Instant Privacy-Preserving Biometric Authentication for Hamming Distance

Joohee Lee[1], Dongwoo Kim[1], Duhyeong Kim[1], Yongsoo Song[2], Junbum Shin[3], and Jung Hee Cheon[1]

[1] Seoul National University, Republic of Korea
{skfro6360, dwkim606, doodoo1204, jhcheon}@snu.ac.kr
[2] University of California, San Diego, USA
yongsoosong@ucsd.edu
[3] Software R&D Center, Samsung Electronics, South Korea
junbum.shin@samsung.com

Abstract. In recent years, there has been enormous research attention in privacy-preserving biometric authentication, which enables a user to verify him or herself to a server without disclosing raw biometric information. Since biometrics is irrevocable when exposed, it is very important to protect its privacy. In IEEE TIFS 2018, Zhou and Ren proposed a privacy-preserving user-centric biometric authentication scheme named PassBio, where the end-users encrypt their own templates, and the authentication server never sees the raw templates during the authentication phase. In their approach, it takes about 1 second to encrypt and compare 2000-bit templates based on Hamming distance on a laptop. However, this result is still far from practice because the size of templates used in commercialized products is much larger: according to NIST IREX IX report of 2018 which analyzed 46 iris recognition algorithms, size of their templates varies from 4,632-bit (579-byte) to 145,832-bit (18,229-byte).

In this paper, we propose a new privacy-preserving user-centric biometric authentication (HDM-PPBA) based on Hamming distance, which shows a big improvement in efficiency to the previous works. It is based on our new single-key function-hiding inner product encryption, which encrypts and computes the Hamming distance of 145,832-bit binary in about 0.3 seconds on Intel Core i5 2.9GHz CPU. We show that it satisfies simulation-based security under the hardness assumption of Learning with Errors (LWE) problem. The storage requirements, bandwidth and time complexity of HDM-PPBA depend linearly on the bit-length of biometrics, and it is applicable to any large templates used in NIST IREX IX report with high efficiency.

Keywords: privacy-preserving biometric authentication, inner product encryption, learning with errors.

1 introduction

Biometrics is gaining popularity in the field of authentication due to its usability and high entropy. We can see at least two use cases: one is a user authentication in a device, such as device unlock using fingerprint, face, and/or iris, which are commercialized in many smartphones. Another is to provide an access control of private keys for a public-key based authentication, such as FIDO UAF (Universal Authentication Framework) [1]. One of the most important things for biometric authentication is privacy. Biometrics is unique and irrevocable in its permanent nature so that its privacy issues are much more severe than those of passwords or tokens [2]. Obviously, storing raw biometric template in a central database or smart cards can be a risky choice, due to several threats in the literatures [3, 4]. For example, it happened that 5.6 million finger prints from U.S. government were stolen by hackers in 2015 [5], and 1 billion user's biometrics of Aadhaar were reported to be hacked [6].

To protect privacy of biometrics, there has been a great amount of research attention in privacy-preserving biometric authentication [7, 8, 9, 10, 11, 12, 13, 14, 13, 15]. We focus on one of typical authentication systems composed of two phases; enrollment and authentication phases, which proceeds as follows [16]. In enrollment phase, the service provider stores an enrolled biometric template sent from an end-user along with the end-user's ID in a database. In authentication phase, the service provider compares the stored template with a fresh template sent by an end-user, and authenticates the user if two templates are similar enough with respect to a certain measure of similarity. This approach is called *server-centric* in [15] as it heavily relies on the server's responsibility for the biometric privacy, and end-users should trust the service provider for their biometric data. Hence, if the service provider is malicious or compromised, there is no guarantee for the biometric privacy. For this reason, Zhou and Ren [15] proposed a *user-centric* biometric authentication system in which biometric templates are passed to the service provider only in encrypted forms. Their solution shows some possibilities to get over the limitations of server-centric systems, but the performance is not sufficient to be applied in practice when the size of templates are large, since their solution suffers from the quadratic to triple dependency of cost on the size of templates (it takes over 1 second to encrypt and compute Hamming distance of 2000-bit templates on an ordinary laptop without precomputations).

In fact, to build a user-centric or other secure authentication system, cryptographic tools such as secure Multi-Party Computation (MPC), Homomorphic Encryption (HE), Predicate Encryption, and Inner Product Encryption (IPE) have been employed. For example, in a recent work of Gasti et al. [13] using HE, it takes 3.29 seconds to compute Hamming distance (HD) of two 1600-bit inputs.[4] IPE with function privacy a.k.a. Function-Hiding Inner Product Encryp-

[4] Measured on "Samsung Galaxy S4 smartphone 4-Core 1.9GHz CPU (Qualcomm Snapdragon), 2GB RAM" (for client), and "Intel Xeon E5-2430L v2 6-Core 2.4GHz CPU, 64GB RAM" (for server & cloud)

tion (FH-IPE) was noticed as a simple and secure primitive for secure biometric authentication computing HD in [17] due to the close relationship of HD and inner product in binary strings. Despite its conceptional simplicity and suitability for biometric authentication, known constructions [18, 19, 17] are yet too slow to be used in the real applications. In [17], their implementation results show that it takes 1.6 seconds to compute HD between two 750-bit strings. Overall, as far as we know, there is no known practical authentication system that provides privacy of the large sized templates of hundreds or kilo bytes, in high efficiency.

In this paper, we propose a new user-centric privacy-preserving biometric authentication system with Hamming distance (HD) as a measure for similarity, i.e., a service provider authenticates a user if HD between a queried biometric template and stored template is less than a threshold. Note that this is usual in many cases, including fingercode for fingerprint [20] and all iris recognition algorithms analyzed in [21].

The most outstanding feature of our system is that it can manage large size of biometric template with high efficiency, due to the *linear* dependency of cost on the size of biometric template without sacrificing any privacy of biometric data. We showed that our system takes only about 300ms to encrypt and compute HD of 18KB binary templates on ordinary personal laptop. Since our system is simply constructed with our new primitive named FFB-IPE for secure computation of HD, it implies that usual fingercode template (whose typical template size varies from 8-byte to 640-byte according to [22]), or any iris template analyzed in NIST IRES IX report of 2018 [21] (whose template size varies from 579-byte (FotoNation 4) to 18,229-byte (Decature 5, Decature 6, TigerIT 5, TigerIT 6)) can be secured by our primitive without significantly deteriorating their performance in speed.

The novel primitive FFB-IPE proposed by this paper is a single key Function-hiding Inner Product Encryption for Binary strings, in which a ciphertext of IPE can be generated many times, while generation of secret key for a function is allowed only once. It is a weaker primitive than a general FH-IPE, but we remark that it can be used for privacy-preserving biometric authentication because the enrollment phase occurs only once, while the authentication phase held many times. The security of our primitive is based on the hardness of Learning with Errors (LWE) problem [23] on which security of various recent genuine cryptographic primitives are based. Our idea is to use a one-time pad to achieve function-hiding property when generating a secret key (for function) corresponding to a vector $\boldsymbol{x}$, and to generate an LWE instance for hiding a vector $\boldsymbol{y}$ as a ciphertext corresponding to $\boldsymbol{y}$. To allow a decryptor (or a service provider) to calculate $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$, we publish an additional value in our ciphertext, and prove that it does not lessen the hardness of LWE: More precisely, the security of our primitive is reduced to the hardness of a new variant of LWE problem, "Weak-HintLWE" which is proved by us that a polynomial-time reduction from LWE to Weak-HintLWE exists, hence to the hardness of the original LWE problem.

FFB-IPE can be easily exploited to construct biometric authentication system as follows. In enrollment phase, an end-user registers its identity with a

biometric template secured by FFB-IPE to a service provider who stores it for authentication phase. In authentication phase, end-user sends its identity with fresh biometric template encrypted by FFB-IPE to the service provider who computes HD of two templates with FFB-IPE and proceeds authentication based on it. Note that the service provider can not disclose any information of biometric templates other than the HD between them due to the security of FFB-IPE. For more details, please refer to Section 5.

We implement our FFB-IPE and authentication system based on it on Intel Core i5 CPU running at 2.9GHz processor with 8GB of memory, and the resulting performance is highly practical. With $18,229$-Byte biometrics under 128-bit security, authentication phase takes only a single round, 304 milliseconds and 125 microseconds on end-user and service provider, respectively. The communication cost is 1.178MB. Our source code written in C++ is very simple, and is disclosed on github (https://github.com/dwkim606/IPPBA).

Our contributions can be summarized as follows:

- We propose a new cryptographic primitive named FFB-IPE with standard simulation-based security under the hardness of a new variant of LWE problem "Weak-HintLWE". We then show the Weak-HintLWE problem is as hard as the original LWE problem. It can be used to secure HD-based biometric authentication system, and is highly efficient in terms of asymptotic computational cost which depends *linearly* on the size of biometric template.
- We propose HDM-PPBA built on FFB-IPE, a new practical and secure privacy preserving biometric authentication method, and implement it to show its high efficiency: a running time for HD (or equivalently, inner product) computation of 2,048-bit or 18KB binary strings are 3.12 milliseconds or 304 milliseconds, respectively on Intel Core i5 CPU at 2.9GHz with 8GB RAM. This is several orders of magnitude faster than primitives used in previous work.
- We show that our HDM-PPBA is secure against active attack which is analogous to the user-centric security model proposed in [15].

The outline of the paper is as follows. In Section 2, we present notations and some backgrounds of the LWE problem. In Section 3, we describe our authentication system and security model. In Section 4, we describe a primitive named FFB-IPE, and prove its security under the hardness assumption of LWE. In Section 5, we propose a PPBA system with FFB-IPE, and prove its security. In Section 6, we present actual parameters and implementation results of our authentication system. In Section 7, we present some other works related to ours. In Section 8, we give a summary.

2 Preliminaries

2.1 Notations

$\mathbb{R}$ and $\mathbb{Z}$ denotes the set of real numbers and integers, respectively. $\mathbb{R}^n$ is the n-dimensional vector space over $\mathbb{R}$. $\mathbb{Z}_q$ and $\mathbb{R}_q$ denote $\mathbb{Z}/q\mathbb{Z}$ and $\mathbb{R}/q\mathbb{Z}$, respectively,

with representatives in the range $(-q/2, q/2]$. We denote vectors in bold lower cases, and scalar elements in usual letters. $\langle \cdot, \cdot \rangle$ denotes the usual inner product (dot product) in $\mathbb{R}^n$. $\lfloor \cdot \rfloor$ denotes the largest integer which is not larger than the input, and $\lfloor \cdot \rceil$ denotes the nearest integer rounding upwards in case of a tie. For a (finite) set X, we denote the uniform distribution over X by $U(X)$. For a distribution D, $x \leftarrow D$ denotes sampling x following the distribution D. For the simplicity, we write $x \leftarrow U(X)$ as $x \leftarrow X$. For an integer $n \geq 1$, D^n denotes the product of *i.i.d.* random variables $D_i \sim D$.

2.2 Lattices and Gaussian distribution

A (full rank) n-dimensional *lattice* $\Lambda \subseteq \mathbb{R}^n$ is the set of all $\mathbb{Z}-$linear combinations of n linearly independent vectors (hence, $\mathbb{R}-$basis) $B = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_n\}$ of $\mathbb{R}^n$. The n-dimensional *Gaussian function* $\rho_{\sigma,\boldsymbol{c}}$ with the width $\sigma > 0$ and center $\boldsymbol{c} \in \mathbb{R}^n$ is defined as:

$$\text{for } \boldsymbol{x} \in \mathbb{R}^n, \quad \rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x}) := \exp(-\pi \|\boldsymbol{x} - \boldsymbol{c}\|^2 / \sigma^2).$$

The (continuous spherical) Gaussian distribution $D_{\boldsymbol{c},\sigma}$ is the distribution of which the probabilistic density function is proportional to $\rho_{\sigma,\boldsymbol{c}}$. When $\boldsymbol{c} = \boldsymbol{0}$, we omit $\boldsymbol{c}$ in the subscript, and add n on the superscript, *i.e.*, D^n_σ.

