$\{\mathsf{csk}, \mathbf{c_x}, \mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Enroll}}\}$ (Thm 7)	$\{\mathsf{csk}, \mathbf{c_x}, \mathcal{O}_{\mathbf{c_y}}\}$ (Thm 11)
fh-IND, $\mathcal{O}'_{\mathsf{Probe}}$-RUF	$\{\mathsf{csk}, \mathbf{c_x}, \mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Probe}}\}$ (Thm 8)	$\{\mathsf{csk}, \mathbf{c_x}, \mathcal{O}_{\mathbf{c_y}}\}$ (Thm 11)
fh-IND + sEUF-CMA Sig	$\{\mathsf{esk}, \mathsf{csk}, \mathbf{c_x}, \mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Probe}}\}$ (Thm 1)	$\{\mathsf{csk}, \mathbf{c_x}, \mathcal{O}_{\mathbf{c_y}}\}$ (Thm 11)
fh-IND + MC-IND-CPA PKE	✗ (Thm 5)	$\{\mathsf{csk}, \mathbf{c_x}, \mathcal{O}_{\mathbf{c_y}}\}$ (Thm 11) $\{\mathsf{esk}, \mathsf{psk}, \mathbf{c_x}, \mathcal{O}_{\mathbf{c_y}}\}$ (Thm 4)
fh-IND + sEUF-CMA Sig + MC-IND-CPA PKE	$\{\mathsf{esk}, \mathsf{csk}, \mathbf{c_x}, \mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Probe}}\}$ (Thm 1)	$\{\mathsf{csk}, \mathbf{c_x}, \mathcal{O}_{\mathbf{c_y}}\}$ (Thm 11) $\{\mathsf{esk}, \mathsf{psk}, \mathbf{c_x}, \mathcal{O}_{\mathbf{c_y}}\}$ (Thm 4)

Table 1: fh-IPFE. We can try to provide option-UF security with psk.

	UF	IND
Unforgeable for $\mathcal{X}$	$\{\mathsf{esk}, \mathsf{psk}, \mathsf{csk}, \mathbf{c_x}\}$ (Thm 12)	✗ (Thm 2)
Unforgeable for $\mathcal{X}$ + sEUF-CMA Sig	$\{\mathsf{esk}, \mathsf{psk}, \mathsf{csk}, \mathbf{c_x}\}$ (Thm 12) $\{\mathsf{esk}, \mathsf{csk}, \mathbf{c_x}, \mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Probe}}\}$ (Thm 1)	✗ (Thm 2)
Unforgeable for $\mathcal{X}$ + MC-IND-CPA PKE	$\{\mathsf{esk}, \mathsf{psk}, \mathsf{csk}, \mathbf{c_x}\}$ (Thm 12)	$\{\mathsf{esk}, \mathsf{psk}, \mathbf{c_x}, \mathcal{O}_{\mathbf{c_y}}\}$ (Thm 4)
Unforgeable for $\mathcal{X}$ + sEUF-CMA Sig + MC-IND-CPA PKE	$\{\mathsf{esk}, \mathsf{psk}, \mathsf{csk}, \mathbf{c_x}\}$ (Thm 12) $\{\mathsf{esk}, \mathsf{csk}, \mathbf{c_x}, \mathcal{O}_{\mathcal{B}}, \mathcal{O}_{\mathsf{Probe}}\}$ (Thm 1)	$\{\mathsf{esk}, \mathsf{psk}, \mathbf{c_x}, \mathcal{O}_{\mathbf{c_y}}\}$ (Thm 4)

Table 2: RH. We can try to provide option-IND security with csk.

