PBio: Enabling Cross-organizational Biometric Authentication Service through Secure Sharing of Biometric Templates

Anonymous Author(s)

ABSTRACT

Biometric authentication has been deployed to authenticate users locally to unlock phones and applications. Increasingly, proposals were made to extend the use of biometric to remotely authenticate users through matching the captured biometric features with centrally stored templates. However, security and privacy remain a main concern due to potential leakage of these templates. There are privacy-preserving biometric authentication schemes that address this concern. Most of these schemes consider a one-to-one setting, where an enterprise registers and authenticates users using the registered biometric information.

In this work, we consider a new setting. A trusted organization in possession of a raw biometric database shares its database to enable other organizations to provide authentication services through a derived (partial or whole) dataset, without these other organizations learning the underlying biometric information. Given such an option, these organizations do not need to collect users' biometrics, or always connect to the central raw database, thus reducing the number of potential attack points. We propose a privacy-preserving biometric authentication system, PBio, for the above setting. The core component of PBio is a new protocol that we design, which comprises of a distance-preserving encryption scheme and secure distance computation. The organizations only hold a derived, encrypted biometric dataset and need not be fully trusted. PBio is secure even when the organizations collude. We also introduce an encrypt-then-split mechanism such that each of the entities holds encrypted partial templates. It enables faster verification for non-match instances in our early rejection setting and reduces risk of template reconstruction in the event that an encrypted partial template database and its encryption key are leaked. We analyse the security of our system. Our experiment demonstrated that our scheme is practical, and can be readily deployed. Our system achieves an encrypted facial image verification time of around 3.21 milliseconds. While this is higher than some of the recent state-of-the-arts in our experiments, the overhead is within the acceptable range, and is a reasonable trade-off to provide such cross-organizational authentication services.

KEYWORDS

Privacy-preserving; Biometric; Orthogonal matrices; Euclidean distance; Data security and privacy; Encrypted data computation

1 INTRODUCTION

With the increasingly widespread usage of biometric information such as fingerprint, face and iris for authentication purpose, providing organizations access to a collection of comprehensive biometric templates hosted by a trusted organizations will enable more effective authentication of an individual, instead of relying only on the documentation carried by the said individual. The benefits are two fold. First it allows organizations that currently have no access or require lengthy administrative and legal processes to have direct access to a readily available database. Secondly, these agencies and private entities do not need to invest in the infrastructure to register users and construct a biometric database of their own. This also reduces potential breaches of the templates, especially if there are many different copies residing in each of the organizations.

However, having many different agencies and private entities directly accessing the database increases the possibility of breaches that may cause the templates to be leaked. A straightforward solution is to share the template database to these organizations. There was a general assumption that the widely used raw biometric templates were secure. For instance, it was believed that a binary template (e.g. Iriscode) does not contain sufficient information to enable its reconstruction [1]. However in more recent times, significant progress has been made in the domain of biometric template reconstruction [10, 11, 21, 23, 43]. This is a major privacy concern since the reconstructed biometric templates can be used to identify or impersonate an individual.

Therefore, a more viable approach is to share a derivation (i.e. via encryption) of the database that would enable other organizations to authenticate the said individuals, but in such a way that the organizations are not able to learn any biometric information from the derived database. Also in this case, the raw biometric templates remain with the trusted organization and is isolated from being accessed by other parties.

There are schemes that provide privacy-preserving biometric authentication, but most of them consider a direct user-to-organisation registration and authentication setting. In other words, a user registers his biometrics with an organization. At a later stage the user authenticates to the organisation using his biometrics [5, 7, 20, 41, 63, 83]. The organization may also outsource the computation or storage to a cloud service [14, 26, 28, 70, 73, 76, 81, 82]. There are a number of notable differences between our biometric system with these existing works. Here, we consider a new setting involving four or more parties where a trusted organization shares a derived database with other organizations in a privacy-preserving manner. The parties are the end-users, one or more data subscribers, the cloud provider and the data owner. In this context, the data owner is a trusted organization in possession of the biometric database while the data subscribers are the respective service providers (i.e. other organizations such as hotels, banks).

Moreover, our solution avoids the usage of computationally expensive homormorphic encryption tools implemented in many existing works [5, 14, 20, 63, 70]. Instead, we utilize lightweight encryption which enables fast authentication using distance preserving transformation. Since there exists inherent security weaknesses in such transformations, we augment our systems with a number of other mechanisms to overcome such limitations. Firstly, each end-user is assigned a unique secret key (i.e. distinct from

other end-users). Secondly, our biometric system implements a secure distance protocol such that neither the data subscriber nor the cloud has access to the numerical similarity match score during the authentication phase. Thirdly, periodical refreshments of the encrypted biometric templates are performed. We discuss several ways to go about this update process in a later section.

Our Contributions. We propose a privacy-preserving biometric authentication system, PBio, which enables secure sharing of a biometric database to one or more organisations. Our contributions are summarized as follows.

- We encrypt raw biometric templates using a transformation with orthogonal matrix, which preserve Euclidean distance. To overcome the weakness in linearity, we use this distance preserve encryption like a *one-time pad*, that is, in legitimate usage, an encryption key will never be re-used to encrypt a different object. Furthermore, we employ existing secure two party computation (GSHADE), to allow two parties with two private vector inputs $\vec{x}$ and $\vec{y}$, respectively, to securely decide if the Euclidean distance between $\vec{x}$ and $\vec{y}$ is smaller than a given threshold without leaking extra information. The above ideas constitute the cryptography core of our proposed biometric authentication solution.
- Our protocol also includes a new encrypt-then-split construction. Here, the derived and encrypted biometric templates are split into two or more copies where one copy is given to a cloud service provider and the other copies to organisations that subscribe to the authentication services. During verification, a captured biometric feature can be tested with the encrypted, partial template hosted by the organisation. Our protocol is able to compute a partial result in such a way that if the computed distance is over a predefined threshold then the result is a no match. Only when the result is ambiguous will a second computation involving the cloud provider be required to ascertain the final result. The advantages of splitting the biometric template is two-fold. The first is that each split portion of a biometric template has a smaller dimension compared to its entirety. As such, this enables early rejection of a non-matching biometrics during the authentication phase. The second advantage is to ensure no single entity has in possession the full raw biometric template of any user. This is critical as recent research has shown that the original features or images can be reconstructed from various types of raw biometric templates. The encryption, on the other hand, circumvent the issue of partial information leakage of a user's features due to the partial biometric template. We tested our construction on public datasets of 853 biometric facial images and achieve identical accuracy in comparison with one without encryption.
- Our protocol provides security against collusion between two or more entities. No collusion of entities can derive the full raw biometric template of any user without the secret key of the encryption. Our encryption employed is lightweight, collision-free and is compatible with our splitting mechanism. To our knowledge, this is the first such hybrid encrypt-split approach which provides multiple security guarantees with practical running time.
- Based on our experiments and the prototype developed, our scheme is efficient and practical. In particular, the time taken to generate a single 128 dimensional encrypted facial template is under 1.7 milliseconds. Furthermore, the time taken to verify a user's encrypted biometric feature is only around 3.21 milliseconds.

2 SYSTEM OVERVIEW

In this section, we describe the different parties, system architecture and security models of our proposed privacy-preserving biometric authentication system, PBio.

2.1 Parties Involved

Our system assumes there are four parties, which we define in the followings.

- Data Owner: A fully trusted party, e.g. a trusted organization who owns the raw biometric information and templates. The data owner outsources the encrypted biometric database.
- Cloud Provider: This is an honest-but-curious party, e.g. a cloud, who stores encrypted biometric database and helps to verify an individual without the need of decrypting an encrypted template.
- Data Subscribers: These are honest-but-curious parties, e.g. banks and malls who subscribe to the system to authenticate a user.
- Users: Users e.g. clients of a bank and customers of a mall who submit their biometric information for authentication. Users are not trusted but we assume there exists a tamper-proof device that extracts and encrypts user's biometric information into an encrypted template.

2.2 System Architecture

Figure 1 illustrates the overview of PBio system during the setup and registration phase. First of all, the data owner performs key generation to obtain a master key. The data owner then derives the user key for every m of users. A data owner encrypts then splits the biometric templates into two parts. In brief, the encrypt-then-split approach allows the data owner to provide a partial copy of the encrypted templates to the cloud provider and another partial copy to the data subscriber i. For every data subscriber i, the data owner again encrypts the encrypted partial templates using the respective derived sub key from the user key. It thus reduces risk of leakage of the full templates if one of the party's encrypted templates are compromised. Furthermore by doing so, it is possible to first perform matching based on the partial biometric templates hosted by the data subscriber i, reducing the computation and communication cost during the authentication phase in the event of an early rejection, which is illustrated in Figure 2. The implementation is discussed in Section 5.

2.3 Definitions

This section defines our scheme, which consists of the following algorithms:

- $Setup$: On input security parameter 1^k, it outputs system parameter PM.
- $MKGen$: On input PM, it generates a master secret key msk for the data owner.
- $KeyGen$: On input (PM, msk) and the user unique identity ID, it generates a user secret key $skID$.