2.3 The Learning with Errors Problem

In these days, there are a plenty of cryptosystems based on the LWE problem introduced by Regev [23]. The LWE problem and its ring variant exploit mathematical reductions from the worst-case of the lattice problems. The problem has been offering various functionalities for the cryptosystems, exhibiting its versatility.

For a secret vector $\boldsymbol{s} \in \mathbb{Z}_q^n$ and an error distribution χ over $\mathbb{R}_q$, denote the LWE *distribution* over $\mathbb{Z}_q^n \times \mathbb{R}_q$ by $A^{\mathsf{LWE}}_{n,q,\chi}(\boldsymbol{s})$ obtained by choosing a vector $\boldsymbol{a}$ randomly from $\mathbb{Z}_q^n$ and $\boldsymbol{e}$ from χ, and outputting $(\boldsymbol{a},\, b = \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{R}_q$. For a distribution D over $\mathbb{Z}_q^n$, the decision-LWE problem is to distinguish, given arbitrary many samples, the distribution $A^{\mathsf{LWE}}_{n,q,\chi}(\boldsymbol{s})$ for a fixed $\boldsymbol{s} \leftarrow D$ from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{R}_q$ with non-negligible advantage. We denote the decision-LWE problem by $\mathsf{LWE}_{n,m,q,\chi}(D)$ where D is a distribution for the secret vector, n is the dimension of the secret vector, q is the modulus, and m is the number of samples. In this paper, we will consider multi-secret LWE problem, which is the LWE problem with secret matrix other than a vector. The multi-secret LWE *distribution* $A^{\mathsf{LWE}}_{n,q,\chi,k}(S)$ over $\mathbb{Z}_q^n \times \mathbb{R}_q^k$ is obtained by, for a secret matrix $S \in \mathbb{Z}_q^{n \times k}$, choosing a vector $\boldsymbol{a}$ randomly from $\mathbb{Z}_q^n$, and $\boldsymbol{e}$ from χ^k, and outputting $(\boldsymbol{a}, \boldsymbol{b} = S^t\boldsymbol{a} + \boldsymbol{e}) \in \mathbb{Z}_q^n \times \mathbb{R}_q^k$. For a distribution D' over $\mathbb{Z}_q^{n \times k}$, the multi-secret LWE problem is to distinguish between the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{R}_q^k$ and $A^{\mathsf{LWE}}_{n,q,\chi,k}(S)$ for a fixed $S \leftarrow D'$. As in the case of LWE, we denote the decision multi-secret LWE problem by $\mathsf{LWE}^k_{n,m,q,\chi}(D')$, where k is

the number of secret vectors. In this paper, we consider $\chi = D_\sigma$ for some $\sigma > 0$. In this case, we substitute χ by σ in the subscript of LWE. In this paper, the term "LWE assumption" means the hardness assumption of LWE.

3 System Model and Security Considerations

3.1 Hamming Distance and Inner product of Binary strings

We start with a simple note on the Hamming distance (HD) of two binary strings. We encode a k-bit binary string $\mathbf{x}$ as an k-dimensional vector $\boldsymbol{x} = (x_1, \cdots, x_k)$ whose components are ± 1 such that

$$x_i = \begin{cases} 1 & \text{if the } i\text{-th bit of } \mathbf{x} \text{ is 1,} \\ -1 & \text{otherwise.} \end{cases}$$

Let $\mathbf{x}$ and $\mathbf{y}$ be k-bit binary strings, and $\boldsymbol{x}$ and $\boldsymbol{y}$ be their encodings, respectively. Then, HD between two binary strings $\mathbf{x}$ and $\mathbf{y}$ is $\frac{1}{2}(k - \langle \boldsymbol{x}, \boldsymbol{y} \rangle)$ so that an inner product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ of $\boldsymbol{x}$ and $\boldsymbol{y}$ represents HD between binary strings.

In this paper, we assume that biometric templates are represented by binary strings, and identify them with their vector encodings as above. We also assume that the similarity between two biometric templates is measured by HD, or equivalently, the inner product of their corresponding vector encodings. It is one of the most typical cases which can be found in many biometric recognition algorithms, including FingerCode for fingerprint [20] and all iris recognition schemes analyzed in [21].

3.2 Biometric Authentication System

We consider biometric authentication system composed of two phases: enrollment and authentication whose participants are service provider and a set of end-users as in [15]. An end-user is a person with his/her own device which scans the end-user's biometric information as a binary vector and generates an encrypted template from it. The service provider can be an authentication server or an online service provider whose goal is to discriminate legitimate user according to the encrypted biometric template.

In the enrollment phase, an end-user U_i retrieves his/her biometric bit string x_i and sends it in an encrypted form, say $Enc(x_i)$ along with the end-user's identifier ID_i. Server stores $(ID_i, Enc(x_i))$ in the database. In the authentication phase, a user U_i generates his/her fresh biometric templates y_i's, and sends encrypted forms of them $Enc(y_i)'s$ with its identifier ID_i to the service provider. The service provider finds $(ID_i, Enc(x_i))$ with the same ID_i in its database and determines whether U_i is a legitimate user or not according to the similarity between x_i and y_i which can be derived from $Enc(x_i)$ and $Enc(y_i)$. An example of the whole process is shown in Fig. 1.

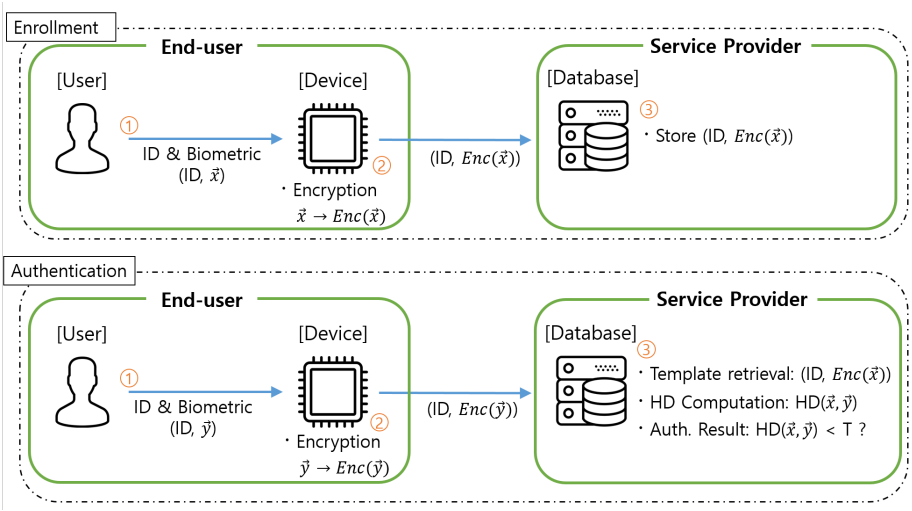

Fig. 1: Our biometric authentication system.

3.3 Adversarial Model

We follow the approach in [15] to deal with an adversarial model. We assume that the end-user is fully trusted in the enrollment phase, and the secret key of the end-user is kept secret at the local storage of the end-user's device during the whole enrollment/authentication phases. An adversary is allowed to pass her own biometric templates to the devices so that the devices act as oracles to encrypt them and send the encrypted templates to the service provider. The service provider can be malicious, that is, it may collude with an adversary and query biometric templates through the devices, watching the matching scores according to them. Eventually, there are two considered attack scenarios as follows:

- (Passive attack) The only information the service provider knows is the record $\{(Enc(x_i), ID_i)\}$, where x_i is a registered biometric of the i-th user U_i. In this case, the service provider does not know x_i's in their plaintext forms.
- (Active attack) The service provider knows $\{(Enc(x_i), ID_i)\}$ as in the previous case, but it also generates queries for the authentication phase y_i^j for $j = 1, \cdots, Q$, and sees all pairs $\{(y_i^j, Enc(y_i^j)\}$.

The first attack, which is a passive one, corresponds to the Ciphertext-Only attack, while the second attack corresponds to the Chosen-Plaintext Attack in the cryptographic context. We will give a formal definition of these attack models in Section 5.2.

4 Single Key Function-hiding Inner Product Encryption for Binary Strings from Lattice

In this section, we introduce a cryptographic primitive, single key Function-hiding Inner Product Encryption for Binary strings (hereafter, FFB-IPE), which is specialized for biometric authentication, and present a concrete construction of which security is based on the hardness assumption of LWE.

4.1 Overview

We first provide a brief sketch of our approach in this subsection: we discuss what brought us to define a new primitive FFB-IPE, and how we construct FFB-IPE based on the LWE assumption.

We adapt and relax the definition of FH-IPE to be sufficient in HD-based biometric authentication, and propose a new practical construction based on the LWE assumption. In a high-level, biometric authentication with FH-IPE includes the following procedures: (i) In the enrollment phase, the end-user generates and sends a secret key corresponding to $\boldsymbol{x}$ to the service provider, using the stored master secret key. (ii) In the authentication phase, the end-user sends a ciphertext corresponding to $\boldsymbol{y}$ to the service provider so that the service provider decrypts it and achieves $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ which directly implies HD between $\boldsymbol{x}$ and $\boldsymbol{y}$. Note that, in the HD-based biometric authentication, the secret key generation for $\boldsymbol{x}$ is done only for once (in enrollment), the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are bit strings (*i.e.,* $\boldsymbol{x}, \boldsymbol{y} \in \{-1, 1\}^k$), and the inner product value $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ has to be calculated in $\mathbb{Z}$. Since we focus on the biometric authentication with HD matcher, we set the new primitive FFB-IPE to be FH-IPE for a single key query with binary function/message spaces, where inner products are calculated in $\mathbb{Z}$. The FFB-IPE definition allows us to construct a scheme followed by a record-breaking performance.

We briefly explain how we solve the problem of constructing FFB-IPE in three steps as follows.

Step 1: Basic Scheme – Single Key Construction for Efficiency and Functionality. We first draw a basic scheme which is not function-hiding even for a single key query, and then show how to convert it to FFB-IPE. In the basic scheme (not secure), msk is a matrix T of structure $T^t = [I_k || S^t] \in \mathbb{Z}_q^{k \times (k+n)}$ where I_k is the $k \times k$ identity matrix. For a binary vector $\boldsymbol{x}$, the secret key is set to be $T\boldsymbol{x}$. Encryption of zero is a multi-secret LWE sample $(\boldsymbol{b} = -S^t\boldsymbol{a} + \boldsymbol{e}, \boldsymbol{a})$ with secret matrix S, where $\boldsymbol{a}$ is uniform randomly sampled from $\mathbb{Z}_q^n$, and $\boldsymbol{e}$ is an error vector from some distribution. Encryption of $\boldsymbol{y}$ is obtained by adding an encoding of $\boldsymbol{y}$ $(= ((q/p) \cdot \boldsymbol{y}, \boldsymbol{0}))$ to encryption of zero. In this way, it can be seen by a simple calculation that we can derive $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ from the inner product of the secret key corresponding to $\boldsymbol{x}$ and the ciphertext corresponding to $\boldsymbol{y}$. This scheme enables decryptor to calculate the inner product; however, it is obviously not function-hiding since the secret key shows $\boldsymbol{x}$ in raw.

Step 2: Full Scheme – Use One-Time Pad in Key Generation. To modify the basic scheme to be function-hiding for a single key query, we additionally

generate one more uniform random vector $\boldsymbol{u}$ together with the matrix T setting the master secret key as $\mathsf{msk} := [\boldsymbol{u}||T]$, and use $\boldsymbol{u}$ as one time pad in the key generation: we define secret key corresponding to $\boldsymbol{x}$ by $\boldsymbol{sk} := T\boldsymbol{x} + \boldsymbol{u}$, instead of $T\boldsymbol{x}$. For correctness, a ciphertext contains one additional component which is probabilistically close to the inner product of $\boldsymbol{u}$ and the original ciphertext (of the basic scheme).