D One-Way Game

Algorithm 18 $\mathsf{OW}_{\Pi,\mathbb{B},\mathsf{option}}(\mathcal{A})$

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: $s \leftarrow \mathsf{BioCompare}(\mathbf{b}, \tilde{\mathbf{b}})$
7: **return** $\mathsf{Verify}(s)$

The auxiliary information option can be nothing or include $\mathsf{esk}, \mathsf{psk}, \mathsf{csk}, \mathbf{c_x}$ or the oracles $\mathcal{O}_{\mathbf{c_y}}, \mathcal{O}_{\mathsf{Enroll}}, \mathcal{O}_{\mathsf{Probe}}, \mathcal{O}'_{\mathsf{Enroll}}, \mathcal{O}'_{\mathsf{Probe}}$. We can also consider providing $\mathbf{c_x}^{(i)} \leftarrow \mathsf{Enroll}(\mathsf{esk}_i, \mathbf{b})$ for different honest esk_i. Note that the OW game is trivial if FP is not negligible.

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

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

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

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

Its relation with UF security is as follows.

Theorem 17. *Let option include psk. If Π is option-UF secure, then Π is option-OW.*

Proof. Suppose Π is not option-OW, the adversary can derive $\tilde{\mathbf{b}}$ such that

$$\mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}, \tilde{\mathbf{b}})) = 1$$

With psk, the adversary can further generate $\tilde{\mathbf{c_x}} \leftarrow \mathsf{Probe}(\mathsf{psk}, \tilde{\mathbf{b}})$ to win the option-UF game. ☐

Assumption 3. Assume that for any $\mathcal{B}^{(0)}, \mathcal{B}^{(1)} \in \mathbb{B}$, $\mathcal{B}^{(0)} \neq \mathcal{B}^{(1)}$, and $\tilde{\mathbf{b}}$ in the domain of $\mathsf{BioCompare}$, let $\mathbf{b}^{(0)}, \mathbf{b}^{(0)\prime} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}^{(0)}}}()$ and $\mathbf{b}^{(1)\prime} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}^{(1)}}}()$, then

$$\begin{aligned}&\Pr[\mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}^{(0)\prime}, \tilde{\mathbf{b}})) = 1 \mid \mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}^{(0)}, \tilde{\mathbf{b}})) = 1] \\ &- \Pr[\mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}^{(1)}, \tilde{\mathbf{b}})) = 1 \mid \mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}^{(0)}, \tilde{\mathbf{b}})) = 1] \geq \frac{1}{\mathsf{poly}}.\end{aligned}$$

Theorem 18. *Given Assumption 3. If Π is option-IND secure, then Π is option-OW.*

Proof. Suppose Π is not option-OW, we can construct an adversary in the option-IND game that can derive $\tilde{\mathbf{b}}$ such that

$$\mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}, \tilde{\mathbf{b}})) = 1$$

where $\mathbf{b} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}^{(b)}}}()$ and b is the challenge bit. By assumption 3, this implies that if the adversary runs $\mathbf{b}^{(0)\prime} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}^{(0)}}}()$ and $\mathbf{b}^{(1)\prime} \leftarrow \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}^{(1)}}}()$, then

$$\Pr[\mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}^{(b)\prime}, \tilde{\mathbf{b}})) = 1] - \Pr[\mathsf{Verify}(\mathsf{BioCompare}(\mathbf{b}^{(1-b)\prime}, \tilde{\mathbf{b}})) = 1] \geq \frac{1}{\mathsf{poly}}.$$

By comparing the results of the two cases, the adversary can find b with a probability greater than $\frac{1}{\mathsf{poly}}$. □

We have seen some examples that authentication schemes using current constructions of fh-IPFE and RH may not be OW-secure.

fh-IPFE Suppose Π is instantiated by the construction in [Kim+16] in the way described in Section 4.1. The scheme is not $\mathcal{O}'_{\mathsf{Enroll}}$-OW. The adversary can set the fake esk' to include the matrix $\mathbf{B} = \mathbf{I}$, the size-$(k+2)$ identity matrix, and query $\mathcal{O}'_{\mathsf{Enroll}}$ to get

$$g_1^{\alpha \cdot \det(\mathbf{B})}, g_1^{\alpha \cdot \mathbf{x} \cdot \mathbf{B}} = g_1^{\alpha}, (g_1^{\alpha \cdot x_1}, g_1^{\alpha \cdot x_2}, \cdots, g_1^{\alpha \cdot x_{k+2}})$$

where α is an unknown element in $\mathbb{Z}_q$, and $\mathbf{x} = (x_1, \cdots, x_{k+2}) = (b_1, \cdots, b_k, 1, \|\mathbf{b}\|^2)$. Since b_i is bounded in $\{0, 1, \cdots, m\}$ for some fixed integer m for each i, the adversary can raise the power of g_1^{α} to retrive all the coefficients of $\mathbf{b}$. Similarly, the scheme is not $\mathcal{O}'_{\mathsf{Probe}}$-OW.