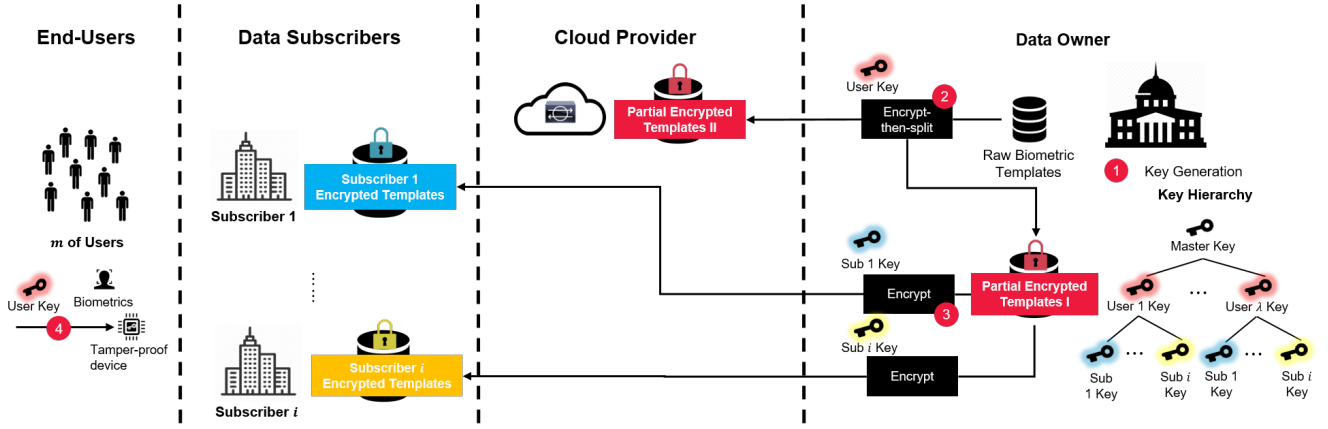

Figure 1: PBio: Setup & Registration Diagram. (1) Generate cryptography keys; (2) Encrypt-then-split the raw biometric templates; (3) Generate subscriber template for every subscriber; (4) Deliver user key for every registered user device.

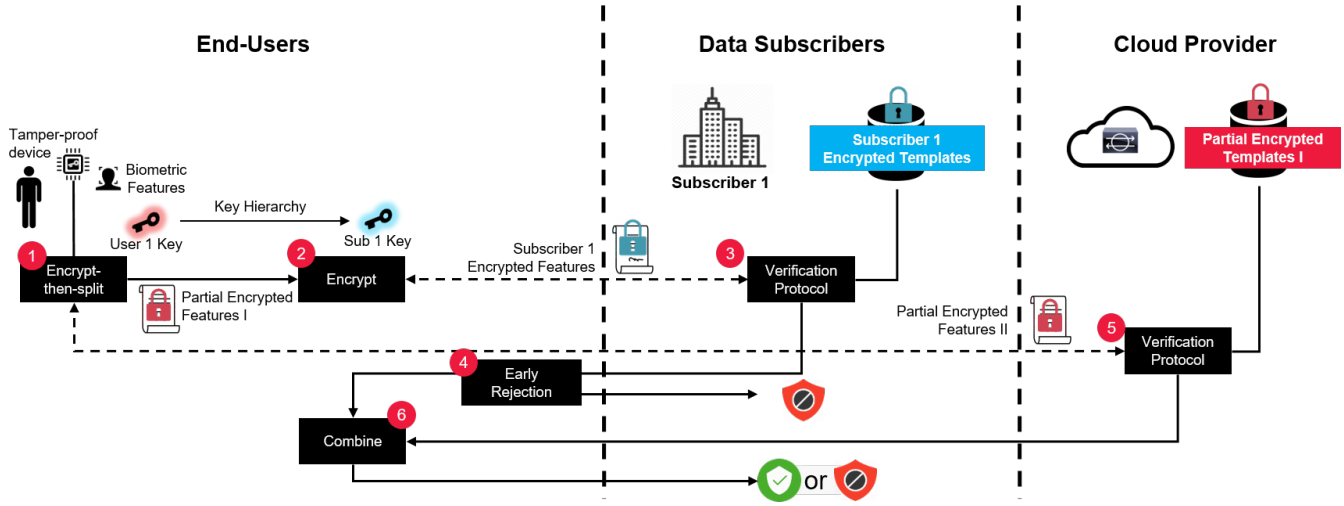

Figure 2: PBio: Authentication Diagram. (1) Encrypt-then-split freshly submitted biometric features; (2) Generate the subscriber encrypted features; (3) Verify the subscriber encrypted features; (4) Reject if step (3) is invalid; (5) Proceed to verify the remaining part of encrypted features; (6) Combine the final authentication result from (3) and (5).

- *Enc*: On input (PM, sk_{ID}) and user biometric templates in n-dimension vector $\vec{x}_i = \{x_1, \cdots, x_n\}$, it computes an encrypted vector in n-dimension $\vec{c}_{ID}$.
- *ReEnc*: On input $(PM, sk_{ID}, \vec{c}_{ID})$ and subscriber identity i, it computes a re-encrypted vector $\vec{c}_{ID,i}$.
- *EncT* On input (PM, sk_{ID}, i) where i is optional, and a threshold t that serves to authentication a person, it computes an encrypted threshold $t_{ID,i}$.
- *Ver*: On input PM, a tuple of encrypted vectors $(\vec{c}_0, \vec{c}_1)$ which is encrypted with the same sk, and authenticated threshold value t, it computes their distance d. The output is "1" if $d \leq t$ and "0".

3 THE PROTOCOL

In this section, we describe in detail our proposed scheme termed PBio, where the cloud provider and the data subscriber i each hosts a partial copy of the encrypted biometric template database (Figure 1 and 2). We first state the main components deployed in our scheme.

3.1 Main Components

In the followings we describe the building blocks. Our scheme deploys a biometric recognition scheme to extract features and construct templates from the raw biometric information (e.g. fingerprint, face, iris). A distance-recoverable encryption is proposed to encrypt these templates. For authentication, a secure distance computation mechanism is used.

3.1.1 Biometrics Recognition Scheme. A biometrics recognition system requires feature extraction to transform raw biometric traits (e.g., fingerprints, voice patterns, facial patterns, iris, etc.) into templates. The extracted features are then called feature vectors

with n elements. Given two feature vectors $\vec{x} = (x_1, \cdots, x_n)$ and $\vec{y} = (y_1, \cdots, y_n)$, one metric of matching is the squared Euclidean distance [1], which is defined as:

$$\begin{aligned} Dist(\vec{x}, \vec{y}) &= \sum_{i=1}^{n} (x_i - y_i)^2 \\ &= \sum_{i=1}^{n} x_i^2 - 2x_i y_i + y_i^2 \\ &= \sum_{i=1}^{n} x_i^2 - 2\sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} y_i^2 \\ &= \|\vec{x} - \vec{y}\|_2^2 \\ &= d \end{aligned}$$

The authentication result is based on the Squared Euclidean distance in relation to the defined threshold t. In particular, $\vec{x}$ and $\vec{y}$ belong to the same person if and only if $Dist(\vec{x}, \vec{y}) \leq t^2$, which indicates a match. Note that the lower value of t means the system requires higher similarity to pass the authentication. We define a biometric recognition scheme which consists of the following components:

- Ext: On input a raw biometric trait, it outputs a feature vector $\vec{x}$
- $Dist$: On input two feature vectors $(\vec{x}, \vec{y})$, it outputs a distance $d = Dist(\vec{x}, \vec{y})$
- $Match$: On input a threshold t and d, it outputs "1" if $d \leq t^2$ and "0" otherwise.

3.1.2 Distance-Recoverable Encryption. Distance-recoverable encryption (DRE) allows one to calculate the distance between two encrypted data points such that the distance between two plain data points is equal to the distance between the corresponding two encrypted data points, i.e. $Dist(x, y) = Dist(E(x), E(y))$. A distance-preserving transformation (DPT) [55] scheme is an example of DRE which can be instantiated with orthogonal matrices.

Orthogonal Matrix. An orthogonal matrix M is a $n \times n$ square matrix such that its inverse and transpose are equal, i.e. $M^{-1} = M^\top$. M satisfies the following properties:

- Identity transformation: M and $M^\top$ commute such that $M^\top M = MM^\top = I$, where I is the identity matrix.
- Product transformation: Given $M = M_0 M_1$, if M_0 and M_1 are orthogonal matrices, M is also an orthogonal matrix.
- Preservation of length: Given two pairs of vectors $(\vec{x}, \vec{y})$ and $(\vec{x}M, \vec{y}M)$, their respective Euclidean distances are equal as given by $Dist(\vec{x}, \vec{y}) = Dist(\vec{x}M, \vec{y}M) = \|\vec{x}M - \vec{y}M\|_2^2 = \|\vec{x} - \vec{y}\|_2^2$.

Distance-Preserving Transformation. Let $E(\cdot, \cdot)$ be an encryption function with the input of n-dimension vector $\vec{x} = (x_1, \cdots, x_n)$ and secret key $sk = (w, \vec{v}, M)$ that outputs an encrypted vector $E(\vec{x}, sk) = w \cdot (\vec{x} + \vec{v}) \cdot M$ such that M is an $n \times n$ orthogonal matrix, $\vec{v}$ is a random vector, and w is a scale factor. The distance between two encrypted vectors $(E(\vec{x}, sk), E(\vec{y}, sk))$ is as follows:

$$\begin{aligned} Dist(E(\vec{x}, sk), E(\vec{y}, sk)) &= \|(w \cdot (\vec{x} + \vec{v}) \cdot M) - (w \cdot (\vec{y} + \vec{v}) \cdot M)\|_2^2 \\ &= w^2 \|\vec{x} - \vec{y}\|_2^2 \\ &= w^2 \cdot d \end{aligned}$$

[1] So we can save computation of square root every time.

Proposition 3.1. *E is collision-free under the same secret key.*

Proof. Suppose $E(\vec{x}, sk) = E(\vec{y}, sk)$ for some $\vec{x}, \vec{y}$. Then

$$w \cdot (\vec{x} + \vec{v}) \cdot M = w \cdot (\vec{y} + \vec{v}) \cdot M \implies \vec{x} + \vec{v} = \vec{y} + \vec{v}$$

since $\det(M) \neq 0$. It follows that $\vec{x} = \vec{y}$. □

Security of DPT. Obviously, DPT is easily broken by solving a large linear equation system, if the adversary obtains sufficient pairs of plaintexts and ciphertexts [44]. As shown in [75], such DPT scheme can resist ciphertext-only attack. To overcome the weakness of DPT in our application, we ensure that every user will use different encryption key sk. Furthermore, different fingerprints (e.g. thumb finger and index finger) of the same person may also be encrypted with different keys. Basically, we use DPT like "One-Time Pad". In legitimate usage, the encryption key will never be re-used for different objects.