Step 3: Security Proof – A Reduction from LWE to Weak-HintLWE. The additional component of a ciphertext eventually gives an additional information for the LWE secret matrix S which can be also seen as an information for the error vector (in one sample of multi-secret LWE) generated in encryption. Hence, the security of our construction is reduced to the LWE problem with an additional published value, which we call Weak-HintLWE. Informally speaking, a Weak-HintLWE sample contains a multi-secret LWE sample together with an approximate linear combination of the errors which correspond to the respective secrets of the multi-secret LWE. Needless to say, to prove the hardness of Weak-HintLWE is very crucial in our security proof. We resolve it by suggesting a polynomial-time reduction from LWE to Weak-HintLWE under the reasonable condition for the error distributions. Consequently, the security of our construction is based on the LWE assumption, and our proposed parameters in Section 6 are also set to make the original LWE problem as hard as desired.

4.2 Formalization of the Primitive

The proposed primitive FFB-IPE Π consists of four probabilistic polynomial-time algorithms Setup, KeyGen, Enc and Dec. Then FFB-IPE $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is described as below.

- $\mathsf{Setup}(1^\lambda)$: The setup algorithm outputs public parameter pp and a master secret key msk for the security parameter λ.
- $\mathsf{KeyGen}(pp, \mathsf{msk}, \boldsymbol{x} \in \{-1,1\}^k)$: The key generation algorithm on the inputs of the public parameter pp, the master secret key msk and a vector $\boldsymbol{x} \in \{-1,1\}^k$ outputs a secret key $\boldsymbol{sk}$.
- $\mathsf{Enc}(pp, \mathsf{msk}, \boldsymbol{y} \in \{-1,1\}^k)$: The encryption algorithm takes in the public parameter pp, the master secret key msk and a vector $\boldsymbol{y} \in \{-1,1\}^k$, and returns a ciphertext $\boldsymbol{c}$.
- $\mathsf{Dec}(pp, \boldsymbol{sk}, \boldsymbol{c})$: The decryption algorithm takes as the input a public parameter pp, a ciphertext $\boldsymbol{c}$, and a secret key $\boldsymbol{sk}$. It returns a decrypted value $z \in \mathbb{Z}$.

Correctness. We define the correctness of the primitive FFB-IPE $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ as follows.

Definition 1. *FFB-IPE* $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is said to be correct if for all* $(\mathsf{msk}, pp) \leftarrow \mathsf{Setup}(1^\lambda)$ *and* $\boldsymbol{x}, \boldsymbol{y} \in \{-1,1\}^k$,

$$\Pr\left[\langle \boldsymbol{x}, \boldsymbol{y}\rangle = v \;\middle|\; \begin{array}{c} \boldsymbol{sk} \leftarrow \mathsf{KeyGen}(pp, \mathsf{msk}, \boldsymbol{x}) \\ \boldsymbol{c} \leftarrow \mathsf{Enc}(pp, \mathsf{msk}, \boldsymbol{y}) \\ v \leftarrow \mathsf{Dec}(pp, \boldsymbol{sk}, \boldsymbol{c}) \end{array}\right] > 1 - 2^{-\lambda}$$

where λ is the security parameter.

Security. We define the security of FFB-IPE as follows. Our definition is similar to the simulation-based security definition in [17] with some relaxation in the sense that the oracle for KeyGen can be queried only once and beforehand.

Definition 2. *A FFB-IPE scheme $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is called 1-sSIM-secure if for all polynomial-time adversary $\mathcal{A}$, there exists a polynomial-time simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ such that the outputs of the following two experiments are computationally indistinguishable.*

Table 1: The Real-world Experiment and the Ideal-world Experiment

$Real_{\mathcal{A}}(1^\lambda)$:	$Ideal_{\mathcal{A},\mathcal{S}}(1^\lambda)$:
1. $(pp, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$	1. $(pp, \mathsf{st}) \leftarrow \mathcal{S}_1(1^\lambda)$
2. $b \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\mathsf{msk},\cdot),\mathcal{O}_{\mathsf{Enc}}(\mathsf{msk},\cdot)}(1^\lambda)$	2. $b \leftarrow \mathcal{A}^{\tilde{\mathcal{O}}_{\mathsf{KeyGen}}(\cdot),\tilde{\mathcal{O}}_{\mathsf{Enc}}(\cdot)}(1^\lambda, pp)$
3. output b	3. output b

where $\mathcal{O}_{\mathsf{KeyGen}}(\mathsf{msk},\cdot)$, $\mathcal{O}_{\mathsf{Enc}}(\mathsf{msk},\cdot)$, $\tilde{\mathcal{O}}_{\mathsf{KeyGen}}(\cdot)$, $\tilde{\mathcal{O}}_{\mathsf{Enc}}(\cdot)$ are defined as follows:

- $\mathcal{O}_{\mathsf{KeyGen}}(\mathsf{msk}, \boldsymbol{x}) = \mathsf{KeyGen}(\mathsf{msk}, \boldsymbol{x})$ *only for the first query, and aborts otherwise.*
- $\mathcal{O}_{\mathsf{Enc}}(\mathsf{msk},\cdot)$ *aborts if* $\mathcal{O}_{\mathsf{KeyGen}}(\mathsf{msk},\cdot)$ *has not been queried before. Otherwise,* $\mathcal{O}_{\mathsf{Enc}}(\mathsf{msk}, \boldsymbol{y}) = \mathsf{Enc}(\mathsf{msk}, \boldsymbol{y})$.
- $\tilde{\mathcal{O}}_{\mathsf{KeyGen}}(\cdot)$, $\tilde{\mathcal{O}}_{\mathsf{Enc}}(\cdot)$ *are stateful, and shares a simulator state* st *and a collection* $\mathcal{P} = \{\langle \boldsymbol{x}, \boldsymbol{y}^{(i)} \rangle\}_i$*, where i is a counter for* $\tilde{\mathcal{O}}_{\mathsf{Enc}}(\cdot)$ *initialized to* 0 *at the beginning, and* $\boldsymbol{x}$ *and* $\boldsymbol{y}^{(i)}$ *are the inputs for invocation of* $\tilde{\mathcal{O}}_{\mathsf{KeyGen}}(\cdot)$ *and i-th invocation of* $\tilde{\mathcal{O}}_{\mathsf{Enc}}(\cdot)$*, respectively (At the beginning,* $\mathcal{P}$ *is set to be empty).*
  - *On the adversary's invocation of* $\tilde{\mathcal{O}}_{\mathsf{KeyGen}}(\cdot)$ *with input* $\boldsymbol{x}$*,* $\tilde{\mathcal{O}}_{\mathsf{KeyGen}}(\cdot)$ *aborts unless it is the first query. Otherwise,* $\tilde{\mathcal{O}}_{\mathsf{KeyGen}}(\cdot)$ *invokes the simulator* $\mathcal{S}_2$ *on input* st*. The simulator responds with a tuple* $(\boldsymbol{sk}, \mathsf{st}') \leftarrow \mathcal{S}_2(\mathsf{st})$*. The oracle updates the state* $\mathsf{st} \leftarrow \mathsf{st}'$ *and replies to the adversary with* $\boldsymbol{sk}$*.*
  - *On the adversary's i-th invocation of* $\tilde{\mathcal{O}}_{\mathsf{Enc}}(\cdot)$ *with input* $\boldsymbol{y}^{(i)}$*, the oracle aborts unless* $\tilde{\mathcal{O}}_{\mathsf{KeyGen}}(\cdot)$ *is queried before. Otherwise, it updates the collection* $\mathcal{P} \leftarrow \mathcal{P} \bigcup \{\langle \boldsymbol{x}, \boldsymbol{y}^{(i)} \rangle\}$*, sets* $i \leftarrow i+1$*, and invokes the simulator* S_3 *on input* $\mathcal{P}$ *and* st*. The simulator responds with a tuple* $(\boldsymbol{c}, \mathsf{st}') \leftarrow \mathcal{S}_3(\mathcal{P}, \mathsf{st})$*. The oracle updates the state* $\mathsf{st} \leftarrow \mathsf{st}'$ *and replies to the adversary with* $\boldsymbol{c}$*.*

Our security definition aims to capture that all adversaries that have both $\boldsymbol{sk}$ and $\boldsymbol{c}$'s cannot obtain any information about $\boldsymbol{x}$ or $\boldsymbol{y}^{(i)}$ other than the inner products $\langle \boldsymbol{x}, \boldsymbol{y}^{(i)} \rangle$: Note that simulator in the ideal world does not take any of $\boldsymbol{x}$ or $\boldsymbol{y}$ as inputs, and it instead takes $\mathcal{P} = \{\langle \boldsymbol{x}, \boldsymbol{y}^{(i)} \rangle\}_i$ as inputs.

4.3 Our Construction

In this subsection, we propose a concrete instantiation of FFB-IPE, which satisfies the correctness and security conditions defined in the previous subsection. For a pre-determined threshold value $\mathsf{T} \in \mathbb{Z}$, our FFB-IPE $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is described as follows:

- $\mathsf{Setup}(1^\lambda)$: Choose parameters $q(\lambda)$, $p(\lambda)$, $n(\lambda)$, $k(\lambda)$, $m = n + k$, and set a distribution D_S over $\mathbb{Z}_q^{n\times k}$. Sample a random vector $\boldsymbol{u} \leftarrow \mathbb{Z}_q^m$ and a random matrix $S \leftarrow D_S$, and return the master secret key $\mathsf{msk} = (\boldsymbol{u}, S)$ and a set of public parameters $pp = (q, p, n, k, m)$. Let us denote $T := \begin{bmatrix} I_k \\ S \end{bmatrix} \in \mathbb{Z}_q^{m\times k}$ where I_k is the $k \times k$ identity matrix.
- $\mathsf{KeyGen}(pp, \mathsf{msk}, \boldsymbol{x})$: For given $\boldsymbol{x} \in \{-1, 1\}^k$, return $\boldsymbol{sk} = \boldsymbol{u} + T\boldsymbol{x} \in \mathbb{Z}_q^m$.
- $\mathsf{Enc}(pp, \mathsf{msk}, \boldsymbol{y})$: For $\boldsymbol{y} \in \{-1, 1\}^k$, do the following.
  1. Sample a random vector $\boldsymbol{a} \leftarrow \mathbb{Z}_q^n$ and an error vector $\boldsymbol{e} \leftarrow D_\sigma^k$. Let
  $$\boldsymbol{b} = -S^t\boldsymbol{a} + (q/p) \cdot \boldsymbol{y} + \boldsymbol{e} \in \mathbb{R}_q^k,$$
  and $\boldsymbol{c}_1 = (\boldsymbol{b}, \boldsymbol{a}) \in \mathbb{R}_q^k \times \mathbb{Z}_q^n$. Note that $T^t\boldsymbol{c}_1 = (q/p) \cdot \boldsymbol{y} + \boldsymbol{e}$.
  2. Sample an error $e^* \leftarrow D_{\sigma^*}$ and compute $c_0 = -\langle \boldsymbol{u}, \boldsymbol{c}_1\rangle + e^* \in \mathbb{R}_q$.
  3. Return $\boldsymbol{c} := (c_0, \boldsymbol{c}_1) \in \mathbb{R}_q \times (\mathbb{R}_q^k \times \mathbb{Z}_q^n)$.
- $\mathsf{Dec}(pp, \boldsymbol{sk}, \boldsymbol{c}, \mathsf{T})$: Parse $\boldsymbol{c} = (c_0, \boldsymbol{c}_1) \in \mathbb{R}_q \times (\mathbb{R}_q^k \times \mathbb{Z}_q^n)$. Compute and output $v = \lfloor (p/q) \cdot (c_0 + \langle \boldsymbol{sk}, \boldsymbol{c}_1\rangle)\rceil \in \mathbb{Z}_p$.

Correctness. The following theorem shows that the correctness holds for our construction with certain conditions.