In addition, any fh-IPFE scheme instantiated in the way described in Section 4.1 is neither $\{\mathbf{c}_{\mathbf{x}}, \mathsf{esk}\}$-OW nor $\{\mathbf{c}_{\mathbf{x}}, \mathsf{psk}\}$-OW. Note that $\mathsf{esk} = \mathsf{psk} = \mathsf{msk}$, the master secret key of the fh-IPFE. With msk, the adversary can generate encryptions of any vector and find $\mathbf{x}$ by solving linear equations of inner products.

RH Suppose Π is instantiated by the construction in [MR14] in the way described in Section 5.1. The scheme is neither $\mathcal{O}'_{\mathsf{Enroll}}$-OW nor $\mathcal{O}'_{\mathsf{Probe}}$-OW. Recall that the enrollment key esk is the public key pk of the relational hash scheme. In the construction of [MR14],

- pk includes an encoding algorithm Encode of an error correcting code.
- $\mathbf{h}_{\mathbf{x}}$ includes $\mathbf{x} + \mathsf{Encode}(\mathbf{r})$ for some random $\mathbf{r} \leftarrow_{\$} \{0, 1\}^k$.

Now, the adversary can set the fake esk' to include an invalid Encode such that $\mathbf{x} + \mathsf{Encode}(\mathbf{r})$ reveals $\mathbf{x} = \mathbf{b}$.

The scheme is $\{\mathsf{esk}, \mathsf{psk}, \mathsf{csk}, \mathbf{c}_{\mathbf{x}}\}$-OW, which is given in [MR14, Theorem 4].

E Fixing the Proof of Theorem 8

In Theorem 8, we simulate the oracle $\mathcal{O}_{\mathsf{Probe}}$ in the adversary $\mathcal{A}'$ in Algorithm 13 in the $\mathsf{RUF}^{\mathcal{O}'_{\mathsf{Enc}}}$ game by the following steps:

1. Sample $k+2$ independent vectors $\mathbf{e}^{(1)}, \cdots, \mathbf{e}^{(k+2)}$.
2. For $i \in [k+2]$, $\mathsf{ct}^{(i)} \leftarrow \mathcal{O}'_{\mathsf{Enc}}(\mathbf{e}^{(i)})$.
3. For $i \in [k+2]$, $d_i \leftarrow \mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}_{\mathbf{r}}, \mathbf{c}^{(i)})$, where $\mathsf{sk}_{\mathbf{r}}$ is $\mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{r})$.
4. Find the vector $\mathbf{r}$ by solving the linear system $\{\langle \mathbf{r}, \mathbf{e}^{(i)} \rangle = d_i\}_{i=1}^{k+2}$.
5. On query $\mathcal{O}_{\mathsf{Probe}}(\mathbf{b}')$, first encode $\mathbf{b}'$ into $\mathbf{y}'$ and find $d \leftarrow \langle \mathbf{x}^*, \mathbf{y}' \rangle$. Then find a vector $\mathbf{y}''$ such that $\langle \mathbf{r}, \mathbf{y}'' \rangle = d$. Return $\mathcal{O}'_{\mathsf{Enc}}(\mathbf{y}'')$.

However, $\mathsf{FE.Dec}$ of constructions in [DDM15; TAO16; Kim+16] rely on some *prior knowledge* of the inner product. Their basic idea is to find d given g and g^d, where g is in a group of exponential size. Therefore, $\mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}_{\mathbf{r}}, \mathbf{c}^{(i)})$ will probably return $\perp$, representing that d is too large to find.