Theorem 3.2 (Security of our DPT). *Let $\vec{x}$ and $\vec{y}$ denote two points in the plaintext domain, and $\vec{c}$ is any valid ciphertext where the encryption key is randomly chosen from its domain. We have*

- $$\Pr[\vec{x}|\vec{c}] = \Pr[\vec{y}|\vec{c}], \tag{1}$$
  which means a single ciphertext leaks no information of the plaintext.
- *Let real number $t \in (0, 1)$ be the threshold. Given a ciphertext, there are at least $1/t^n$ number of possible plaintexts under distinct encryption keys, such that the distance between every two plaintexts is at least $2t$.*

Proof. Part I.

Recall that our DPT is

$$E(sk, \vec{x}) = w \cdot (\vec{x} + \vec{v}) \cdot M \tag{2}$$

where $sk = (w, \vec{v}, M)$ and w. Let encryption key $sk = (w, \vec{v}, M)$ and ciphertext $\vec{c} = E(sk, \vec{x}) = w \cdot (\vec{x} + \vec{v}) \cdot M$, we have

$$\vec{v} = f(w, M, c, \vec{x}) = w^{-1} \cdot \vec{c} \cdot M^{-1} - \vec{x} \tag{3}$$

Therefore, we show that the conditional probability $\Pr[\vec{x}|\vec{c}]$ is independent on $\vec{x}$:

$$\Pr[\vec{x}|\vec{c}] = \Pr[sk = (w, \vec{v}, M) \text{ where } \vec{v} = f(w, M, c, \vec{x})] \tag{4}$$

$$= \sum_{w,M} \Pr[sk = (w, \vec{v} = f(w, M, c, \vec{x}), M)] \tag{5}$$

$$= \frac{\#w \times \#M}{\#sk}, \tag{6}$$

where $\#w$ denotes total number of possible scaling factor, $\#M$ denotes total number of possible orthogonal matrix of dimension as indicated in our DPT scheme, and $\#sk$ denotes the total number of possible encryption keys in our scheme. As a result, for any two distinct plaintexts $\vec{x}$ and $\vec{y}$, we have

$$\Pr[\vec{x}|\vec{c}] = \Pr[\vec{y}|\vec{c}] \tag{7}$$

Part II.

Let t be the threshold. We request the distance between the two points $\vec{x}$ and $\vec{y}$ to be larger than $2t$, so they cannot represent the same bio-object. Then we count how many such distinct points with pairwise distance $\geq 2t$. Within the n-dimension cube $[-1.0, 1.0]^n$, each person's biometric measurement could be treated as a n-dimension

sphere with center u and radius t, where u is the measurement of the dimension during the registration phases. So the total number N of such n-dimension sphere is

$$N \geq \frac{2.0^n}{(2t)^n} = (1/t)^n. \quad (8)$$

For example, in our experiment, $t = 0.6$. □

3.1.3 Secure Distance Computation. Bringer et al. proposed GSHADE [7], the generalization of SHADE [8], which allows two parties, a sender S and a verifier V, to securely compute the distance of two biometric features. It guarantees one party does not get more information about the other party's inputs than what can be deduced from its own inputs and outputs. A central building block for the secure distance computation of GSHADE is oblivious transfer (OT). Oblivious transfer is an interactive protocol whereby the sender has a number of messages, and the receiver wishes to obtain a specific one, without the sender knowing which it is, while also ensuring that the receiver gets no information about the other messages which the sender holds. In brief, let $\vec{x} = (x_1, \cdots, x_k)$ with $x_i = (x_{k(i-1)+1}, \cdots, x_{k(i-1)+\ell})$ and $\vec{y} = (y_1, \cdots, y_k)$ with $y_i = (y_{k(i-1)+1}, \cdots, y_{k(i-1)+\ell})$ are $n = k \times \ell$-bit integer vectors. Three functions are defined where $f_x(\vec{x}) = \sum_{i=1}^{k} x_i^2$, $f_y(\vec{y}) = \sum_{i=1}^{k} y_i^2$, and $f_{k(i-1)+j}(x_{n(i-1)+j}, \vec{y}) = -2^j \cdot x_{k(i-1)+j} \cdot y_i$ for $i = 1, \cdots, k$ and $j = 1, \cdots, \ell$. S and V run the protocol as follows:

- S and V on input $\vec{y}$ and $\vec{x}$ respectively
- S chooses n random values $r_1, \cdots, r_n \in_R \mathbb{Z}_m$
- For each $i = 1, \cdots, n$, S and V engage in a $OT^{log_2(m)}$ where
  - V's selection bit is x_i
  - S's input is $(r_i + f_i(0, \vec{y}), r_i + f_i(1, \vec{y}))$
  - The output obtained by V is $t_i = r_i + f_i(x_i, \vec{y})$
- V computes and outputs $T = \sum_{i=1}^{n} t_i + f_x(\vec{x})$
- S computes and outputs $R = \sum_{i=1}^{n} r_i - f_y(\vec{y})$
- At the end, either S or V learns the distance by computing $Dist(\vec{x}, \vec{y}) = T - R = d$

Secure Comparison Protocol. As stated in [7], GSHADE allows one to add on GMW protocol [24], which on input the partial results T, R and a threshold t, to compute $T - R = d$ and check if $d \leq t^2$ in a secure manner such that one does not learn the distance.

Theorem 3.3. *[7] Security is proven by simulation in the OT-hybrid setting, where OTs are simulated by a trusted oracle. We recall that each simulator is provided with the input and output of the corrupted party. Case 1 - V is corrupted. Since V receives no messages beyond those in OT, its view can be perfectly simulated. Case 2 - S is corrupted. Given S's output T and input X, S's view can be perfectly simulated by sending random values $t_1, ..., t_{n-1} \in_R Z_m$ and $t'_n = T - \sum_{i=1}^{n-1} t'_i - f_x(\vec{x})$ to S in the OTs.*

3.2 PBio: Construction

We first define $BR = \{Ext, Dist\}$ to be any biometric recognition scheme based on Euclidean distance where a raw biometric image from ID is provided to BR to extract a vector, such that $BR.Ext(img_{ID}) \rightarrow \vec{x}_{ID} \in \mathbb{R}^n$, and $GSHADE(\cdot, \cdot) \rightarrow d$ be a secure distance computation protocol that on input two vectors, it outputs the distance d. In addition, we define a pseudorandom permutation function $PRP(\cdot, \cdot)$, which is run during the encryption $Enc(\cdot)$. $PRP(\cdot, \cdot)$ allows the data owner to reorder the user biometric templates $\vec{x} = (x_1, \cdots, x_n)$ where n is the dimension of vector. The proposed scheme consists of a tuple $\{Setup, MSKGen, KeyGen, Enc, EncT, ReEnc, Ver\}$ as follows.

- $Setup(1^k)$: It uses DRE to generate pseudorandom orthogonal function $PRF_M(\cdot) \rightarrow M \in \mathbb{R}^{n \times n}$, pseudorandom vector function $PRF_V(\cdot) \rightarrow \vec{v} \in \mathbb{R}^n$, pseudorandom scale function $PRF_W(\cdot) \rightarrow w \in [1.0, 2.0]$, and pseudorandom permutation function $PRP(\cdot, \cdot)$ which reorders the given vector based on the given secret and ID. The three PRF functions should have the same common input such as a secret key and ID, and PRP function should have an additional input such as a vector. The final output is a system parameter $PM = (BR, PRF_M, PRF_V, PRF_W, PRP\})$
- $MSKGen(PM)$: It randomly returns a master secret key $msk \in \mathbb{Z}_p^*$.
- $KeyGen(PM, msk, ID)$: This algorithms runs a keyed-hash message authentication code $HMAC$ with the input of msk, user unique identity ID, and a timestamp $time$ to generate a user secret key sk_{ID}.
- $Enc(PM, sk_{ID}, \vec{x}_{ID})$: This algorithm runs $PRF_M(sk_{ID}) \rightarrow M_{ID}$, $PRF_V(sk_{ID}) \rightarrow \vec{v}_{ID}$, and $PRF_W(sk_{ID}) \rightarrow w_{ID}$. It then runs $PRP(\vec{x}_{ID}, (sk_{ID})) \rightarrow \vec{x'}_{ID}$, and the encrypted vector is then generated such that $\vec{c}_{xID} = w_{ID}(\vec{x'}_{ID} + \vec{v}_{ID})M_{ID}$
- $ReEnc(PM, sk_{ID}, i, \vec{c}_{xID})$: This algorithm runs $PRF_M(sk_{ID}, i) \rightarrow M_{ID,i}$, $PRF_V(sk_{ID}, i) \rightarrow \vec{v}_{ID,i}$, and $PRF_W(sk_{ID}, i) \rightarrow w_{ID,i}$. It then runs $PRP(\vec{c}_{ID}, (sk_{ID}, i)) \rightarrow \vec{c'}_{ID}$, and the encrypted vector is then generated such that $\vec{c'}_{xID,i} = w_{ID,i}(\vec{c'}_{ID} + \vec{v}_{ID,i})M_{ID,i}$
- $EncT(PM, sk_{ID}, i, t)$: This algorithms runs $PRF_W(sk_{ID}) \rightarrow w_{ID}$ and $PRF_W(sk_{ID}, i) \rightarrow w_{ID,i}$. It then encrypts a threshold $t_{ID,i} = w_{ID} \cdot w_{ID,i} \cdot t$.
- $Ver(PM, \vec{c}_{xID}, \vec{c}_{y_{ID}}, t)$: An interactive protocol $GSHADE$ that is run by party A and B where A on input $(\vec{c}_{xID}, t_{ID})$ and B on input $\vec{c}_{y_{ID}}$. At the end of the protocol, either one party can receive the "1" or "0" which indicates the authentication result, such that $BR.Match(t_{ID}, d)$ = "1" or "0".