Theorem 1 (correctness). *For $\boldsymbol{x}, \boldsymbol{y} \in \{-1, 1\}^k$, let $2k < p$, and msk and $\boldsymbol{c}$ are legitimately generated, i.e., $\mathsf{msk} := (\boldsymbol{u}, S) \leftarrow \mathbb{Z}_q^m \times \mathbb{Z}_q^{n\times k}$, $T := \begin{bmatrix} I_k \\ S \end{bmatrix} \in \mathbb{Z}_q^{m\times k}$, $\boldsymbol{sk} \leftarrow \boldsymbol{u} + T\boldsymbol{x}$, and $\boldsymbol{c} = (c_0, \boldsymbol{c}_1 = (\boldsymbol{b}, \boldsymbol{a}))$, where $\boldsymbol{a} \leftarrow \mathbb{Z}_q^n$, $\boldsymbol{b} = -S^t\boldsymbol{a} + (q/p) \cdot \boldsymbol{y} + \boldsymbol{e}$, and $c_0 = -\langle \boldsymbol{u}, \boldsymbol{c}_1\rangle + e^*$. Then the resulting value $v \leftarrow \lfloor (p/q) \cdot (c_0 + \langle \boldsymbol{sk}, \boldsymbol{c}_1\rangle)\rceil$ equals to $\langle \boldsymbol{x}, \boldsymbol{y}\rangle$ except with probability $2^{-\lambda}$ for the security parameter λ, as long as the following inequality holds:*

$$\Pr\left[\left|\sum_{i=1}^{k} e_i + e'\right| \geq \frac{q}{2p} \ : \ e_i \leftarrow D_\sigma, e' \leftarrow D_{\sigma^*}\right] < 2^{-\lambda}.$$

proof. Note that

$$\langle T\boldsymbol{x},\ \boldsymbol{c}_1\rangle = \langle \boldsymbol{x},\ T^t\boldsymbol{c}_1\rangle.$$

The LHS is $\langle \boldsymbol{sk} - \boldsymbol{u}\ ,\ \boldsymbol{c}_1\rangle = \langle \boldsymbol{sk}, \boldsymbol{c}_1\rangle - \langle \boldsymbol{u}, \boldsymbol{c}_1\rangle$, and the RHS is $\langle \boldsymbol{x}, (q/p) \cdot \boldsymbol{y} + \boldsymbol{e}\rangle = (q/p) \cdot \langle \boldsymbol{x}, \boldsymbol{y}\rangle + \langle \boldsymbol{x}, \boldsymbol{e}\rangle$. Hence,

$$\langle \boldsymbol{sk}, \boldsymbol{c}_1\rangle = \langle \boldsymbol{u}, \boldsymbol{c}_1\rangle + (q/p) \cdot \langle \boldsymbol{x}, \boldsymbol{y}\rangle + \langle \boldsymbol{x}, \boldsymbol{e}\rangle.$$

Therefore, we have $c_0 + \langle \boldsymbol{c}_1, sk\rangle = (q/p) \cdot \langle \boldsymbol{x}, \boldsymbol{y}\rangle + \langle \boldsymbol{x}, \boldsymbol{e}\rangle + e^*$, which implies $v = \langle \boldsymbol{x}, \boldsymbol{y}\rangle \pmod p$ if and only if $|\langle \boldsymbol{x}, \boldsymbol{e}\rangle + e^*|$ is bounded by $q/2p$. Note that $v = \langle \boldsymbol{x}, \boldsymbol{y}\rangle$ if and only if $v = \langle \boldsymbol{x}, \boldsymbol{y}\rangle \pmod p$ since $2k < p$. □

4.4 Security Proof

In this subsection, we prove the security of our construction of FFB-IPE in Section 4.3. The security of the primitive relies on the hardness of a variant of LWE, called Weak-HintLWE. We denote this problem by $\mathsf{WHintLWE}^k_{n,q,\sigma_1,\sigma_2}(D)$ in the rest of the paper for positive integers n, q, k, real numbers $\sigma_1, \sigma_2 > 0$, and the secret distribution D. The definition of $\mathsf{WHintLWE}^k_{n,q,\sigma_1,\sigma_2}(D)$ is as follow.

Definition 3 (Weak-HintLWE). *Let n, q and k be positive integers, $\sigma_1, \sigma_2 > 0$ be real numbers, $\boldsymbol{z}$ be a vector in $\{-1,1\}^k$ and S be a matrix in $\mathbb{Z}_q^{n\times k}$. The Weak-HintLWE distribution, denoted by $A^{\mathsf{WHintLWE}}_{n,q,\sigma_1,\sigma_2,k}(\boldsymbol{z}, S)$, is the distribution of $(\boldsymbol{a}, S^t\boldsymbol{a}+\boldsymbol{e}, \langle \boldsymbol{z}, \boldsymbol{e}\rangle + f) \in \mathbb{Z}_q^n \times \mathbb{R}_q^k \times \mathbb{R}_q$ where $\boldsymbol{a} \leftarrow \mathbb{Z}_q^n$, $\boldsymbol{e} \leftarrow D_{\sigma_1}^k$ and $f \leftarrow D_{\sigma_2}$. The Weak-HintLWE problem $\mathsf{WHintLWE}^k_{n,q,\sigma_1,\sigma_2}(D)$ is to distinguish, given arbitrary many independent samples for $\boldsymbol{z} \leftarrow \{-1,1\}^k$ chosen by an adversary, between $A^{\mathsf{WHintLWE}}_{n,q,\sigma_1,\sigma_2,k}(\boldsymbol{z}, S)$ for a fixed $S \leftarrow D$ and the distribution of $(\boldsymbol{a}, \boldsymbol{u}, \langle \boldsymbol{z}, \boldsymbol{e}\rangle + f)$ where $\boldsymbol{u} \leftarrow \mathbb{R}_q$.*

We discuss about the hardness of $\mathsf{WHintLWE}$ in Section 4.5. To be precise, we prove that there exists a reduction from LWE to $\mathsf{WHintLWE}$ so that $\mathsf{WHintLWE}$ is at least as hard as worst-case lattice problems such as the shortest independent vectors problem.

The following theorem is the security proof of our scheme under the hardness assumption of $\mathsf{WHintLWE}$.

Theorem 2. *Assuming that $\mathsf{WHintLWE}^k_{n,q,\sigma,\sigma^*}(D_S)$ is hard, our construction Π in Section 4.3 is a 1-sSIM-secure FFB-IPE.*

Proof. Fix an efficient adversary who makes a single query to the oracle for KeyGen and at most $Q = poly(\lambda)$ queries to the oracle for Enc. Note that an adversary has to query the oracle for KeyGen first, since otherwise the queries for Enc will be aborted. We construct a simulator $\mathcal{S}$ as follows:

- On adversary's query $\boldsymbol{x} \in \{-1,1\}^k$ to the oracle for KeyGen, the simulator receives as input a new collection $\mathcal{P}'$ of inner products and sets $\mathcal{P} \leftarrow \mathcal{P}'$. The simulator generates $\boldsymbol{sk} \leftarrow \mathbb{Z}_q^m$ and responds with it.
- On adversary's query $\boldsymbol{y}^{(i)} \in \{-1,1\}^k$ to the oracle for Enc, the simulator receives as input a new collection $\mathcal{P}'$ of inner products and updates $\mathcal{P} \leftarrow \mathcal{P}'$ (retrieving $\langle \boldsymbol{x}, \boldsymbol{y}^{(i)}\rangle$). The simulator samples $\boldsymbol{b}^{(i)} \leftarrow \mathbb{R}_q^k$, $\boldsymbol{a}^{(i)} \leftarrow \mathbb{Z}_q^n$, and $\boldsymbol{c}_1^{(i)} \leftarrow (\boldsymbol{b}^{(i)}, \boldsymbol{a}^{(i)})$. It also samples $e_j^{(i)} \leftarrow D_\sigma$ for $1 \le j \le k$ and $e^{*(i)} \leftarrow D_{\sigma^*}$, and then sets $c_0^{(i)} \leftarrow -\langle \boldsymbol{sk}, \boldsymbol{c}_1^{(i)}\rangle + (q/p)\cdot\langle \boldsymbol{x}, \boldsymbol{y}^{(i)}\rangle + \sum_{j=1}^k e_j^{(i)} + e^{*(i)}$ and $\boldsymbol{c}^{(i)} \leftarrow (c_0^{(i)}, \boldsymbol{c}_1^{(i)})$. The simulator responds with $\boldsymbol{c}$.

Let Expt 0 be the real world experiment. That is, for an efficient adversary $\mathcal{A}$, we generate $\mathsf{msk} := (\boldsymbol{u}, S)$ from Setup to answer the oracle queries with the legitimate KeyGen and Enc outputs consistently in Expt 0. We show that Expt 0, real world experiment, is indistinguishable from the simulated one which is

numbered by Expt 3, using a hybrid argument. We define Expt 1 and Expt 2 as follows.

Expt 1. substitutes $\boldsymbol{sk} = \boldsymbol{u} + T\boldsymbol{x}$ in Expt 0 with $\boldsymbol{sk} \leftarrow \mathbb{Z}_q^m$. Generates $\boldsymbol{c}_1^{(i)}$ by $\boldsymbol{c}_1^{(i)} := (\boldsymbol{b}^{(i)} = -S^t\boldsymbol{a}^{(i)} + (q/p)\cdot\boldsymbol{y}^{(i)} + \boldsymbol{e}^{(i)}, \boldsymbol{a}^{(i)})$ where $\boldsymbol{a}^{(i)} \leftarrow \mathbb{Z}_q^n$, $\boldsymbol{e}^{(i)} \leftarrow D_\sigma^k$ (as in the Enc algorithm). Replaces $c_0^{(i)}$ with $c_0^{(i)} \leftarrow -\langle \boldsymbol{sk}, \boldsymbol{c}_1^{(i)}\rangle + (q/p)\cdot\langle \boldsymbol{x}, \boldsymbol{y}^{(i)}\rangle + \langle \boldsymbol{x}, \boldsymbol{e}^{(i)}\rangle + e^{*(i)}$, where $e^{*(i)} \leftarrow D_{\sigma^*}$.

Observe that $\boldsymbol{u}$ involved in generating $\boldsymbol{sk}$ is uniformly random and is used for only one time. Hence, the distributions of $(\boldsymbol{sk}, \{\boldsymbol{c}^{(i)}\}_i)$ in Expt 0 and Expt 1 are the same in the adversary's view.

Expt 2. substitutes $\boldsymbol{c}_1^{(i)}$ in Expt 1 with uniformly chosen $\boldsymbol{c}_1^{(i)} = (\boldsymbol{b}^{(i)}, \boldsymbol{a}^{(i)}) \leftarrow \mathbb{R}_q^k \times \mathbb{Z}_q^n$, and sets $c_0^{(i)} \leftarrow -\langle \boldsymbol{sk}, \boldsymbol{c}_1^{(i)}\rangle + (q/p)\cdot\langle \boldsymbol{x}, \boldsymbol{y}^{(i)}\rangle + \langle \boldsymbol{x}, \boldsymbol{e}^{(i)}\rangle + e^{*(i)}$, where $\boldsymbol{e} \leftarrow D_\sigma^k$, $e^{*(i)} \leftarrow D_{\sigma^*}$.

The distributions of $(\boldsymbol{sk}, \{\boldsymbol{c}^{(i)}\}_i)$ in Expt 1 and Expt 2 are computationally indistinguishable when assuming the hardness of the $\mathsf{WHintLWE}^k_{n,q,\sigma,\sigma^*}(D_S)$ problem. The distributions of $(\boldsymbol{sk}, \{\boldsymbol{c}^{(i)}\}_i)$ in Expt 2 and Expt 3 are identical, since $\boldsymbol{e}^{(i)}$ in Expt 2 is independent from other variables.

