Privacy-Preserving Fingerprint Authentication Resistant to Hill-Climbing Attacks

Haruna Higo[1(✉)], Toshiyuki Isshiki[1], Kengo Mori[1], and Satoshi Obana[2]

[1] NEC Corporation, Tokyo, Japan
h-higo@aj.jp.nec.com, {t-issiki,ke-mori}@bx.jp.nec.com
[2] Hosei University, Tokyo, Japan
obana@hosei.ac.jp

Abstract. This paper proposes a novel secure biometric authentication scheme that hides the biometric features and the distance between the enrolled and authenticated biometric features, and prevent impersonation. To confirm that the proposed scheme has such properties, we formally model secure biometric authentication schemes by generalizing the related and proposed schemes. As far as we know, the proposed scheme is the first one that has been proved to satisfy all the properties. In particular, the proposed scheme achieves security under the decisional Diffie-Hellman assumption.

Keywords: Biometric authentication · Fingerprint minutiae · Hill-climbing attack · Privacy-preserving technology

1 Introduction

Background. Traditionally, ID/passwords and tokens including cards are widely used as a means of authentication. However, they are at risk of being forgot or stolen. In contrast, biometric characteristics including fingerprint, face, palm veins, palm print, iris, and retina cannot be forgot or stolen. Therefore, biometric authentication has an advantage compared to traditional authentication means.

Biometric authentication makes use of the similarity of biometric features extracted from the same biometric characteristic. If a biometric feature presented by a client is similar enough to an enrolled biometric feature in some distance metrics, the client is successfully authenticated by the server. Since biometric features are unchangeable private information, it is required to prevent them from being leaked. Moreover, impersonation should also be prevented.

Related Works. For protecting biometric features from being leaked and preventing impersonation, many schemes have been proposed [16,18]. Some of them exploit secret information (i.e., helper data) that is remembered [6,12,13,15] or tokens (e.g., smart cards or devices) brought along [5,14] by the clients in addition to their biometric characteristics. However, to exploit the advantage of biometric characteristics that there is no risk of them being forgotten or stolen, it is preferable not to use other secret information that has such risks.

O. Dunkelman and L. Keliher (Eds.): SAC 2015, LNCS 9566, pp. 44–64, 2016.
DOI: 10.1007/978-3-319-31301-6_3

Without additional secrets, some schemes [1,3,4,10,11] employ a third entity called a decryptor that manages a secret key to protect biometric features from being leaked and prevent impersonation. In the schemes in [1,3,4,10,11], the decryptor computes the distance between the enrolled and authenticated biometric features and compares them with a predetermined threshold. However, it is known that the distances are useful for hill-climbing attacks [19] in which the attacker guesses the enrolled biometric feature by observing the change in the distance from multiple authentication trials.

In ACISP 2012, Shahandashti, Safavi-Naini, and Ogunbona [17] proposed a fingerprint matching scheme using minutiae. Minutiae are feature points in fingerprints and are widely used for fingerprint matching. Their scheme makes use of polynomials that are evaluated to be 0 or 1 in accordance with the correspondence of the two input fingerprints. Since the polynomials are evaluated to be binary values, the scheme prevents the distances from being leaked. However, the scheme requires the server to store the enrolled fingerprint itself (i.e., not an encrypted version) while the scheme hides the biometric features during the authentication due to leveraging homomorphic encryption.

Contributions of this Paper. The main contribution of this paper is proposing a secure biometric authentication scheme that uses fingerprint minutiae. The proposed scheme is designed for hiding information of enrolled and authenticated minutiae and the distance between them, and preventing impersonation. The comparison of the minutiae is done in accordance with their locations and orientations, which is the well-known method as in [14,17]. In the proposed scheme, enrolled minutiae are represented in the form of polynomials that are evaluated to be 1 or a random value in accordance with whether the input minutia is considered to be the same as the enrolled one. The scheme utilizes the modified Elgamal cryptosystem to evaluate the polynomials without leaking information of minutiae and distance [8,17]. Similar to the previous schemes [1,3,4,10,11], the decryptor is employed in addition to the server and clients, and it manages the secret key of the homomorphic encryption scheme. Since the operations handled by the decryptor are only decryption and comparison of plaintexts, the decryptor of our scheme can be implemented by hardware security modules (HSM). Therefore, it seems that utilization of the decryptor is realistic with respect to the proposed scheme.