Correctness. Given the same system parameter PM, the following verification always holds:

- $Ver(PM, \vec{c}_{xID}, \vec{c}_{y_{ID}}, t_{ID}) = 1$ if $\vec{c}_{xID} = Enc(PM, sk_{ID}, \vec{x}_{ID}) \wedge \vec{c}_{y_{ID}} = Enc(PM, sk_{ID}, \vec{y}_{ID}) \wedge t_{ID} = EncT(PM, sk_{ID}, t)$
- $Ver(PM, \vec{c}'_{xID,i}, \vec{c}'_{y_{ID,i}}, t'_{ID,i}) = 1$ if $\vec{c}'_{xID,i} = ReEnc(PM, sk_{ID}, i, \vec{c}_{xID}) \wedge \vec{c}'_{y_{ID,i}} = ReEnc(PM, sk_{ID}, i, \vec{c}_{y_{ID}}) \wedge t'_{ID} = EncT\ (PM, sk_{ID}, i, t_{ID})$

3.3 PBio System

The proposed PBio consists of three phases: (1) setup phase, (2) registration phase, and (3) authentication phase.

Setup Phase

In the setup phase, the data owner has a set of biometric templates, which is pre-collected from all the users. Such that the users have registered their identity id_U along with their raw biometric Bio_U (e.g. fingerprints, face) with a data owner. Firstly,

The registration protocol

Input:

- Data owner inputs the encrypted partial database $\vec{c0}_{xID}$ and the user table $Table_U$.
- Subscriber i inputs the proof of identity.
- User inputs the unique identity ID along with the proof.

Output:

- Subscriber i obtains the subscriber i encrypted template $\vec{c}'_{ID,i}$.
- User obtains the user key sk_{ID}.

Protocol:

1 The protocol can be initialed by the subscriber in (a) or the user in (b):
   (a) The subscriber contacts data owner to request a copy of encrypted template. We assume the subscriber will proof the subscription in a secure manner.
   (b) The user request the user key sk_{ID} by giving a proof of identity.
2 Upon receiving the request from either (a) or (b), data owner performs the following steps respectively:
   (a) Data owner verifies the provided subscription and generates the subscriber i encrypted template $\vec{c}'_{ID,i}$ by running $ReEnc(PM, sk_{ID}, i, \vec{c0}_{xID})$ using every user secret key sk_{ID} and encrypted partial template $\vec{c0}_{xID}$.
   (b) Data owner verifies the user identity and embeds sk_{ID} to the user registered device.

Figure 3: Registration phase

the data owner runs setup and the master key generation functions to generate system parameter $Setup(1^k) \rightarrow PM$ and master secret key $MSKGen(PM) \rightarrow msk$. The data owner then applies a biometric recognition scheme BR (e.g. fingercode for fingerprints) to extract the biometric feature and stores the biometric template $BR(Bio_U) \rightarrow \vec{x}$. For every user with unique identity ID, the data owner runs key generation to generate user secret key $KeyGen(PM, msk, ID) \rightarrow sk_{ID}$. The data owner stores $(sk_{ID}, ID, time)$ in a user table $Table_U$. For every user biometric template $\vec{X} = \{\vec{x}_1, \cdots, \vec{x}_m\}$ where m is the total number of users, data owner generates the encrypted database by running $Enc(PM, sk_{ID}, \vec{x}_{ID}) \rightarrow \vec{c}_{xID}$. Data owner splits the encrypted database into two parts e.g. $\{\vec{c0}_x \mid\mid \vec{c1}_x\} = \vec{c}_x$. The first part $\vec{c0}_x$ will be applied during the registration phase and the second part $\vec{c1}_x$ is outsourced to a cloud.

Registration Phase

In the registration phase, it consists of two components: (a) subscriber registration and (b) user registration. The details of the registration protocol is described in Figure 3. In precise, the new subscriber may register and receive the encrypted template, and the new user may also register a device to install the user key for the authentication service.

The Authentication protocol

Input:

- User inputs freshly captured biometric image Bio_U along with his ID and sk_{ID}.
- Subscriber i inputs the subscriber i encrypted template $\vec{c}'_{ID,i}$ that belongs to the user ID.
- Cloud inputs the encrypted partial template $\vec{c1}_{xID}$ that belongs to user ID.

Output:

- Both the user and subscriber obtain the authentication result "1" or "0".

Protocol:

1 The user scans his biometric image Bio_U with the provided tamper-proof device. The device then runs $BR.Ext(Bio_U) \rightarrow \vec{y}_{ID}$ to extract the feature vector and runs $Enc(PM, sk_{ID}, i, \vec{y}_{ID}) \rightarrow \vec{c}_{y_{ID}}$ to generate encrypted biometric features. Additionally, the device splits the encrypted biometric features into two parts, where $\{\vec{c0}_{y_{ID}} || \vec{c1}_{y_{ID}}\} = \vec{c}_{y_{ID}}$. The device re-encrypts $\vec{c}_{y0_{ID}}$ by running $ReEnc(PM, sk_{ID}, i, \vec{c}_{y0_{ID}}) \rightarrow \vec{c'}_{ID,i}$.
2 The device then runs the verification protocol with the data subscriber to compute the first partial distance d_0 where the device has input $\vec{c}'_{y_{ID,i}}$ and the data subscriber has input $(\vec{c}'_{x_{ID,i}}$. The device verifies the first part of authentication and proceeds if and only if $d_0 \leq t^2_{ID,i}$ where $t_{ID,i} = EncT(PM, sk_{ID}, i, t)$. Otherwise, the process stops and the device returns the authentication result as "0" to the data subscriber.
3 The second partial distance d_1 is run with the cloud where the device has input $\vec{c}_{y1_{ID}}$ and the cloud has input $\vec{c}_{x1ID}$. The device obtains $d_1 = (T_1 - R_1) \cdot w^2_{ID,i}$ where $w_{ID,i} = PRF_W(sk_{ID}, i)$ and checks if $(d_0 - t^2_{ID,i}) + d_1 \leq 0$, which also indicates $d \leq t^2_{ID}$.
4 Finally, the device replies the authentication result "1" or "0" to the data subscriber.

Figure 4: Authentication phase

Authentication Phase

In the authentication phase, the user submits and encrypts the biometric feature with the proposed encryption scheme, and transmits it to the subscriber and the cloud. PBio utilizes the components described in 3.1 to perform the authentication in a secure manner. At the end of the protocol, the user device obtains a one-bit authentication result. The details of authentication protocol is described in Figure 4.

4 SECURITY OF PBIO

4.1 Security Models

We follow classical security formulation [17] for an authentication scheme, which includes *correctness*, *soundness* and optionally, *zero-knowledge*. Here we will not repeat these well-known definitions. We remark that, the *correctness* definition of biometric authentication is slightly different from transitional definition (e.g. [17]), since a legitimate user might be rejected with a small probability—that's the definition of *false rejection rate* or *false negative rate*, due to the noise nature in measurements of biometric feature. In the real world scenarios, the user may re-try after some adjustment (e.g. clean the finger).

In this work, every secret key used to encrypt the biometric templates for every user is derived from a master secret key owned by the data owner. The encrypted biometric templates are stored by the cloud or the data subscriber. We allow collusion between the cloud and the data subscriber. The goal of the adversary is to masquerade a victim user and be accepted by the authentication solution under the victim's name (i.e. breaking soundness property), or to lean some secret information of victim's raw biometric feature via our authentication system (i.e. breaking the zero-knowledge property.

We emphasize that an authentication scheme will suffer from online brute-force attack, since it always leaks at least 1 bit information—accepting or rejecting a user, even if a matching scheme contains some cryptography primitive (e.g. [76]), which is semantic secure. In other words, semantic secure building blocks in authentication scheme may be an overkill. Indeed, some of our building block (i.e. distance preserving encryption) is not semantic secure.

4.2 Security Analysis

The proposed PBio schemes apply the distance-preserving transformation (DPT) scheme in Section 3.1.2 and secure distance computation protocol (GSHADE) in Section 3.1.3. Hence, its security depends on the security of these underlying schemes.

Besides, we also provide the insight of security analysis to different forms of splitting, such as splitting arrangement is public information, splitting arrangement is secret, and splitting arrangement is secret and dummy dimensions are added to raw biometric template templates.

Proposition 4.1 (Correctness). *Our proposed authentication solution is correct, i.e. any legitimate user who is following our authentication solution exactly, will be accepted, except a small probability (i.e. the false negative rate of biometric feature).*

The above proposition follows directly from the property of distance preserving encryption and correctness of GSHADE.

Theorem 4.2 (Zero Knowledge Proof). *After interacting with a user Alice by executing our authentication solution for many times, both the cloud and subscriber learn nothing about Alice's biometric raw data, beyond the ciphertext.*

Sketch proof of Theorem 4.2. The two party secure computation GSHADE does not leak useful information to Cloud/Subscriber. This security guarantee is derived from the privacy of the underlying oblivious transfer protocol as there is no other message being exchanged during the protocol. The proof follows from Theorem 3.3. □

Theorem 4.3 (Soundness). *Probabilistic polynomial time adversary (even colluded with some subscriber and cloud), cannot pass our authentication with non-negligible probability.*

Sketch proof of Theorem 4.3. First of all, we remark that, the authentication client software in user's device is trusted (e.g. ARM TrustZone enabled program), and is verified by the authentication server every time, before user starts to authenticate to the server. Thus, our official authentication client software is the only way to authenticate to the server, and third party authentication client software can be easily detected and rejected by authentication server.