4.5 The Hardness of HintLWE and Weak-HintLWE

We first define another variant of (multi-secret) LWE, the HintLWE problem, which additionally publishes a *full-dimensional hint* on the error of original LWE. Contrary to the Weak-HintLWE problem, instead of publishing $\langle \boldsymbol{z}, \boldsymbol{e}\rangle + f$ for some known vector $\boldsymbol{z}$ and a Gaussian error f, the HintLWE problem gives the whole vector $\boldsymbol{e} + \boldsymbol{f}$ for a Gaussian vector $\boldsymbol{f}$ which contains much more information. We will first reduce LWE to HintLWE, and then reduce HintLWE to Weak-HintLWE which is relatively a simple step.

Definition 4. *Let n, q and k be positive integers, $\sigma_1, \sigma_2 > 0$ be real numbers, and S be a matrix in $\mathbb{Z}_q^{n\times k}$. The HintLWE distribution, denoted by $A^{\mathsf{HintLWE}}_{n,q,\sigma_1,\sigma_2,k}(S)$, is the distribution of $(\boldsymbol{a}, S^t\boldsymbol{a}+\boldsymbol{e}, \boldsymbol{e}+\boldsymbol{f}) \in \mathbb{Z}_q^n \times \mathbb{R}_q^k \times \mathbb{R}_q^k$ where $\boldsymbol{a} \leftarrow \mathbb{Z}_q^n$, $\boldsymbol{e} \leftarrow D^k_{\sigma_1}$ and $\boldsymbol{f} \leftarrow D^k_{\sigma_2}$.*

Definition 5 (HintLWE). *For positive integers n, q, k, real numbers $\sigma_1, \sigma_2 > 0, \sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$ and a distribution D over $\mathbb{Z}_q^{n\times k}$, the HintLWE problem $\mathsf{HintLWE}^k_{n,q,\sigma_1,\sigma_2}(D)$ is to distinguish, given arbitrary many independent samples, between $U(\mathbb{Z}_q^n) \times U(\mathbb{R}_q^k) \times D_\sigma^k$ and $A^{\mathsf{HintLWE}}_{n,q,\sigma_1,\sigma_2,k}(S)$ where $S \leftarrow D$.*

When $k = 1$ in Definition 5, we omit the superscript k and substitute the capital letter S by the small letter $\boldsymbol{s}$.

Now, we present a theorem about a polynomial-time reduction from LWE to $\mathsf{HintLWE}$. Before introducing the theorem, we describe a crucial lemma on a conditional Gaussian distribution which will be exploited in the hardness proof. The proof of the lemma is given in Appendix A.

Lemma 1. *For real numbers $\sigma_1, \sigma_2 > 0$, let e and f be variables of distributions D_{σ_1} and D_{σ_2}, respectively. Let $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$, then $(e+f, e|(e+f))$ follows the distribution $(D_\sigma, D_{L\sigma_1^2/\sigma^2,\ \sigma_1\sigma_2/\sigma})$ where L denotes the value of $e+f$.*

Proof. In Appendix A.

Theorem 3. *Let n, q, k be positive integers, $\sigma_1, \sigma_1', \sigma_2'$ be positive real numbers satisfying $\sigma_1 = \sigma_1'\sigma_2'/\sqrt{\sigma_1'^2 + \sigma_2'^2}$, and D be a distribution over $\mathbb{Z}_q^{n\times k}$. Then there exists a polynomial-time reduction from $\mathsf{LWE}^k_{n,q,\sigma_1}(D)$ to $\mathsf{HintLWE}^k_{n,q,\sigma_1',\sigma_2'}(D)$ which preserves the advantage.*

Proof. We first prove the case $k = 1$. For a given $\mathsf{LWE}_{n,q,\sigma_1}(D)$ sample $(\boldsymbol{a}, b)$, we transform the sample to $(\boldsymbol{a}, b+f, \kappa f)$ where $f \leftarrow D_{\sigma_2}$ for $\sigma_2 = \sigma_1\sigma_1'/\sigma_2'$ and $\kappa = (\sigma_1'^2+\sigma_2'^2)/\sigma_1'^2$. We now claim that $(\boldsymbol{a}, b+f, \kappa f)$ is exactly the $\mathsf{HintLWE}_{n,q,\sigma_1',\sigma_2'}(D)$ sample we want.

First we think of the case $b = \langle \boldsymbol{a}, \boldsymbol{s}\rangle + e$ where $e \leftarrow D_{\sigma_1}$ for a fixed $\boldsymbol{s} \leftarrow D$. Then we can check

$$(\boldsymbol{a}, b+f, \kappa f) = (\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s}\rangle + e + f, \kappa f).$$

Note that it is enough to show that the distribution of $(\kappa f, (e+f)|f)$ equals to the distribution of $(e'+f', e'|(e'+f'))$ where $e' \leftarrow D_{\sigma_1'}$ and $f' \leftarrow D_{\sigma_2'}$, which implies that $(\boldsymbol{a}, b+f, \kappa f)$ is distributed exactly as same as $(\boldsymbol{a}, b+e', e'+f')$.

It is easy to check that the distribution of $(\kappa f, (e+f)|f)$ is $\mathcal{D}_1 = (D_{\kappa\sigma_2}, D_{L/\kappa,\sigma_1})$ where L denotes the value of κf. Let $\sigma' = \sqrt{\sigma_1'^2 + \sigma_2'^2}$, then by Lemma 1, $(e'+f', e'|(e'+f'))$ follows the distribution $\mathcal{D}_2 = (D_{\sigma'}, D_{L'\sigma_1'^2/\sigma'^2,\ \sigma_1'\sigma_2'/\sigma'})$ where L' denotes the value of $e'+f'$. By the condition $\sigma_1 = \sigma_1'\sigma_2'/\sqrt{\sigma_1'^2+\sigma_2'^2}$ and the definitions of σ_2, κ and σ', we obtain $\kappa\sigma_2 = \sigma'$, $1/\kappa = \sigma_1'^2/\sigma'^2$ and $\sigma_1 = \sigma_1'\sigma_2'/\sigma'$ so that distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ are identical.

In case that b is uniform over $\mathbb{R}_q$, the distribution of $(\boldsymbol{a}, b+f, \kappa f)$ equals to the distribution of $(\boldsymbol{a}, b, kf)$, which is exactly $U(\mathbb{Z}_q^n)\times U(\mathbb{R}_q)\times D_\sigma$ for $\sigma = \sqrt{\sigma_1^2+\sigma_2^2}$. Therefore, the reduction is correct for the case $k = 1$, and obviously preserves the advantage.

For the case of arbitrary k, the reduction process is exactly the transformation from a multi-secret LWE sample $(\boldsymbol{a}, \boldsymbol{b})$ to $(\boldsymbol{a}, \boldsymbol{b}+\boldsymbol{f}, \kappa\boldsymbol{f})$ where $\boldsymbol{f} \leftarrow D_{\sigma_2}^k$. For $1 \le i \le k$, we have already proved that $(\kappa f_i, (e_i+f_i)|f_i)$ and $(e_i'+f_i', e_i'|(e_i'+f_i'))$ are equally distributed where $e_i' \leftarrow D_{\sigma_1'}$ and $f_i' \leftarrow D_{\sigma_2'}$, which are all mutually independent cases (for i). Therefore, in case of $\boldsymbol{b} = S^t\boldsymbol{a} + \boldsymbol{e}'$, the distribution of $(\boldsymbol{a}, \boldsymbol{b}+\boldsymbol{f}, \kappa\boldsymbol{f})$ equals to the distribution of $(\boldsymbol{a}, S^t\boldsymbol{a}+\boldsymbol{e}', \boldsymbol{e}'+\boldsymbol{f}')$. In case that $\boldsymbol{b}$ is uniform over $\mathbb{R}_q^k$, it is just the analogue of the case $k = 1$.

One simple case of Theorem 3 is the case of $\sigma_1' = \sigma_2'$. Then, we can check that $\sigma_1 = \sigma_2 = \sigma_1'/\sqrt{2}$ and $\kappa = 2$.

Theorem 4. *Let n, q, k be positive integers, $\sigma_1, \sigma_2, \sigma_2'$ be positive real numbers satisfying $\sigma_2' = \sqrt{k}\sigma_2$, and D be a distribution of a secret matrix over $\mathbb{Z}_q^{n\times k}$. Then there exists a polynomial-time reduction from $\mathsf{HintLWE}^k_{n,q,\sigma_1,\sigma_2}(D)$ to $\mathsf{WHintLWE}^k_{n,q,\sigma_1,\sigma_2'}(D)$ which preserves the advantage.*

Proof. Assume that we are asked to provide samples for a vector $\boldsymbol{z} \in \{-1, 1\}^k$, and get a $\mathsf{HintLWE}^k_{n,q,\sigma_1,\sigma_2}(D)$ sample $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{e}+\boldsymbol{f})$. Then, we transform $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{e}+\boldsymbol{f})$ to $(\boldsymbol{a}, \boldsymbol{b}, \langle \boldsymbol{z}, \boldsymbol{e}+\boldsymbol{f}\rangle) \in \mathbb{Z}_q^n \times \mathbb{R}_q^k \times \mathbb{R}_q$. Since each component of $\boldsymbol{z}$ is 1 or -1, the distribution of $\langle \boldsymbol{z}, \boldsymbol{f}\rangle$ is identical to the Gaussian distribution $D_{\sqrt{k}\sigma_2}$. Also, since $\boldsymbol{e}$ and $\boldsymbol{f}$ are independent, the distribution of $\langle \boldsymbol{z}, \boldsymbol{e}+\boldsymbol{f}\rangle$ is actually the distribution of $\langle \boldsymbol{z}, \boldsymbol{e}\rangle + f'$ where $f' \leftarrow D_{\sqrt{k}\sigma_2}$. From this point, the output of the reduction is exactly the $\mathsf{WHintLWE}^k_{n,q,\sigma_1,\sigma_2'}(D)$ sample.

We fixed the domain of the vector $\boldsymbol{z}$ to $\{-1, 1\}^k$, but actually our reduction does not depend on the domain.

From Theorem 3 and Theorem 4, we finally obtain the hardness of the $\mathsf{WHintLWE}$ problem under the hardness assumption of the LWE problem.

Corollary 1. *Let n, q, k be positive integers, $\sigma_1, \sigma_1', \sigma_2'$ be positive real numbers satisfying $\sigma_1 = \sigma_1'\sigma_2'/\sqrt{\sigma_1'^2 + \sigma_2'^2}$, and D be a distribution over $\mathbb{Z}_q^{n\times k}$. Then there exists a polynomial-time reduction from $\mathsf{LWE}^k_{n,q,\sigma_1}(D)$ to $\mathsf{WHintLWE}^k_{n,q,\sigma_1',\sqrt{k}\sigma_2'}(D)$ which preserves the advantage.*

5 HDM-PPBA

In this section, we formally introduce a new user-centric privacy-preserving biometric authentication for HD matcher called HDM-PPBA derived from FFB-IPE in Section 4.3, and analyze its security.

5.1 HDM-PPBA protocol

Our protocol consists of two phases: enrollment, and authentication. The protocol is based on the proposed FFB-IPE $\Pi = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ constructed in Section 4 for a secure computation of HD.

5.2 Security Analysis

We formally define the attack models in Section 3.3 following the approach in [15] and prove that our HDM-PPBA equipped with 1-sSIM-secure FFB-IPE is secure against both attacks.

Security Against Passive Attack According to the attack experiment depicted in Figure 2, we define the security of FFB-IPE against passive attack as follows.

Definition 6. *The FFB-IPE is secure against passive attack if, for all polynomial-time adversary $\mathcal{A}$, there exists a cryptographically negligible function negl such that*

$$|Pr(\mathsf{Passive}_{\mathcal{A}}^{FFB\text{-}IPE}(\lambda) = 1) - \frac{1}{2}| < \mathsf{negl}(\lambda).$$

Protocol 1 Our HDM-PPBA system

Input: $\boldsymbol{x}, \boldsymbol{y} \in \{-1, 1\}^k$, $T \in \mathbb{R}$
Output: $res \in \{authenticate, reject\}$
Registration: An end-user U registers with his/her identity ID and a biometric template $\boldsymbol{x}$ to the service provider S.