For this problem, I have three proposals:

1. Let the adversary choose $\mathbf{r}$.
2. Modify the RUF game.
3. Do nothing. Claim [DDM15; TAO16; Kim+16] are not ideal FE.

E.1 Solution I

Consider the SUF game.

Algorithm 19 $\mathsf{SUF}^{\mathcal{O}}_{\mathsf{FE}}(\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2))$

$\mathbf{r}, \mathsf{st} \leftarrow \mathcal{A}_1(1^\lambda)$
if $\mathbf{r} = \mathbf{0}$ **then**
  return $\perp$
end if
$\mathsf{msk}, \mathsf{pp} \leftarrow \mathsf{FE.Setup}(1^\lambda)$
$\mathsf{sk}_{\mathbf{r}} \leftarrow \mathsf{FE.KeyGen}(\mathsf{msk}, \mathsf{pp}, \mathbf{r})$
$\tilde{\mathbf{z}} \leftarrow \mathcal{A}_2^{\mathcal{O}}(\mathsf{st}, \mathsf{pp}, \mathsf{sk}_{\mathbf{r}})$
if $\tilde{\mathbf{z}}$ is equal to any output of $\mathcal{O}'_{\mathsf{Enc}}$ **then**
  return 0
end if
$s \leftarrow \mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}_{\mathbf{r}}, \tilde{\mathbf{z}})$
return $1_{s \neq \perp}$

$\mathbf{r} \neq 0$ is because for constructions [TAO16; Kim+16], there exists an algorithm $\mathsf{RandEnc}(\mathsf{pp})$ that can generate ciphertexts $\mathsf{FE.Enc}(\mathsf{msk}, \mathbf{r}')$ of a random unknown vector $\mathbf{r}'$.

I call this security property *selective unforgeability (SUF)*. An SUF secure FE is RUF secure since the adversary $\mathcal{A}_1$ can choose $\mathbf{r} \leftarrow_\$ \mathbb{F}^k$. We can also add a signature scheme to an FE to make it SUF secure. Moreover, if we use SUF security, we do not need fh-IND security in our main results in 4.4. One can reduce option-UF security of Π to SUF security of FE in a simple and intuitive way.

Theorem 19 (Revision of Theorem 7). *Let option $= \{\mathbf{c_x}, csk, \mathcal{O}_{\mathcal{B}}, \mathcal{O}_{Enroll}\}$. For any distribution family $\mathbb{B}$, if either one of the following is satisfied:*

- *FE is both fh-IND security and $\mathcal{O}'_{KeyGen}$-RUF secure (Theorem 7)*
- *FE is $\mathcal{O}'_{KeyGen}$-SUF secure*

then Π is option-UF.

Theorem 20 (Revision of Theorem 8). *Let option $= \{\mathbf{c_x}, csk, \mathcal{O}_{\mathcal{B}}, \mathcal{O}_{Probe}\}$. For any distribution family $\mathbb{B}$, if FE is $\mathcal{O}'_{Enc}$-SUF secure, then Π is option-UF secure.*

We can also leave both RUF and SUF security.

E.2 Solution II

Instead of sampling $\mathbf{r} \leftarrow_\$ \mathbb{F}^k$, one samples $\mathbf{r} \leftarrow_\$ \{0,1\}^{k+2}$.

1. Pick $k+2$ random one-hot vectors $\mathbf{e}^{(i)} \leftarrow (0, \cdots, \overset{ith}{u_i}, \cdots, 0)$, where $u \leftarrow_\$ \mathbb{F}$.
2. For $i \in [k+2]$, $\mathsf{ct}^{(i)} \leftarrow \mathcal{O}'_{\mathsf{Enc}}(\mathbf{e}^{(i)})$.
3. For $i \in [k+2]$, $d_i \leftarrow \mathsf{FE.Dec}(\mathsf{pp}, \mathsf{sk}_\mathbf{r}, \mathsf{ct}^{(i)})$. Note that d_i can only be u_i or 0.
4. Find the vector $\mathbf{r} \in \{0,1\}^{k+2}$.
5. On query $\mathcal{O}_{\mathsf{Probe}}(\mathbf{b}')$, first encode $\mathbf{b}'$ into $\mathbf{y}'$ and find $d \leftarrow \langle \mathbf{x}^*, \mathbf{y}' \rangle$. Then find a vector $\mathbf{y}''$ such that $\langle \mathbf{r}, \mathbf{y}'' \rangle = d$. Return $\mathcal{O}'_{\mathsf{Enc}}(\mathbf{y}'')$.