The adversary may collude with both Cloud and subscriber, and thus is able to find the DPT ciphertext `ct` of user's bio template vector $\vec{x}$, and observe any network communications of GSHADE.

Note that in our invocation of GSHADE protocol between bio-device and cloud (or subscriber), the authentication server learns only one bit information—accepting or rejecting this user.

Furthermore, due to Theorem 3.2, a single ciphertext `ct` does not leak any information of plaintext, i.e. the user's bio template vector $\vec{x}$.

Consequently, the adversary is unable to find an estimation $\vec{x}'$ such that $Dist(\vec{x}, \vec{x}')$ is smaller than the given threshold, and thus cannot pass our authentication scheme. □

5 IMPLEMENTATION AND EVALUATION

5.1 Prototype Implementation

For proof of concept, we implemented P2Bio prototype with four machines to represent the data owner, the organisation, the cloud provider, and the user device respectively. The four machines are with the same hardware specification, which is Intel Core i7-8700 CPU @3.20GHz with 8GB RAM and two cores. We then applied a face recognition python library[2] as the biometrics recognition scheme, which enables us to detects a face in a raw image, extracts the feature vectors, and matches the similarity later. We also numpy library[3] to generate vectors and matrices and perform mathematics operation.

During the setup phase, a master secret key *msk* was randomly selected in a 256 bits domain. We then extracted a face template database with the respective user identity based on the collected raw images from our colleagues. The template database was then encrypted following the proposed encryption scheme. Note that the data owner will generate a set of encrypted database $\vec{c_x}$, which is then split into two parts $\{c\vec{0}_x \mid\mid c\vec{1}_x\}$, where $c\vec{1}_x$ is stored by the cloud. We then generated the subscriber i encrypted template $\vec{c}'_{ID,i}$ and passed it to the subscriber. We also forwarded each user secret key sk_{ID} to the respectively user device.

During the authentication phase, we supposed that a user would like to prove himself to an organisation. The user took his registered device, e.g. a laptop or a smartphone, to capture his face. The device then run the face recognition scheme to generate the feature vectors

[2] https://pypi.org/project/face-recognition/
[3] https://pypi.org/project/numpy/

Table 1: Fist Layer Encryption Performance

No. of User	Encryption Time			
	64-n	128-n	320-n	640-n
1	0.61 ms	1.14 ms	4.49 ms	20.93 ms
1,000	0.6 s	1.17 s	4.51 s	20.74 s
10,000	5.93 s	11.73 s	44.82 s	3 m 29 s
100,000	58.83 s	1 m 58 s	7m 35s	34 m 45 s
1,000,000	9 m 49 s	19 m 36 s	1 h 15 m	5 h 47 m
2,000,000	19 m 25 s	39 m 8 s	2 h 31 m	11 h 33 m
5,000,000	49 m 7 s	1 h 35 m	6h 17 m	28h 52 m
10,000,000	1h 37m	3 h 4 m	12h 30 m	57h 44 m

Table 2: Encryption Performance Per User in PBio

Dimension	64-n	128-n	320-n	640-n
First Layer Encryption Time	0.58 ms	1.14 ms	4.52 ms	20.79 ms
Re-encryption Time	0.24 ms	0.60 ms	1.43 ms	4.53 ms
Total Encryption Time	0.82 ms	1.74 ms	5.95 ms	25.32 ms

$\vec{y}$ and run the proposed encryption scheme to generate $\vec{cy}_{ID,i}$. Lastly, a secure matching protocol GSHADE was run between the device and the cloud where the device on input $\vec{cy}_{ID,i}$ and the cloud on input $\vec{cx}_{ID,i}$. The output of the protocol indicates that the similarity of the two encrypted ciphertexts are smaller than or equal to the encrypted threshold $t_{ID,i}$ such that $Dist(\vec{cx}_{ID,i}, \vec{cy}_{ID,i}) \leq t^2_{ID,i}$. The cloud then forwards "1" or "0" to the organisation to show the authentication is accepted or rejected. The organization followed the result.

5.2 Evaluation

For performance evaluation purpose, we follow the similar approach, which has been commonly applied in evaluating the performance of biometric encryption in [14, 28, 83]. Firstly, a set of random vectors was generated to represent the original biometric template database because one can apply any biometric recognition schemes to extract the feature vectors in practice, hence we do not consider the time required for feature extraction in the experiment. For the remainder of this section, we denote 64-n, 128-n, 320-n, 640-n to represent dimensions of 64, 128, 320 and 640 respectively.

We randomly generated $m \times n$ vectors. This means there are m number of user in the database with n dimension of biometric feature vector. The experimental results in Table 1 shows the encryption time required for $m \times n$ biometric template database.

Note that the encryption time is for the first layer encryption only. We require additional encryption for every subscriber in half of the dimension. For example, if we apply face recognition scheme that consists of 128-n dimension for a template, the encryption time took approximate 1.14 ms per user. In Pbio, we split n into half after the first layer encryption and we re-encrypt the second layer encryption in 64-n dimension. Hence, there is an additional 0.61 ms requires for every user, which indicates that PBio requires 1.74 ms encryption time. We summarized the encryption time per user in Table 2. We notice that the encryption time increases with the dimensional size of a template.

Table 3: Size of Biometric Templates

No. of User	Size of Database			
	64-n	128-n	320-n	640-n
1	512 B	1024 B	2560 B	5120 B
1,000	512 KB	1.024 MB	2.56 MB	5.12 MB
10,000	5.12 MB	10.24 MB	25.6 MB	51.2 MB
100,000	51.2 MB	1024 MB	256 MB	512 MB
1,000,000	512 MB	1.02 GB	2.56 GB	5.12 GB
2,000,000	1.024 GB	2.04 GB	5.12 GB	10.24 GB
5,000,000	2.56 GB	5.1 GB	12.8 GB	25.6 GB
10,000,000	5.12 GB	10.2 GB	25.6 GB	51.2 GB

Table 4: PBio: Verification Performance. Recall that ℓ denotes the network latency.

Verification Time	
Part I	Part II
2.21 ms + ℓ	1.004 ms + ℓ
Total Time	
3.214 ms + 2ℓ	

Table 3 summarised the various sizes of biometric templates. The size of the original database and the encrypted database are the same because our encryption technique transforms an original value into a random value, e.g. a biometric template in 128-n dimension and its encrypted template are both in 1024 bytes (B).

We then analyse the verification time of PBio in Table 4. In the experiment, the fresh submitted biometric features with 128-n was first encrypted into a subscriber encrypted features, which is two layers encryption in 128-n and 64-n, and then we applied GSHADE for the secure distance computation. PBio took approximate 3.26 ms + 2ℓ in total for the verification where ℓ is the network latency for GSHADE. We noticed that ℓ is very depended by the network itself. In our environment, we first tested in our local machine which achieved the result in Table 4. We then connected the machines over the internet and estimated that ℓ is 59ms in our internet environment.

One of the merits with the partial verification is to achieve early rejection. For example, in the situation that Part I is rejected at the first stage, the process can be terminated without proceeding to Part II. This reduce cost of the communication and network latency ℓ. A LFW dataset with the test case in http://vis-www.cs.umass.edu/lfw/pairsDevTest.txt was extracted, which consists of 409 pairs are true positive and 444 pairs are true negative. In the plain manner, we first set a threshold t as 0.6 and the matched result is 402 out of 409, which shows 98.29% of them is true positive, and non-matched result is 442 out of 444, which shows 99.55% is true negative. In PBio, we also had the same accuracy in both the true positive and true negative results, and the early rejection is 190 out of 442, which is 42.99% in the non-matched results.

5.3 Comparison

As shown in Table 5, we compare the encryption performance for 128-n dimension among the previous works [14, 28, 83]. Note that

Table 5: Comparison of Encryption Performance

No. of User	Encryption Time for 128-n			
	[14]	[28]	[83]	PBio
1	1.314 s	0.88 ms	0.73 ms	1.74 ms
100,000	36 h 26m	1 m 27 s	1 m 12 s	2 m 58 s
1,000,000	364 h 25m	14 m 40 s	12 m 9 s	29 m 13 s

Table 6: Comparison of Encrypted Database Size

Size of Database	Size of Encrypted Database			
	[14]	[28]	[83]	PBio
1 KB	32.77 KB	270.4 KB	141.51 KB	1 KB
1.02 GB	32.77 GB	269.34 GB	140.96 GB	1.02 GB
5.1 GB	163.84 GB	1346.72 GB	707.56 GB	5.1 GB

Table 7: Comparison of Verification Performance

Schemes	[14]	[28]	[83]	PBio
Verification per user	5.84 s	1.78 ms	0.73 ms	3.21 ms

[14] was implemented using the Paillier encryption scheme [56] with 1024-bit modulus, and [28, 83] are the similar technique as ours. We notice that our scheme is 0.42× slower than [83], but the ciphertext size is much smaller than them, which preserves the same database size as in the original database. Table 6 shows the comparison of the encrypted database size. The verification for each scheme is summarized in Table 7.

6 DISCUSSION

We adopt a number of additional measures in our biometric system to further enhance its security. Firstly upon registration, the data owner assigns a unique key to each end-user. This ensures that the resulting encryption applied to each raw biometric template will be distinct for different end-users. Secondly, our biometric system is enabled to refresh the partial encrypted database held by the data subscriber and the cloud provider either periodically or when is it necessary. For instance, in the event a user's device is lost and requires a replacement. For the remainder of this section, we provide a comparison between two feasible mechanisms for template encryption as well as a detailed discussion on the key update process.