1: An end-user U sets the parameters of FFB-IPE according to the security parameter λ.
2: U generates a master secret key $(\mathsf{msk}, pp) \leftarrow \mathsf{Setup}(1^\lambda)$, and stores it.
3: U generates a key $\boldsymbol{sk} \leftarrow \mathsf{KeyGen}(pp, \mathsf{msk}, \boldsymbol{x})$ with the end-user's biometrics $(\boldsymbol{x})$, and sends $(ID_U, \boldsymbol{sk}, pp)$ to the service provider S.
4: S stores $(ID_U, \boldsymbol{sk}, pp)$.

Authentication: An end-user U retrieves a fresh biometric template, and sends a ciphertext of it to the service provider S for an authentication.

1: U computes a ciphertext $\boldsymbol{c} \leftarrow \mathsf{Enc}(pp, \mathsf{msk}, \boldsymbol{y})$, and sends it along with user's ID ID_U to S.
2: S retrieves the stored values $(ID_U, \boldsymbol{sk}, pp)$ at the enrollment phase, computes an inner-product value $z \leftarrow \mathsf{Dec}(pp, \boldsymbol{sk}, \boldsymbol{c})$, and gets the hamming distance $d = (k–z)/2$ between the biometrics $\boldsymbol{x}$ and $\boldsymbol{y}$.
3: S output $authenticate$ if a distance d is less than a given threshold T $(d < T)$, and $reject$ otherwise $(d \geq T)$.

Theorem 5. *The proposed HDM-PPBA scheme in Section 4.3 is secure against passive attack.*

We skip the proof of Theorem 5 since the security against passive attack is implied by the security against active attack.

Security Against Active Attack

Definition 7. *The FFB-IPE is secure against active attack if, for all polynomial-time adversary $\mathcal{A}$, there exists a cryptographically negligible function* negl *such that*

$$\left| Pr(\mathsf{Active}_{\mathcal{A}}^{FFB\text{-}IPE}(\lambda) = 1) - \frac{1}{2} \right| < \mathsf{negl}(\lambda).$$

Theorem 6. *The proposed HDM-PPBA scheme in Section 4.3 is secure against active attack.*

Proof. The proposed FFB-IPE is proven to achieve the 1-sSIM-security defined in Definition 2. The security of HDM-PPBA against active attack is an indistinguishability-based version of the simulation-based security in Definition 2, so is straightforward.

Experiment 2 Passive Attack Experiment $\mathsf{Passive}_{\mathcal{A}}^{\text{FFB-IPE}}(\lambda)$:

1: Given a security parameter λ, the challenger $\mathcal{C}$ outputs pp.
2: Given λ and pp, the adversary $\mathcal{A}$ generates and outputs $\boldsymbol{x}_0$, $\boldsymbol{x}_1$, and two sequences of messages in $\{-1,1\}^k$, say $Y_0 = (\boldsymbol{y}_0^{(1)}, \boldsymbol{y}_0^{(2)}, \cdots, \boldsymbol{y}_0^{(Q)})$ and $Y_1 = (\boldsymbol{y}_1^{(1)}, \boldsymbol{y}_1^{(2)}, \cdots, \boldsymbol{y}_1^{(Q)})$ such that $\langle \boldsymbol{x}_0, \boldsymbol{y}_0^{(i)} \rangle = \langle \boldsymbol{x}_1, \boldsymbol{y}_1^{(i)} \rangle$ for all i's.
3: Given λ, $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda)$ to obtain msk.
4: $\mathcal{C}$ chooses a uniform random bit $b \in \{0,1\}$ and computes $\boldsymbol{sk} \leftarrow \mathsf{KeyGen}(pp, \mathsf{msk}, \boldsymbol{x}_b)$ and $\boldsymbol{c}_i = \mathsf{Enc}(\mathsf{msk}, pp, \boldsymbol{y}_b^{(i)})$ for all i's. $\mathcal{C}$ sends the sequence $C = (\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_Q)$ and $\boldsymbol{sk}$ to $\mathcal{A}$.
5: $\mathcal{A}$ outputs $b' \in \{0,1\}$.
6: The output of the experiment is 1 if $b = b'$, and 0 otherwise.

Experiment 3 Active Attack Experiment $\mathsf{Active}_{\mathcal{A}}^{\text{FFB-IPE}}(\lambda)$:

1: Given a security parameter λ, the challenger $\mathcal{C}$ runs $\mathsf{Setup}(1^\lambda)$ to obtain (msk, pp), and sends pp to the adversary $\mathcal{A}$.
2: $\mathcal{A}$ generates and sends $\boldsymbol{x}_0, \boldsymbol{x}_1 \in \{-1,1\}^k$ to $\mathcal{C}$.
3: $\mathcal{C}$ chooses a uniform random bit $b \in \{0,1\}$, calculates $\boldsymbol{sk} \leftarrow \mathsf{KeyGen}(pp, \mathsf{msk}, \boldsymbol{x}_b)$, and sends $\boldsymbol{sk}$ to $\mathcal{A}$.
4: **for** $i = 1 \rightarrow Q$ **do**
5: $\quad$ $\mathcal{A}$ is queries with $\boldsymbol{y}_0^{(i)}$ and $\boldsymbol{y}_1^{(i)}$ such that $\langle \boldsymbol{x}_0, \boldsymbol{y}_0^{(i)} \rangle = \langle \boldsymbol{x}_1, \boldsymbol{y}_1^{(i)} \rangle$.
6: $\quad$ $\mathcal{C}$ computes and sends $\boldsymbol{c} \leftarrow \mathsf{Enc}(pp, \mathsf{msk}, \boldsymbol{y}_b^{(i)})$.
7: **end for**
8: $\mathcal{A}$ outputs $b' \in \{0,1\}$.
9: The output of the experiment is 1 if $b = b'$, and 0 otherwise.

6 Performance Evaluation

In this section, we only evaluate the performance of FFB-IPE because (1) a lot of works have been done for the performance evaluation of TLS, for example, the one showing a fast implementation of TLS reports that the running time of a TLS handshake is less than 10 milliseconds on Intel Core 2 Duo E8400 at 3.0 GHz with 4GB RAM [24], and (2) TLS is widely used to protect network communications and https, a secure http protocol protected by TLS, will now be the default for all Android Apps as announced by Google [25], so there will be a high possibility that TLS is applied to services before adopting our HDM-PPBA.

6.1 Experimental Setup

Our primitive was implemented in C++ 11 standard, and performed on Intel Core i5 CPU running MacOS (64 bit) at 2.9GHz processor with 8GB of memory. We used g++ compiler of Apple LLVM version 9.1.0.

As an optimization for the implementation, we used power-of-2 moduli $p < q \leq 2^{64}$. To be precise, we store elements of $\mathbb{Z}_q$ and $\mathbb{R}_q$ in `uint64_t` type while

scaling them up for $(64 - \log q)$ bits. Then, the modulo q operation is automatically done without any overhead, which makes the implementation very simple and fast overall. Note that the elements in $\mathbb{R}_q$ are rounded off to the $(64 - \log q)$-th bit after the radix point. Furthermore, the rounding operation $\lfloor (p/q) \cdot x \rceil$ for $x \in \mathbb{Z}_q$ included in the Dec algorithm is efficiently done by adding a constant and bit-wise shifting. Since $\lfloor (p/q) \cdot x \rceil = \lfloor (p/q) \cdot (x + q/2p) \rfloor$, it is done by right-shifting for $(\log q - \log p)$ bits after adding a constant $q/2p$. The source code can be found in github (https://github.com/dwkim606/IPPBA).

6.2 Parameters Setting with LWE Estimator

We present several parameter sets in case that $D_S = U(\{0,1\}^{n \times k})$, that is, each component of the secret matrix is binary. In Table 2, parameter sets I and II correspond to the case that $k = 2048$ and $k = 145832$ (for about 18KB biometric templates), respectively.

By Corollary 1, there exists a reduction from $\mathsf{LWE}^k_{n,q,\sigma_1}(D_S)$ to $\mathsf{WHintLWE}^k_{n,q,\sigma,\sigma^*}(D_S)$, where $\sigma_1 = \sigma \cdot \sigma^* / \sqrt{k\sigma^2 + (\sigma^*)^2}$. For the simplicity, we set $\sigma = \sqrt{2}\sigma_1 = \sigma^*/\sqrt{k}$. Rather than considering direct attacks on $\mathsf{WHintLWE}^k_{n,q,\sigma,\sigma^*}(D_S)$, we select the parameters following the reduction from LWE. That is, we set the parameters which make $\mathsf{LWE}^k_{n,q,\sigma_1}(D_S)$ to be secure against the best attacks, which is more conservative approach in parameter selection.

Note that k denotes the number of secret vectors in the LWE problem, and it corresponds to the bit length of a biometric in our scheme. A positive integer n denotes the dimension of secret in the LWE problem. To analyze concrete hardness of LWE problem for certain parameters, we used Albrecht's LWE estimator [26][5]. It estimates the bit security of certain LWE problems considering known attacks (dual attack [27], primal attack [28], etc.) on the LWE problem. We also set our parameters to achieve the correctness of our construction described in Theorem 1, *i.e.*, the Dec algorithm in Section 4.3 computes the exact inner product $v = \langle \boldsymbol{x}, \boldsymbol{y} \rangle$ except for a negligible probability in the security parameter λ.

Table 2: Proposed Parameters for our construction satisfying 128-bit security on $\mathsf{LWE}^k_{n,q,\sigma/\sqrt{2}}(U(\{0,1\}^{n \times k}))$

λ	Name	k	n	q	p	σ	σ^*
128	I	2048	928	2^{32}	2^{20}	2.39	108
	II	145832	1368	2^{64}	2^{32}	2.96×10^5	1.12×10^8

Table 3: Implementation results of our FFB-IPE

Parameter Set	Biometric (bits)	Master Secret Key	Secret Key	Ciphertext	Running Time		
		msk (MB)	$\boldsymbol{sk}$ (KB)	$\boldsymbol{c}$ (KB)	KeyGen (ms)	Enc (ms)	Dec (ms)
I	2048	0.24	23.81	23.82	15 + 0.33	3.12	0.0021
II	145832	26.11	1177.60	1177.61	1535 + 115	304	0.125

6.3 Implementation Results

Here follows the implementation results of our FFB-IPE for each of the parameter sets in Table 3. The size of a master secret key msk, a secret key $\boldsymbol{sk}$, a ciphertext $\boldsymbol{c}$, and the running time of KeyGen, Enc, and Dec algorithms are presented. Note that the running times for KeyGen (resp. Enc and Dec) was averaged over 100 times of measurements.

Adapting this result into our PPBA protocol, the master secret key is stored by end-user, and a secret key (resp. a ciphertext) is sent from the end-user to the service provider as an enrollment message (resp. a query message). As described in Section 5, KeyGen, Enc, and Dec algorithms are included in the enrollment phase, the query phase, and the authentication phase, respectively.

The size ratio of $\boldsymbol{sk}$ to $\boldsymbol{x}$, and that of $\boldsymbol{c}$ to $\boldsymbol{y}$ is no larger than 100 which is somewhat reasonable to be used in practice. The most notable point of our implementation result is running time of algorithms: for a 2048-bit message (parameter set I), it takes only several milliseconds for both key generation and encryption, and the decryption takes only 2 microseconds. The running-time result for the indicates that our primitive is highly efficient, and is capable of handling large templates of NIST IREX IX report considering the result for the parameter set II.

6.4 Comparison & Complexity analysis

In this subsection, we compare our performance with known PPBAs secure in malicious model. More precisely, there are some PPBAs secure in malicious model in which the server stores encrypted biometric templates, and we arrange recent efficient methods regarding authentication using HD of binary biometric to compare with ours.