E.3 Solution III

We can say these fh-IPFE constructions [DDM15; TAO16; Kim+16] are not *ideal*. Their FE.Dec often aborts on unrestricted inputs. On the contrary, constructions like [Che+21] do not need discrete logarithm in FE.Dec. Unfortunately, it is not fh-IND secure.

F Trace users with different identities

The work [Sim+12] discusses a security concept related to reusability. It is called *Trace users with different identities.* It is about when a user registers multiple records of its biometrics on the database. The server or compromised database should not be able to find that two records correspond to the same person.

Based on this concept, I design the following games. $\mathcal{S}_{\mathsf{Enroll}}$ and $\mathcal{S}_{\mathsf{Probe}}$ are two simulators that do not have access to $\mathbb{B}$.

Algorithm 20 $\mathsf{REU}_{\Pi,\mathbb{B}}(\mathcal{A})$

1: $b \leftarrow_{\$} \{0, 1\}$
2: $\mathcal{B} \leftarrow_{\$} \mathbb{B}, \mathbb{B} \leftarrow \mathbb{B} \setminus \mathcal{B}$
3: $\tilde{b} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Reg}}, \mathcal{O}_{\mathsf{auth}}}(1^\lambda)$
4: **return** $1_{\tilde{b}=b}$

- $\mathcal{O}_{\mathsf{Reg}}$: It maintains a table $\mathcal{T}$ and a counter i initialized to 0 at the beginning. On query, it updates $i \leftarrow i + 1$, and behaves depending on b:
  - If $b = 0$: It generates key triplets $(\mathsf{esk}_i, \mathsf{psk}_i, \mathsf{csk}_i) \leftarrow \mathsf{Setup}(1^\lambda)$, samples an enrollment template $\mathbf{b} \leftarrow_{\$} \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}}}()$, stores an entry psk_i in $\mathcal{T}[i]$, and outputs $\mathbf{c}_{\mathbf{x}i} \leftarrow \mathsf{Enroll}(\mathsf{esk}_i, \mathbf{b})$ and csk_i.
  - If $b = 1$: It generates key triplets $(\mathsf{esk}_i, \mathsf{psk}_i, \mathsf{csk}_i) \leftarrow \mathsf{Setup}(1^\lambda)$, samples a biometric distribution $\mathcal{B}_i \leftarrow_{\$} \mathbb{B}$ and an enrollment template $\mathbf{b}_i \leftarrow_{\$} \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}_i}}()$, stores an entry $(\mathsf{psk}_i, \mathcal{B}_i)$ in $\mathcal{T}[i]$, and outputs $\mathbf{c}_{\mathbf{x}i} \leftarrow \mathsf{Enroll}(\mathsf{esk}_i, \mathbf{b}_i)$ and csk_i.
- $\mathcal{O}_{\mathsf{auth}}(i)$: This oracle has access to the table $\mathcal{T}$. On input i, it retrieves the entry $\mathcal{T}[i]$ and behaves depending on b:
  - If $b = 0$: Let $\mathsf{psk}_i \leftarrow \mathcal{T}[i]$. It samples a probe template $\mathbf{b} \leftarrow_{\$} \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}}}()$ and outputs $\mathbf{c}_{\mathbf{y}_i} \leftarrow \mathsf{Probe}(\mathsf{psk}_i, \mathbf{b})$.
  - If $b = 1$: Let $(\mathsf{psk}_i, \mathcal{B}_i) \leftarrow \mathcal{T}[i]$. It samples a probe template $\mathbf{b}_i \leftarrow_{\$} \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}_i}}()$ and outputs $\mathbf{c}_{\mathbf{y}_i} \leftarrow \mathsf{Probe}(\mathsf{psk}_i, \mathbf{b}_i)$.