6.1 Encrypt-then-split vs Split-then-encrypt

We further consider two different approaches to perform encryption of the raw biometric template: encrypt then split and split then encrypt as shown in Table 8. The encrypt then split approach first encrypts the raw biometric template and split them into two. For the latter approach, the raw biometric template is first split and each individual split portion is subsequently encrypted. Recall that PBio first checks the Part I and proceed the rest if and only if Part I is successfully passed. Since split-then-encrypt approach performs encryption in half of the n-dimension, we see that split-then-encrypt approach achieves faster early rejection as the encryption time needed for Part I and II can be done separately. However, the encrypt-then-split approach results in an overall faster verification time. As such, we decide to adopt the encrypt-then-split mechanism in our experiments of PBio.

Table 8: Comparison of Verification Performance under Encrypt-then-split and Split-then-encrypt. Recall that ℓ denotes the network latency.

	Verification Time	
	Encrypt-then-Split	Split-then-Encrypt
Part I	2.21 ms + ℓ	1.63 ms + ℓ
Part II	1.004 ms + ℓ	1.63 ms + ℓ
Total Time	3.214 ms+ 2ℓ	3.26 ms+ 2ℓ

6.2 Key Update

Key update is an essential procedure in a biometric system. Periodic key updates of existing database help to safeguard against potential keys leakage or exposure. In addition, should a group of users' keys be compromised, a timely key update process ensure that their biometric templates are still protected. We discuss several methods which enables efficient key updates.

Suppose a raw biometric template x_i corresponds to user U_i, with initial key $k_{i,0}$. Denote $E^{(1)}(x)$ and $E^{(2)}(x)$ to be disjoint halves of encryption $E(x)$. Let the initial encrypted template of U_i held by the cloud and the data subscriber to be $Y_i = E^{(1)}_{k_{i,0}}(x_i)$ and $Z_i = E^{(2)}_{k_{i,0}}(x_i)$ respectively. When the key update for user U_i is initialized, the device receives Y_i, Z_i from the cloud and data subscriber respectively. The device updates the encrypted template of user U_i by performing $E_{k_{i,1}}(Y_i)$ and $E_{k_{i,1}}(Z_i)$ which are subsequently transmitted to the cloud and data subscriber respectively. Consequently, $E_{k_{i,1}}(Y_i)$ is the encrypted template of U_i with the updated key held by the cloud while $E_{k_{i,1}}(Z_i)$ is the encrypted template of U_i with the updated key held by the data subscriber. A potential limitation of this method is that the device is required to fetch encrypted templates of the associated users from the data subscriber and cloud whenever a key update process is called upon.

One other feasible way is for the data owner to be involved in the key update process. In this way, whenever a key update process is called upon for a group of users U_i, the data owner can simply generate new keys and send the corresponding new encrypted templates of U_i to the cloud and data subscribers. The device will also be notified of the generation and values of these new keys. However, this requires the data owner to be online during every key update process.

To overcome the above issues and limitations, we introduce a trusted key management server to be involved in the key update process. This key server which can be continuously online is hosted by the data owner. The main role of this key server is to issue new keys whenever a key update process is initiated. When the key update for user U_i is initialized, the key server fetches $Y_i = E^{(1)}_{k_{i,0}}(x_i)$, $Z_i = E^{(2)}_{k_{i,0}}(x_i)$ from the cloud and data subscriber respectively. The key server decrypts these encrypted templates to obtain x_i. New keys are generated to perform a re-encryption of x_i. The new keys

are sent to the trusted device. Additionally, encrypted templates $E^{(1)}_{k_{i,1}}(x_i)$ and $E^{(2)}_{k_{i,1}}(x_i)$ are sent to the cloud and data subscriber respectively which represent the updated encrypted templates.

7 RELATED WORK

In this section we describe relevant works, focusing on techniques to protect biometric templates and existing privacy-preserving biometric authentication schemes.

7.1 Security of Biometric Templates

There are numerous methods enabling the reconstruction of biometric images from certain types of raw biometric templates. In general, the security considerations relating to biometric templates can be classified into two main types: template inversion and hill climbing. In template inversion methods, an adversary is assumed to have a copy of a biometric template and attempts to reconstruct it back to its original biometric image. In the case of hill climbing approaches, an adversary is not required to have a copy of a biometric template. Instead, a collection (one or more) of synthetic biometric templates are generated and sent to the matcher. The synthetic template is then modified according to the numerical result of the matcher. This process is carried out iteratively until a match is attained. Table 9 partially taken from [50] highlights some of the known techniques involving template inversion and hill climbing approaches.

As shown in Table 9, there exist numerous known methods which demonstrate the insecurity relating to certain types of raw biometric templates, in particular those of Minutiae and Iriscode. A comprehensive survey on recent advances in inverse biometrics is presented in [25].

This highlights that storing a biometric template in the clear is not sufficient to protect its underlying biometric information. Our proposed solution addresses this issue by employing a lightweight encryption on the raw biometric templates.

7.2 Securing Biometric Templates

According to [26], there are two generic solutions to protect biometrics, namely based on image processing [38, 54, 68] and based on cryptographic techniques [15, 26, 58, 61, 76]. Image-based techniques are computationally efficient but as stated in [77], most of the image processing techniques result in decreasing accuracy due to the distortion applied on the original image. Solutions based on cryptographic techniques commonly known as secure computation, on the other hand, preserve accuracy comparable to that of the ordinary recognition schemes but incur higher performance overhead. Secure computation can be achieved by applying various cryptography tools such as Homomorphic Encryption (HE) [49, 56, 65], Predicate Encryption (PE) [35, 64, 83], Inner Product Encryption (IPE) [2, 18, 36], Oblivious Transfer (OT) [29, 53], and Garbled Circuit Evaluation (GCE) [80]. These tools allow the verification to be done securely via calculating the distance between the enrolled and probed biometric templates in the encrypted domain [5, 6, 74], but it is also known that most of the cryptography-based techniques are computation intensive. Barni et al. [5] applied Paillier's HE [56] which allows user to verify whether the submitted biometric feature is in the server database.

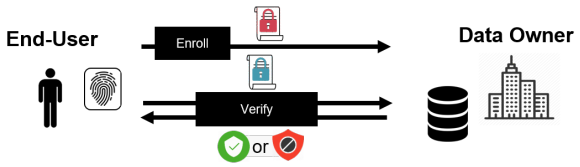

Figure 5: Privacy-preserving Biometrics System Overview [63]

7.3 Biohashing and Fuzzy Extractor

Biohashing [9, 34] were proposed to generate a unique value based on similar biometric features captured from the same person. The unique value is stored as a representation of the biometric feature of an individual. During authentication, a value is generated based on the freshly captured biometrics. The generated value can be used to perform exact match, just like password matching, with the unique value stored. Another related work is cancellable biometrics [4, 30, 59], which applies similar techniques to biohashing. The main purpose of cancellable biometrics is to allow revocation if the value stored is compromised. There are, however, successful attacks on cancellable biometrics and biohashing [13, 37, 40, 42].

Fuzzy commitment and secure sketch ensure the privacy of biometrics by providing information-theoretic guarantees using error correcting codes [19, 32, 33, 67, 79] or signal embeddings [16, 51, 52, 66, 72]. Chun et al. [15] proposed a fuzzy extraction technique to create a metadata during enrollment that can be stored on a user device, and a secret token is recovered from the metadata to complete the authentication. However, these techniques face security issues when it is used multiple times and it assumes certain conditions on the distribution of biometrics, as stated in [33]. Chatterjee et al. [12] proposed a scheme that protect biometric templates based on secure sketches. It prevents an attacker from learning the owner of biometric templates by collecting and randomly permuting multiple fingerprints of the users.

7.4 Privacy-Preserving Biometric Authentication

In general, most of the existing privacy-preserving biometric authentication (and identification) systems can be categorised into two settings. The first is a direct user-to-organisation setting as can be seen from the system proposed by Šeděnka et al. [63]. Their system allows every user, with a trusted device that holds a secret key, to enroll his encrypted biometric templates using homomorphic encryption (HE). During authentication, garbled circuit evaluation (GCE) is applied to decrypt and compare the similarity among the encrypted template and the submitted feature. Figure 5 illustrate the setting deployed by Šeděnka et al. [63]. It represents a common setting whereby the organisation (data owner) hosts the (encrypted) biometric templates. Similar to [63], the biometrics systems by Zhou and Ren [83], PassBio, and Lee et al. [41] also requires a trusted device that encrypts the biometric features during enrollment and authentication by using predictable encryption (PE) and inner production encryption (IPE) respectively. The merit of [41, 83] is that decryption is not required during authentication as PE and IPE allows one to find the similarity given two encrypted templates.

Table 9: Techniques of Biometric Features Reconstruction.