Karabat et al. [29] proposed a PPBA named THRIVE exploiting Goldwasser-Micali's threshold (XOR-) homomorphic encryption [30]. The XOR homomorphic encryption enables computing HD of biometrics without revealing their exact value, and decryption key for the threshold encryption is distributed to client and server so that the server can not disclose any information from the

[5] https://bitbucket.org/malb/lwe-estimator

Table 4: Comparison of various schemes with malicious server model for authentication phase

Protocol	Primitive	Biometric (bits)	Communication Cost (KB)	Client (ms)	Server (ms)
THRIVE [29][a]	Threshold HE	2048	787	2051	6146
Gasti et al. [13][b]	(Outsourced) MPC	1600	490	1130 + (1150)[c]	1010
PassBio [15][d,⋆]	TPE	2000	500⋆⋆	600 + (600)[c]	0.3⋆⋆
Kim et al. [17][e,⋆]	FH-IPE	750	96	556	1600
This Work (I, II)[f,⋆]	**FFB-IPE**	**2048**	**24**	**3.12**	**0.0021**
		145832	**1177.61**	**304**	**0.125**

[a] Benchmarked on "Intel Core 3.2 GHz processor".
[b] Benchmarked on
- Client: "Samsung Galaxy S4 smartphone 4-Core 1.9GHz processor (Qualcomm Snapdragon), 2 GB RAM".
- Server & Cloud: "Intel Xeon E5-2430L v2 6-Core 2.4GHz processor, 64 GB RAM".
[c] Offline precomputation.
[d] Benchmarked on "Intel Core i5 1.60 GHz processor, 4 GB RAM".
[e] Benchmarked on "Intel Core i7 4.00 GHz processor, 16 GB RAM".
[f] Benchmarked on "Intel Core i5 2.90 GHz processor, 8 GB RAM".
⋆ The performance is measured for the primitive without using TLS: Communication Cost is the size of a ciphertext, and Client and Server denotes the running times of encryption and decryption, respectively.
⋆⋆ We estimated the expected values from the asymptotic values proposed in [15].

encrypted biometric. However, since biometric is encrypted bitwisely, and signature is necessary to prevent malicious behavior of each participants, its efficiency is quite doubtful when the size of biometric is large.

Recently, Gasti et al. [13] proposed an outsourced PPBA improving GC and OT technique in MPC. In their system, computational burden of client (modeled as a smart phone) is mitigated by an (untrusted) cloud with more computational power. It is secure against malicious participants, and is more efficient than other state-of-the-art general purpose outsourced MPC [31, 32, 33] offering security under the presence of malicious participants. The security of [13] is highly satisfiable since it is secure even if the server is malicious or client hire malicious cloud or they collude.

Recently, Zhou and Ren [15] proposed a privacy preserving biometric authentication with a new primitive called Threshold Predicate Encryption (TPE) which can be applied to authentication based on Euclidean metric or Hamming distance. It has a remarkable feature that the server can only see the result of comparison if the distance between two template is bigger than the threshold or not. The primitive is based on simple matrix randomization, and shows a

simple implementation only composed of matrix multiplication and random permutation. However, it suffers inefficient asymptotic complexity on Client, Server, Communication such as $O(k^3), O(k^2), O(k^2 l)$ where k is the dimension of a biometric vector, l is the size of each component, and is only feasible for k near 2000 or a little more.

We can compare our primitive with Kim et al. [17]'s FH-IPE. It offers a security of biometric by the security of FH-IPE under the presence of malicious server, and is practical for small-size biometric. Also, it does not have any key management problem contrary to SHE, since the function key directly reveals the result of computation. However, it accompanies inefficient decryption process disclosing computation result by calculating discrete logarithm over a group, and will be quite hard to be applied in systems regarding biometrics of larger bit size such as 18KB. It shows the impracticality of the pairing (bilinear map)-based FH-IPE constructed so far.

On the other hand, our authentication system provides the same security of biometric as usual FH-IPE, and is highly efficient since it is algorithmically simple, and is based on simple LWE problem which shows outstanding scalability. More precisely, we can easily see that the asymptotic complexities for operations of Client, Server, Communication are $O(k), O(k), O(kl)$, where k is the bit length of a binary biometric, l is the size of $\mathbb{R}_q$ or $\mathbb{Z}_q$. Note that n of FFB-IPE is the dimension of (secret vector of) LWE problem, which depends only on the security and correctness, is much smaller than k as k increases, and omitted as a constant. It grants our FFB-IPE an outstanding efficiency and scalability for large biometric as seen by Table 4 where we summarized the performance of other schemes and ours. Note that other schemes satisfy 80-bit security, ours satisfies 128-bit security for both parameter sets I and II.

7 Related Works

There are many researches related to Privacy Preserving Biometric Authentication. Since Jarrous and Pinkas [7] first used MPC to achieve HD-based PPBA, there have been similar approaches [8, 10] which improve the performance achieving efficient MPC for HD in Honest-but-Curious (HBC) model,[6] where adversary follows the protocol honestly but attempts to deduce additional information from it.[7] However, as Simoens et al. [9] pointed out, PPBA secure in HBC model is not sufficient since malicious adversary will try any attempts to get biometric information stored in a server or to be authenticated by the server. More seriously, in many previous works regarding PPBA, biometric templates are stored in the server in plain forms, so they can be leaked in the case of server compromise.

There have been active studies [34, 35, 36, 37, 38, 39] regarding MPC in malicious model. However, as [13] pointed out, MPC in malicious model accom-

[6] It takes only 0.05 seconds or less for 900-bit inputs, and computation time depends linearly on the bit size.

[7] They also proposed theoretic construction and security proof in the malicious model, but without implementation result.

panies inefficiency making it impractical to be applied to biometric authentication. Recently, outsourcing some computations of MPC [31, 40, 32, 41, 33, 13] have been proposed to resolve efficiency and/or security issues. However, they achieved practical performance on some small-sized circuits only. In a recent work of Gasti et al. [13], computing HD of two 1600-bit inputs takes 3.29 seconds.[8]

On the other hand, Somewhat Homomorphic Encryption (SHE) can be used in biometric authentication. Yasuda et al. [11, 12] proposed efficient HD-based biometric authentications in three-party setting and HBC model that exploit SHE based on ideal lattices [42, 43] or the Ring Learning with Errors problem [44]. The efficiency comes from their packing technique encrypting 2048-bit biometric into one ciphertext and representation of HD by one multiplication of ciphertexts. The later one [12] takes only 5.31 ms for 2048-bit matching. However, the security highly depends on the honest behavior of computation server, and suffers from simple hill-climbing attack [45] which enables malicious computation server to learn biometric templates. Abidin et al. [14] proposed another way to use HE with XOR-linear Message Authentication Code, and the protocol is held by three parties which are client, service provider, and cloud server, assuming the former two are semi-honest, but the cloud server is malicious. Implementation result is not reported in the paper, but the performance is expected to be much worse than those in [11, 12].

Fuzzy extractors or Secure sketches [46] can be used for template protection and authentication by handling noisy (or fuzzy) property of biometrics with error correcting techniques. Particularly, fuzzy extractor enables to extract reliable key from noisy biometrics. Only legitimate user with similar biometric to the pre-enrolled biometric can retrieve the valid key. However, due to its powerful functionality (both authentication and reliable key extraction), it has some deficiencies. It requires some conditions (*e.g.* sufficient min entropy) on the distribution of biometric, and has some security issues when used multiple times. Recently, Canetti et al. [47] mitigated the conditions on the distributions and resolved the security issues proposing reusable fuzzy extractors. However, the error correction rate (sublinear in [47]) tolerated by fuzzy extractors are still quite severe to be applied to some biometrics (especially, for iris which requires linear error tolerance [48, 49]).

Cancellable biometrics or biohashing [50, 51, 52, 53, 54] have been proposed to protect biometrics and to deal with the noisy nature of biometrics. It extracts features from raw biometric using non-invertible transformation with randomized token for cancellability, and provides low error rates and quick authentication process. However, the privacy of biometric in this methods is not completely provided as cryptographic hash function, and several analysis or attacks [55, 56, 57, 58] are known. Especially, the authentication accuracy (FAR or FRR) highly depends on the randomized token which is different among users,

[8] Measured on
client: "Samsung Galaxy S4 smartphone 4-Core 1.9GHz CPU (Qualcomm Snapdragon), 2GB RAM."
server & Cloud: "Intel Xeon E5-2430L v2 6-Core 2.4GHz CPU, 64GB RAM"

and the quality of authentication is deteriorated when the token is leaked [59]. Therefore, each user should keep their own token secretly, which weakens the versatility of biometric authentication. On the contrary, the authentication quality of our primitive is not deteriorated by an illegitimate user encrypting his/her biometric template with the same key (msk) as the legitimate user.

Many renowned researches we have not mentioned are classified to Privacy Preserving Biometric Identification (PPBI). They are usually based on MPC [60, 61, 62, 63, 64, 65], and enables a client to match his/her biometric to the database of server without revealing his/her input. We remark that they should be distinguished from biometric authentication which allows the server to check the matching result. One major difference is that authentication system is highly necessary to be secure under the presence of malicious client, since an adversary will actively attempt to be accepted by the server as a legitimate user or to capture the biometric data stored in the server during the authentication phase. In this sense, many PPBIs which are secure under the HBC model is not appropriate to be regarded as a solution for PPBA. In addition, many PPBIs consider that it is allowable for the server to manage biometric database in a plaintext form, and many PPBIs don't work efficiently if this is not allowed. However, storing raw biometric data in a server is a serious threat these days as we pointed out in Section 1.

8 Conclusion

Privacy-preserving biometric authentication is a protocol to authenticate users with their biometrics while preserving the privacy of biometric information. Due to the usability and high entropy of biometrics, many of the researches on PPBA has been done recently. However, currently proposed PPBAs with sufficient accuracy have drawbacks especially in speed to be applied in practice. In this work, we propose a new practical PPBA for Hamming distance matcher, which is secure against active attack under the standard LWE assumption. Our experimental results support the practical feasibility of our protocol toward the real world.

A Proof of Lemma 4.8

lemma 4.8. *For real numbers $\sigma_1, \sigma_2 > 0$, let e and f be variables of distributions D_{σ_1} and D_{σ_2}, respectively. Let $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$, then $(e+f, e|(e+f))$ follows the distribution $(D_\sigma, D_{L\sigma_1^2/\sigma^2,\ \sigma_1\sigma_2/\sigma})$ where L denotes the value of $e+f$.*

Proof. It is enough to show that

$$\Pr[e = x | e + f = y] = \frac{\sigma}{\sigma_1\sigma_2} \exp\left(-\frac{\pi\left(x - y\sigma_1^2/\sigma^2\right)^2}{\sigma_1^2\sigma_2^2/\sigma^2}\right).$$

We can check it by a direct computation as follow:

$$
\begin{aligned}
&\frac{\sigma_1\sigma_2}{\sigma} \cdot \Pr[e = x | e + f = y] \\
=&\frac{\sigma_1\sigma_2}{\sigma} \cdot \frac{\Pr[e = x] \cdot \Pr[f = y - x]}{\Pr[e + f = y]} \\
=&\exp\left(-\frac{\pi x^2}{\sigma_1^2} - \frac{\pi(y-x)^2}{\sigma_2^2} + \frac{\pi y^2}{\sigma_1^2 + \sigma_2^2}\right) \\
=&\exp\left(-\pi(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2})x^2 + \frac{2\pi y}{\sigma_2^2}x - \pi(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2 + \sigma_2^2})y^2\right) \\
=&\exp\left(-\pi(\frac{\sigma^2}{\sigma_1^2\sigma_2^2})x^2 + \frac{2\pi y}{\sigma_2^2}x - \pi(\frac{\sigma_1^2}{\sigma_2^2\sigma^2})y^2\right) \\
=&\exp\left(-\pi(\frac{\sigma}{\sigma_1\sigma_2}x - \frac{\sigma_1}{\sigma_2\sigma}y)^2\right) \\
=&\exp\left(-\frac{\pi(x - y\sigma_1^2/\sigma^2)^2}{\sigma_1^2\sigma_2^2/\sigma^2}\right).
\end{aligned}
$$