If any adversary $\mathcal{A}$ cannot recover the bit b; that is, if $\Pr[\mathsf{REU}(\mathcal{A}) \to 1] = \mathsf{negl.}$, then a real-world server cannot distinguish whether a list of enrollment records all corresponding to the same person or not.

To provide more power for the adversary, we can also consider the following oracles.

- $\mathcal{O}'_{\mathsf{Reg}}(\mathsf{esk}')$: This oracle maintains a table $\mathcal{T}$ and a counter i initialized to 0 at the beginning. If esk' has been queried before, it aborts. Otherwise, it updates $i \leftarrow i + 1$ and behaves depending on b:
  - If $b = 0$: It samples an enrollment template $\mathbf{b} \leftarrow_{\$} \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}}}()$ and outputs $\mathbf{c}_{\mathbf{x}} \leftarrow \mathsf{Enroll}(\mathsf{esk}', \mathbf{b})$.

– If $b = 1$: It samples a biometric distribution $\mathcal{B}_i \leftarrow\!\!\$\ \mathbb{B}$ and an enrollment template $\mathbf{b}_i \leftarrow\!\!\$\ \mathsf{getEnroll}^{\mathcal{O}_{\mathcal{B}_i}}()$, stores $\mathcal{B}_i$ in $\mathcal{T}[i]$, and outputs $\mathbf{c}_{\mathbf{x}i} \leftarrow \mathsf{Enroll}(\mathsf{esk}', \mathbf{b}_i)$.

- $\mathcal{O}'_{\mathsf{auth}}(i, \mathsf{psk}')$: This oracle has access to the table $\mathcal{T}$ and behaves depending on b:

  – If $b = 0$: It samples a probe template $\mathbf{b} \leftarrow\!\!\$\ \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}}}()$ and outputs $\mathbf{c}_{\mathbf{y}} \leftarrow \mathsf{Probe}(\mathsf{psk}', \mathbf{b})$.

  – If $b = 1$: Let $\mathcal{B}_i \leftarrow \mathcal{T}[i]$. It samples a probe template $\mathbf{b}_i \leftarrow\!\!\$\ \mathsf{getProbe}^{\mathcal{O}_{\mathcal{B}_i}}()$ and outputs $\mathbf{c}_{\mathbf{y}i} \leftarrow \mathsf{Probe}(\mathsf{psk}', \mathbf{b}_i)$.

We forbid the adversary to query $\mathcal{O}'_{\mathsf{Reg}}$ on the same esk' twice to avoid trivial attacks. Without this restriction, the adversary can generate honest key triplet $(\mathsf{esk}, \mathsf{psk}, \mathsf{csk})$, ask for two records $\mathbf{c}_{\mathbf{x}1}, \mathbf{c}_{\mathbf{x}2}$ both corresponding to esk, and use $\mathbf{c}_{\mathbf{y}} \leftarrow \mathcal{O}'_{\mathsf{auth}}(1, \mathsf{psk})$ and $s \leftarrow \mathsf{Compare}(\mathsf{csk}, \mathbf{c}_{\mathbf{x}2}, \mathbf{c}_{\mathbf{y}})$ to know the challenge bit b. If $b = 0$, $\mathsf{Verify}(s) \to 1$ with probability TP; otherwise, $\mathsf{Verify}(s) \to 1$ with probability FP.

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.

[JP09] Ayman Jarrous and Benny Pinkas. "Secure Hamming Distance Based Computation and Its Applications". In: *Applied Cryptography and Network Security*. Ed. by Michel Abdalla et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 107–124. ISBN: 978-3-642-01957-9.

[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.

[Sim+12] Koen Simoens et al. "A framework for analyzing template security and privacy in biometric authentication systems". In: *IEEE Transactions on Information forensics and security* 7.2 (2012), pp. 833–841.

[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.