Method	Feature	Template	Reconstruction	Type
Potzsch et al. [57]	Face	Elastic bunch graph	Face image	Template inversion
Hill [27]	Fingerprint	Minutiae	Fingerprint image	Template inversion
Ross et al. [60]	Fingerprint	Minutiae	Fingerprint image	Template inversion
Cappelli et al. [11]	Fingerprint	Minutiae	Fingerprint image	Template inversion
Testoni and Kirovski [69]	Iris	Iriscode	Iris	Template inversion
Feng and Jain [21]	Fingerprint	Minutiae	Fingerprint image	Template inversion
Li and Kot [43]	Fingerprint	Minutiae	Fingerprint image	Template inversion
Galbally et al. [23]	Iris	Iriscode	Iris	Template inversion
Cao and Jain [10]	Fingerprint	Minutiae	Fingerprint image	Template inversion
Mai et al. [45]	Face	FaceNet [62]	Face image	Template inversion
Adler [3]	Face	–	Face image	Hill climbing
Uludag and Jain [71]	Fingerprint	–	Minutiae	Hill climbing
Yamazaki et al. [78]	Signature	–	Time series data	Hill climbing
Mohanty et al. [47]	Face	–	Face image	Hill climbing
Muramatsu et al. [48]	Signature	–	Time series data	Hill climbing
Galbally et al. [22]	Face	–	Face image	Hill climbing
Martinez-Diaz et al. [46]	Fingerprint	–	Minutiae	Hill climbing

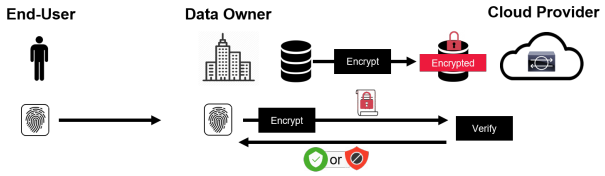

Figure 6: Outsourced Privacy-preserving Biometrics Systems Overview

The second is an outsourced setting where the data owner securely outsources the authentication (and identification) processes to a cloud [31, 39] in order to reduce the computation and operational costs. Figure 6 illustrates the setting. Chun et al. [14] proposed a system that uses HE to encrypt the biometric templates. The encrypted templates and secret key are distributed to two clouds respectively with the assumption that no collusion happens. During authentication, the fresh encrypted biometric features are submitted to the two clouds to find the similarity using HE and GCE so that the cloud can neither learn the plain biometric templates nor the features. Xiang et al. [76] proposed a scheme which relies on a hybrid encryption scheme, which is a variant of fully homomorphic encryption scheme. However, it results in higher communication overheads as compared to the previous schemes. Tian et al. [70] introduced remote user authentication that is similar to [41, 63, 83] where user biometric templates are stored in the encrypted format and the authentication is done in an anonymous and unlinkable manner with a cloud. A matrix based technique was proposed by Yuan and Yu [81] that allows the cloud to verify an individual without decryption. However, [81] was found to be insecure and enhanced by [73], CloudBI. Further enhancement was proposed in [28], which also stated that such the matrix based technique is weakly secure. Recently, Guo et al. [26] applied randomization technique which allows feature extraction and authentication to be done in the encrypted domain. However, it supports only face recognition and the security is not guaranteed as only completeness analysis was provided.

8 CONCLUSIONS

We proposed a new privacy-preserving biometric authentication system, PBio, which allows the data owner to outsource a biometric database to authenticate users in a privacy-preserving manner. The proposed system supports any biometric recognition schemes that is based on euclidean distance. The accuracy is preserved even in the encrypted domain. Besides, we introduced a split-then-encrypt construction, which allows the data owner to split the encrypted biometric templates into two or more copies. One copy is then given to a cloud and the other copies to other subscribed organisation. With this property, the proposed system allows user to authenticate a partial encrypted biometric feature with the partial encrypted biometric template stored by the organisation, which computes a partial result that allows the organisation to make early rejection. In the case if the final result is necessary, the cloud provider is required to involve in ascertaining the final result. Assuming data breach happens in either the cloud or the organisation, the attacker will only obtain partial information of the templates. We developed a prototype for the proposed system. The experiment shows that our proposed system is practical.

REFERENCES

[1] 2002. *International Biometric Group.: Generating Images from Templates.* White paper.

[2] Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. 2015. Simple functional encryption schemes for inner products. In *IACR International Workshop on Public Key Cryptography*. Springer, 733–751.

[3] Andy Adler. 2003. Sample images can be independently restored from face recognition templates. In *CCECE 2003-Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology (Cat. No. 03CH37436), vol. 2*. IEEE, 1163–1166.

[4] Russell Ang, Rei Safavi-Naini, and Luke McAven. 2005. Cancelable key-based fingerprint templates. In *Australasian conference on information security and privacy*. Springer, 242–252.

[5] Mauro Barni, Tiziano Bianchi, Dario Catalano, Mario Di Raimondo, Ruggero Donida Labati, Pierluigi Failla, Dario Fiore, Riccardo Lazzeretti, Vincenzo Piuri, Fabio Scotti, et al. 2010. Privacy-preserving fingercode authentication. In *Proceedings of the 12th ACM workshop on Multimedia and security*. ACM, 231–240.

[6] Marina Blanton and Paolo Gasti. 2011. Secure and efficient protocols for iris and fingerprint identification. In *European Symposium on Research in Computer Security*. Springer, 190–209.

[7] Julien Bringer, Herve Chabanne, Melanie Favre, Alain Patey, Thomas Schneider, and Michael Zohner. 2014. GSHADE: faster privacy-preserving distance computation and biometric identification. In *Proceedings of the 2nd ACM workshop on Information hiding and multimedia security*. 187–198.

[8] Julien Bringer, Hervé Chabanne, and Alain Patey. 2013. Shade: Secure hamming distance computation from oblivious transfer. In *International Conference on*

Financial Cryptography and Data Security. Springer, 164–176.

[9] Ran Canetti, Benjamin Fuller, Omer Paneth, Leonid Reyzin, and Adam Smith. 2016. Reusable fuzzy extractors for low-entropy distributions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 117–146.

[10] Kai Cao and Anil K. Jain. 2014. Learning fingerprint reconstruction: From minutiae to image. *IEEE Transactions on information forensics and security* 10, 1 (2014), 104–117.

[11] Raffaele Cappelli, Dario Maio, Alessandra Lumini, and Davide Maltoni. 2007. Fingerprint image reconstruction from standard templates. *IEEE transactions on pattern analysis and machine intelligence* 29, 9 (2007), 1489–1503.

[12] Rahul Chatterjee, M Sadegh Riazi, Tanmoy Chowdhury, Emanuela Marasco, Farinaz Koushanfar, and Ari Juels. 2019. Multisketches: Practical Secure Sketches Using Off-the-Shelf Biometric Matching Algorithms. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1171–1186.

[13] King Hong Cheung, Adams Wai-Kin Kong, Jane You, and David Zhang. 2005. An Analysis on Invertibility of Cancelable Biometrics based on BioHashing.. In *CISST*, Vol. 2005. 40–45.

[14] Hu Chun, Yousef Elmehdwi, Feng Li, Prabir Bhattacharya, and Wei Jiang. 2014. Outsourceable Two-Party Privacy-Preserving Biometric Authentication. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (ASIA CCS '14)*. Association for Computing Machinery, New York, NY, USA, 401–412. https://doi.org/10.1145/2590296.2590343

[15] Park Ho Chung, Wenke Lee, Erkam Uzun, and Carter Yagemann. Privacy preserving face-based authentication. (????).

[16] T Charles Clancy, Negar Kiyavash, and Dennis J Lin. 2003. Secure smartcardbased fingerprint authentication. In *Proceedings of the 2003 ACM SIGMM workshop on Biometrics methods and applications*. ACM, 45–52.

[17] Nicolas T. Courtois. 2001. Efficient Zero-Knowledge Authentication Based on a Linear Algebra Problem MinRank. In *ASIACRYPT*. 402–421.

[18] Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay. 2016. Functional encryption for inner product with full function privacy. In *Public-Key Cryptography–PKC 2016*. Springer, 164–195.

[19] Stark C Draper, Ashish Khisti, Emin Martinian, Anthony Vetro, and Jonathan S Yedidia. 2007. Using distributed source coding to secure fingerprint biometrics. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, Vol. 2. IEEE, II–129.

[20] David Evans, Yan Huang, Jonathan Katz, and Lior Malka. 2011. Efficient privacy-preserving biometric identification. In *Proceedings of the 17th conference Network and Distributed System Security Symposium, NDSS*, Vol. 68.

[21] Jianjiang Feng and Anil K. Jain. 2011. Fingerprint reconstruction: from minutiae to phase. *IEEE transactions on pattern analysis and machine intelligence* 33, 2 (2011), 209–223.

[22] Javier Galbally, Chris McCool, Julian Fierrez, Sebastien Marcel, and Javier Ortega-Garcia. 2010. On the vulnerability of face verification systems to hill-climbing attacks. *Pattern Recognition* 43, 3 (2010), 1027–1038.

[23] Javier Galbally, Arun Ross, Marta Gomez-Barrero, Julian Fierrez, and Javier Ortega-Garcia. 2013. Iris image reconstruction from binary templates: An efficient probabilistic approach based on genetic algorithms. *Computer Vision and Image Understanding* 117, 10 (2013), 1512–1525.

[24] Shafi Goldwasser. 1987. How to play any mental game, or a completeness theorem for protocols with an honest majority. *Proc. the Nineteenth Annual ACM STOC'87* (1987), 218–229.

[25] Marta Gomez-Barrero and Javier Galbally. 2020. Reversing the irreversible: A survey on inverse biometrics. *Computers & Security* 90 (2020).

[26] Shangwei Guo, Tao Xiang, and Xiaoguo Li. 2019. Towards Efficient Privacy-Preserving Face Recognition in the Cloud. *Signal Processing* (2019).

[27] Christopher James Hill. 2001. *Risk of masquerade arising from the storage of biometrics*. Bachelor of Science thesis, The Department of Computer Science, Australian National University.

[28] Shengshan Hu, Minghui Li, Qian Wang, Sherman SM Chow, and Minxin Du. 2018. Outsourced biometric identification with privacy. *IEEE Transactions on Information Forensics and Security* 13, 10 (2018), 2448–2463.

[29] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. 2003. Extending oblivious transfers efficiently. In *Annual International Cryptology Conference*. Springer, 145–161.

[30] Andrew Teoh Beng Jin, David Ngo Chek Ling, and Alwyn Goh. 2004. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. *Pattern recognition* 37, 11 (2004), 2245–2255.

[31] Anthony D JoSEP, RAnDy KAtz, AnDy KonWinSKi, LEE Gunho, DAViD PAttERSon, and ARiEL RABKin. 2010. A view of cloud computing. *Commun. ACM* 53, 4 (2010).

[32] Ari Juels and Madhu Sudan. 2006. A fuzzy vault scheme. *Designs, Codes and Cryptography* 38, 2 (2006), 237–257.

[33] Ari Juels and Martin Wattenberg. 1999. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security*. ACM, 28–36.