References

1. "Fido uaf architectural overview," `https://fidoalliance.org/specs/fido-uaf-v1.1-id-20170202/fido-uaf-overview-v1.1-id-20170202.html`.
2. S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric recognition: Security and privacy concerns," *IEEE security & privacy*, vol. 99, no. 2, pp. 33–42, 2003.
3. A. K. Jain, K. Nandakumar, and A. Nagar, "Biometric template security," *EURASIP Journal on advances in signal processing*, vol. 2008, p. 113, 2008.
4. C. Roberts, "Biometric attack vectors and defences," *Computers & Security*, vol. 26, no. 1, pp. 14–25, 2007.
5. D. Alexander, "5.6 million fingerprints stolen in u.s. personnel data hack: government," 2015. https://www.reuters.com/article/us-usa-cybersecurity-fingerprints/5-6-million-fingerprints-stolen-in-u-s-personnel-data-hack-government-idUSKCN0RN1V820150923.
6. R. Khaira, "Rs 500, 10 minutes, and you have access to billion aadhaar details," 2018. `http://www.tribuneindia.com/news/nation/rs-500-10-minutes-and-you-have-access-to-billion-aadhaar-details/523361.html`.
7. A. Jarrous and B. Pinkas, "Secure hamming distance based computation and its applications," in *International Conference on Applied Cryptography and Network Security*, pp. 107–124, Springer, 2009.
8. Y. Huang, D. Evans, J. Katz, and L. Malka, "Faster secure two-party computation using garbled circuits.," in *USENIX Security Symposium*, vol. 201, pp. 331–335, 2011.
9. K. Simoens, J. Bringer, H. Chabanne, and S. Seys, "A framework for analyzing template security and privacy in biometric authentication systems," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 833–841, 2012.
10. J. Bringer, H. Chabanne, and A. Patey, "Shade: Secure hamming distance computation from oblivious transfer," in *International Conference on Financial Cryptography and Data Security*, pp. 164–176, Springer, 2013.

11. M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, "Packed homomorphic encryption based on ideal lattices and its application to biometrics," in *International Conference on Availability, Reliability, and Security*, pp. 55–74, Springer, 2013.
12. M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, "Practical packing method in somewhat homomorphic encryption," in *Data Privacy Management and Autonomous Spontaneous Security*, pp. 34–50, Springer, 2014.
13. P. Gasti, J. Šeděnka, Q. Yang, G. Zhou, and K. S. Balagani, "Secure, fast, and energy-efficient outsourced authentication for smartphones," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2556–2571, 2016.
14. A. Abidin, A. Aly, E. A. Rúa, and A. Mitrokotsa, "Efficient verifiable computation of xor for biometric authentication," in *International Conference on Cryptology and Network Security*, pp. 284–298, Springer, 2016.
15. K. Zhou and J. Ren, "Passbio: Privacy-preserving user-centric biometric authentication," *IEEE Transactions on Information Forensics and Security*, 2018.
16. S. Rane, Y. Wang, S. C. Draper, and P. Ishwar, "Secure biometrics: Concepts, authentication architectures, and challenges," *IEEE Signal Processing Magazine*, vol. 30, no. 5, pp. 51–64, 2013.
17. S. Kim, K. Lewi, A. Mandal, H. Montgomery, A. Roy, and D. J. Wu, "Function-hiding inner product encryption is practical," in *International Conference on Security and Cryptography for Networks*, pp. 544–562, Springer, 2018.
18. A. Bishop, A. Jain, and L. Kowalczyk, "Function-hiding inner product encryption," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 470–491, Springer, 2015.
19. P. Datta, R. Dutta, and S. Mukhopadhyay, "Functional encryption for inner product with full function privacy," in *Public-Key Cryptography–PKC 2016*, pp. 164–195, Springer, 2016.
20. A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Fingercode: A filterbank for fingerprint representation and matching," in *Conference on Computer Vision and Pattern Recognition (CVPR*, p. 2187, IEEE Computer Society, 1999.
21. G. W. Quinn, J. R. Matey, and P. J. Grother, "Irex ix part one, performance of iris recognition algorithms," *NIST Interagency/Internal Report (NISTIR) - 8207*, 2018.
22. M. Barni, T. Bianchi, D. Catalano, M. Di Raimondo, R. Donida Labati, P. Failla, D. Fiore, R. Lazzeretti, V. Piuri, F. Scotti, and A. Piva, "Privacy-preserving fingercode authentication," in *Proceedings of the 12th ACM Workshop on Multimedia and Security*, pp. 231–240, 2010.
23. O. Regev, "The learning with errors problem," *Proceedings of the 2010 IEEE 25th Annual Conference on Computational Complexity*, pp. 191–204, 2010.
24. E. Käsper, "Fast elliptic curve cryptography in openssl," in *Proceedings of the 2011 International Conference on Financial Cryptography and Data Security*, FC'11, pp. 27–39, 2012.
25. D. Burke, "Previewing android p," `https://android-developers.googleblog.com/2018/03/previewing-android-p.html`.
26. M. R. Albrecht, R. Player, and S. Scott, "On the concrete hardness of learning with errors," *Journal of Mathematical Cryptology*, vol. 9, no. 3, pp. 169–203, 2015.
27. M. R. Albrecht, "On dual lattice attacks against small-secret lwe and parameter choices in helib and seal," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 103–129, Springer, 2017.

28. S. Bai and S. D. Galbraith, "Lattice decoding attacks on binary lwe," in *Australasian Conference on Information Security and Privacy*, pp. 322–337, Springer, 2014.
29. C. Karabat, M. S. Kiraz, H. Erdogan, and E. Savas, "Thrive: threshold homomorphic encryption based secure and privacy preserving biometric verification system," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 1, p. 71, 2015.
30. J. Katz and M. Yung, "Threshold cryptosystems based on factoring," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 192–205, Springer, 2002.
31. S. Kamara, P. Mohassel, and B. Riva, "Salus: a system for server-aided secure function evaluation," in *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 797–808, ACM, 2012.
32. H. Carter, C. Lever, and P. Traynor, "Whitewash: Outsourcing garbled circuit generation for mobile devices," in *Proceedings of the 30th Annual Computer Security Applications Conference*, pp. 266–275, ACM, 2014.
33. H. Carter, B. Mood, P. Traynor, and K. Butler, "Secure outsourced garbled circuit evaluation for mobile devices," *Journal of Computer Security*, vol. 24, no. 2, pp. 137–180, 2016.
34. P. Mohassel and M. Franklin, "Efficiency tradeoffs for malicious two-party computation," in *International Workshop on Public Key Cryptography*, pp. 458–473, Springer, 2006.
35. M. S. Kiraz, "Secure and fair two-party computation," in *Proc. of EUROCRYPT*, vol. 3, 2008.
36. Y. Ishai, M. Prabhakaran, and A. Sahai, "Founding cryptography on oblivious transfer–efficiently," in *Annual International Cryptology Conference*, pp. 572–591, Springer, 2008.
37. V. Goyal, P. Mohassel, and A. Smith, "Efficient two party and multi party computation against covert adversaries," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 289–306, Springer, 2008.
38. A. Shelat and C.-h. Shen, "Two-output secure computation with malicious adversaries," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 386–405, Springer, 2011.
39. Y. Lindell and B. Pinkas, "Secure two-party computation via cut-and-choose oblivious transfer," *Journal of cryptology*, vol. 25, no. 4, pp. 680–722, 2012.
40. H. Chun, Y. Elmehdwi, F. Li, P. Bhattacharya, and W. Jiang, "Outsourceable two-party privacy-preserving biometric authentication," in *Proceedings of the 9th ACM symposium on Information, computer and communications security*, pp. 401–412, ACM, 2014.
41. J. Šeděnka, S. Govindarajan, P. Gasti, and K. S. Balagani, "Secure outsourced biometric authentication with performance evaluation on smartphones," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 384–396, 2015.
42. C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," in *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, (New York, NY, USA), pp. 169–178, ACM, 2009.
43. C. Gentry and S. Halevi, "Implementing gentry's fully-homomorphic encryption scheme," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 129–148, Springer, 2011.
44. M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?," in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pp. 113–124, ACM, 2011.

45. A. Abidin and A. Mitrokotsa, "Security aspects of privacy-preserving biometric authentication based on ideal lattices and ring-lwe," in *Information Forensics and Security (WIFS), 2014 IEEE International Workshop on*, pp. 60–65, IEEE, 2014.
46. Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM journal on computing*, vol. 38, no. 1, pp. 97–139, 2008.
47. R. Canetti, B. Fuller, O. Paneth, L. Reyzin, and A. Smith, "Reusable fuzzy extractors for low-entropy distributions," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 117–146, Springer, 2016.
48. A. I. Desoky, H. A. Ali, and N. B. Abdel-Hamid, "Enhancing iris recognition system performance using templates fusion," *Ain Shams Engineering Journal*, vol. 3, no. 2, pp. 133–140, 2012.
49. K. P. Hollingsworth, K. W. Bowyer, and P. J. Flynn, "Improved iris recognition through fusion of hamming distance and fragile bit distance," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 12, pp. 2465–2476, 2011.
50. N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM systems Journal*, vol. 40, no. 3, pp. 614–634, 2001.
51. A. T. B. Jin, D. N. C. Ling, and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," *Pattern recognition*, vol. 37, no. 11, pp. 2245–2255, 2004.
52. R. Ang, R. Safavi-Naini, and L. McAven, "Cancelable key-based fingerprint templates," in *Australasian conference on information security and privacy*, pp. 242–252, Springer, 2005.
53. S. Kanade, D. Petrovska-Delacrétaz, and B. Dorizzi, "Cancelable iris biometrics and using error correcting codes to reduce variability in biometric data," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 120–127, IEEE, 2009.
54. C. Rathgeb and A. Uhl, "Iris-biometric hash generation for biometric database indexing," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, pp. 2848–2851, IEEE, 2010.
55. K. H. Cheung, A. W.-K. Kong, J. You, and D. Zhang, "An analysis on invertibility of cancelable biometrics based on biohashing.," in *CISST*, vol. 2005, pp. 40–45, 2005.
56. A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, and J. You, "An analysis of biohashing and its variants," *Pattern recognition*, vol. 39, no. 7, pp. 1359–1368, 2006.
57. Y. Lee, Y. Chung, and K. Moon, "Inverse operation and preimage attack on biohashing," in *Computational Intelligence in Biometrics: Theory, Algorithms, and Applications, 2009. CIB 2009. IEEE Workshop on*, pp. 92–97, IEEE, 2009.
58. P. Lacharme, E. Cherrier, and C. Rosenberger, "Preimage attack on biohashing," in *Security and Cryptography (SECRYPT), 2013 International Conference on*, pp. 1–8, IEEE, 2013.
59. K.-H. Cheung, A. Kong, D. Zhang, M. Kamel, and J. You, "Revealing the secret of facehashing," in *International Conference on Biometrics*, pp. 106–112, Springer, 2006.
60. Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *International Symposium on Privacy Enhancing Technologies Symposium*, pp. 235–253, Springer, 2009.
61. A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," in *International Conference on Information Security and Cryptology*, pp. 229–244, Springer, 2009.

62. M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich, "Scifi-a system for secure face identification," in *Security and Privacy (SP), 2010 IEEE Symposium on*, pp. 239–254, IEEE, 2010.
63. D. Evans, Y. Huang, J. Katz, and L. Malka, "Efficient privacy-preserving biometric identification," in *Proceedings of the 17th conference Network and Distributed System Security Symposium, NDSS*, 2011.
64. M. Blanton and P. Gasti, "Secure and efficient protocols for iris and fingerprint identification," in *European Symposium on Research in Computer Security*, pp. 190–209, Springer, 2011.
65. J. Bringer, H. Chabanne, M. Favre, A. Patey, T. Schneider, and M. Zohner, "Gshade: faster privacy-preserving distance computation and biometric identification," in *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*, pp. 187–198, ACM, 2014.