[34] Sanjay Kanade, Dijana Petrovska-Delacrétaz, and Bernadette Dorizzi. 2009. Cancelable iris biometrics and using error correcting codes to reduce variability in biometric data. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 120–127.

[35] Jonathan Katz, Amit Sahai, and Brent Waters. 2008. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *annual international conference on the theory and applications of cryptographic techniques*. Springer, 146–162.

[36] Sam Kim, Kevin Lewi, Avradip Mandal, Hart Montgomery, Arnab Roy, and David J Wu. 2018. Function-hiding inner product encryption is practical. In *International Conference on Security and Cryptography for Networks*. Springer, 544–562.

[37] Adams Kong, King-Hong Cheung, David Zhang, Mohamed Kamel, and Jane You. 2006. An analysis of BioHashing and its variants. *Pattern recognition* 39, 7 (2006), 1359–1368.

[38] Pavel Korshunov and Touradj Ebrahimi. 2013. Using face morphing to protect privacy. In *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, 208–213.

[39] Karthik Kumar and Yung-Hsiang Lu. 2010. Cloud computing for mobile users: Can offloading computation save energy? *Computer* 4 (2010), 51–56.

[40] Patrick Lacharme, Estelle Cherrier, and Christophe Rosenberger. 2013. Preimage attack on biohashing. In *2013 International Conference on Security and Cryptography (SECRYPT)*. IEEE, 1–8.

[41] Joohee Lee, Dongwoo Kim, Duhyeong Kim, Yongsoo Song, Junbum Shin, and Jung Hee Cheon. 2018. Instant Privacy-Preserving Biometric Authentication for Hamming Distance. *IACR Cryptology ePrint Archive* 2018 (2018), 1214.

[42] Yongjin Lee, Yunsu Chung, and Kiyoung Moon. 2009. Inverse operation and preimage attack on biohashing. In *2009 IEEE Workshop on Computational Intelligence in Biometrics: Theory, Algorithms, and Applications*. IEEE, 92–97.

[43] Sheng Li and Alex C. Kot. 2012. An improved scheme for full fingerprint reconstruction. *IEEE Transactions on information forensics and security* 7, 6 (2012), 1906–1912.

[44] Kun Liu, Chris Giannella, and Hillol Kargupta. 2006. An attacker's view of distance preserving maps for privacy preserving data mining. In *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 297–308.

[45] Guangcan Mai, Kai Cao, Pong C. Yuen, and Anil K. Jain. 2018. On the reconstruction of face images from deep face templates. *IEEE transactions on pattern analysis and machine intelligence* 41, 5 (2018), 1188–1202.

[46] Marcos Martinez-Diaz, Julian Fierrez, Javier Galbally, and Javier Ortega-Garcia. 2011. An evaluation of indirect attacks and countermeasures in fingerprint verification systems. *Pattern Recognition Letters* 32, 12 (2011), 1643–1651.

[47] Pranab Mohanty, Sudeep Sarkar, and Rangachar Kasturi. 2007. From scores to face templates: a model-based approach. *IEEE transactions on pattern analysis and machine intelligence* 29, 12 (2007), 2065–2078.

[48] Daigo Muramatsu. 2008. Online signature verification algorithm using hill-climbing method. In *2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, vol. 2*. IEEE, 133–138.

[49] Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. 2011. Can homomorphic encryption be practical?. In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*. ACM, 113–124.

[50] Abhishek Nagar. 2012. *Biometric template security*. Michigan State University. Computer Science.

[51] Abhishek Nagar, Karthik Nandakumar, and Anil K Jain. 2008. Securing fingerprint template: Fuzzy vault with minutiae descriptors. In *2008 19th International Conference on Pattern Recognition*. IEEE, 1–4.

[52] Karthik Nandakumar, Anil K Jain, and Sharath Pankanti. 2007. Fingerprint-based fuzzy vault: Implementation and performance. *IEEE transactions on information forensics and security* 2, 4 (2007), 744–757.

[53] Moni Naor, Benny Pinkas, and Benny Pinkas. 2001. Efficient oblivious transfer protocols. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 448–457.

[54] Elaine M Newton, Latanya Sweeney, and Bradley Malin. 2005. Preserving privacy by de-identifying face images. *IEEE transactions on Knowledge and Data Engineering* 17, 2 (2005), 232–243.

[55] Stanley RM Oliveira and Osmar R Zaiane. 2010. Privacy preserving clustering by data transformation. *Journal of Information and Data Management* 1, 1 (2010), 37–37.

[56] Pascal Paillier. 1999. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*. Springer, 223–238.

[57] Michael Pötzsch, Thomas Maurer, Laurenz Wiskott, and C. von-der Malsburg. 1996. Reconstruction from graphs labeled with responses of Gabor filters. In *International Conference on Artificial Neural Networks*. Springer, 845–850.

[58] Zhan Qin, Jingbo Yan, Kui Ren, Chang Wen Chen, and Cong Wang. 2014. Towards efficient privacy-preserving image feature extraction in cloud computing. In *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 497–506.

[59] Christian Rathgeb and Andreas Uhl. 2010. Iris-biometric hash generation for biometric database indexing. In *2010 20th International Conference on Pattern Recognition*. IEEE, 2848–2851.

[60] Arun Ross, Jidnya Shah, and Anil K. Jain. 2007. From template to image: Reconstructing fingerprints from minutiae points. *IEEE transactions on pattern analysis and machine intelligence* 29, 4 (2007), 544–560.

[61] Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg. 2009. Efficient privacy-preserving face recognition. In *International Conference on Information Security and Cryptology*. Springer, 229–244.

[62] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.

[63] Jaroslav Šeděnka, Sathya Govindarajan, Paolo Gasti, and Kiran S Balagani. 2014. Secure outsourced biometric authentication with performance evaluation on smartphones. *IEEE Transactions on Information Forensics and Security* 10, 2 (2014), 384–396.

[64] Emily Shen, Elaine Shi, and Brent Waters. 2009. Predicate privacy in encryption systems. In *Theory of Cryptography Conference*. Springer, 457–473.

[65] Nigel P Smart and Frederik Vercauteren. 2010. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *International Workshop on Public Key Cryptography*. Springer, 420–443.

[66] Yagiz Sutcu, Qiming Li, and Nasir Memon. 2007. Protecting biometric templates with sketch: Theory and practice. *IEEE Transactions on Information Forensics and Security* 2, 3 (2007), 503–512.

[67] Yagiz Sutcu, Shantanu Rane, Jonathan S Yedidia, Stark C Draper, and Anthony Vetro. 2008. Feature extraction for a Slepian-Wolf biometric system using LDPC codes. In *2008 IEEE International Symposium on Information Theory*. IEEE, 2297–2301.

[68] Andrew BJ Teoh, Alwyn Goh, and David CL Ngo. 2006. Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 12 (2006), 1892–1901.

[69] Vanessa Testoni and Darko Kirovski. 2010. On the inversion of biometric templates by an example. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 1830–1833.

[70] Yangguang Tian, Yingjiu Li, Ximeng Liu, Robert H Deng, and Binanda Sengupta. 2018. Pribioauth: Privacy-preserving biometric-based remote user authentication. In *2018 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE, 1–8.

[71] Umut Uludag and Anil K. Jain. 2004. Attacks on biometric systems: a case study in fingerprints. In *Security, Steganography, and Watermarking of Multimedia Contents VI*. International Society for Optics and Photonics, 622–633.

[72] Umut Uludag and Anil K Jain. 2004. Fuzzy fingerprint vault. In *Proc. Workshop: Biometrics: Challenges arising from theory to practice*. 13–16.

[73] Qian Wang, Shengshan Hu, Kui Ren, Meiqi He, Minxin Du, and Zhibo Wang. 2015. CloudBI: Practical privacy-preserving outsourcing of biometric identification in the cloud. In *European Symposium on Research in Computer Security*. Springer, 186–205.

[74] Yige Wang, Shantanu Rane, and Anthony Vetro. 2009. Leveraging reliable bits: ECC design considerations for practical secure biometric systems. In *2009 First IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 71–75.

[75] Wai Kit Wong, David Wai-lok Cheung, Ben Kao, and Nikos Mamoulis. 2009. Secure kNN computation on encrypted databases. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. 139–152.

[76] Can Xiang, Chunming Tang, Yunlu Cai, and Qiuxia Xu. 2016. Privacy-Preserving Face Recognition with Outsourced Computation. *Soft Comput.* 20, 9 (Sept. 2016), 3735–3744. https://doi.org/10.1007/s00500-015-1759-5

[77] T. Xiang, S. Guo, and X. Li. 2016. Perceptual Visual Security Index Based on Edge and Texture Similarities. *IEEE Transactions on Information Forensics and Security* 11, 5 (May 2016), 951–963. https://doi.org/10.1109/TIFS.2016.2515503

[78] Yasushi Yamazaki, Akane Nakashima, Kazunobu Tasaka, and Naohisa Komatsu. 2005. A study on vulnerability in on-line writer verification system. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*. IEEE, 640–644.

[79] Shenglin Yang and Ingrid Verbauwhede. 2005. Automatic secure fingerprint verification system based on fuzzy vault scheme. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, Vol. 5. IEEE, v–609.

[80] Andrew Chi-Chih Yao. 1986. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*. IEEE, 162–167.

[81] Jiawei Yuan and Shucheng Yu. 2013. Efficient privacy-preserving biometric identification in cloud computing. In *2013 Proceedings IEEE INFOCOM*. IEEE, 2652–2660.

[82] Chuan Zhang, Liehuang Zhu, and Chang Xu. 2017. PTBI: An efficient privacy-preserving biometric identification based on perturbed term in the cloud. *Information Sciences* 409 (2017), 56–67.

[83] Kai Zhou and Jian Ren. 2018. Passbio: Privacy-preserving user-centric biometric authentication. *IEEE Transactions on Information Forensics and Security* 13, 12 (2018), 3050–3063.