To analyze the security of the proposed scheme, we formally model a secure biometric authentication scheme. The model is a generalization of the previous schemes [1,3,4,10,11] and the proposed one. That is, three types of entities, the server, clients, and decryptor, are employed. We formalize the following four security requirements: (a) hide biometric features from the server (which we call template protection against server), (b) hide biometric features from the decryptor (which we call template protection against decryptor), (c) prevent impersonation (which we call security for authentication), and (d) hide distances from the decryptor (which we call security against hill-climbing attacks). Requirements (a) and (c) are defined by generalizing the security definition provided by Hirano et al. [11] while we newly define requirement (b) and (d). We prove that the

proposed scheme satisfies all requirements under standard cryptographic assumptions. In particular, we prove that the proposed scheme satisfies all requirements under the decisional Diffie-Hellman (DDH) assumption.

Table 1. Comparison with previous schemes.

Scheme	[17]	[11]	This paper
Representation of biometric features	Minutiae	Vector	Minutiae
Number of entities	2	3	3
BGN cryptosystem	Not used	Necessary	Not used
Template protection against server	No	Yes	Yes
Template protection against decryptor	–	No	Yes
Security for authentication	Yes	Yes	Yes
Security against hill-climbing attacks	Yes	No	Yes

We compare the related works and our work in Table 1. Shahandashti et al.'s scheme [17] is performed by a server and a client (a decryptor is not included) and hides authenticated biometric features. The template is the information of minutiae in the enrolled fingerprint itself and is not concealed. Therefore, it does not satisfy the requirements for template protection. On the other hand, the scheme satisfies the other two notions. Due to employing a third party, our scheme makes it possible to protect the information of both the enrolled and authenticated biometric features and the distance between them. As mentioned above, the decryptor in Hirano et al.'s scheme [11] does not hide the distance from the decryptor. Also, the scheme makes use of a special type of homomorphic encryption scheme with which evaluation of 2-DNF formulas is feasible when performed on ciphertext introduced by Boneh et al. [2].

2 Preliminaries

In this section, we describe preliminaries that are used in the proposed scheme.

2.1 Homomorphic Encryption Scheme

The homomorphic encryption scheme is a type of public key encryption scheme that has a special property. The property is that from ciphertexts, a new ciphertext corresponding to a result of some operation on the plaintexts can be generated without knowledge of the secret key. We focus on addition as the operation. That is, by using two ciphertexts $c_1 = \mathsf{Enc}(m_1)$ and $c_2 = \mathsf{Enc}(m_2)$, a ciphertext of $m_1 + m_2$ is computable. Such schemes are called additive homomorphic encryption schemes.

We utilize the modified (or lifted) Elgamal cryptosystem [7] in the proposed scheme. The modified Elgamal cryptosystem is an additive homomorphic encryption scheme where algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ run as follows:

- $(pk, sk) := ((p, g, y), (g, x)) \leftarrow \mathsf{Gen}(1^\kappa)$ where G is a group of prime order p, g is a generator of G, $x \in \mathbb{Z}_p$, and $y := g^x$.
- $(c_1, c_2) := (g^r, y^r g^m) \leftarrow \mathsf{Enc}(pk, m)$ where message m is in $\mathbb{Z}_p$ and $r \in \mathbb{Z}_p$ is chosen randomly.
- $m' := \log_g c_2/c_1^x = \mathsf{Dec}(sk, (c_1, c_2))$.

To divide Dec into two subalgorithms, we define two algorithms as $c_2/c_1^x = \mathsf{Dec}_1(x, (c_1, c_2))$ and $log_g z = \mathsf{Dec}_2(g, z)$. Apparently $\mathsf{Dec}(sk, (c_1, c_2)) = \mathsf{Dec}_2(g, \mathsf{Dec}_1(x, (c_1, c_2)))$ holds. The first algorithm Dec_1 is computable for any ciphertext while the second Dec_2 is not always feasible since it requires computing of the discrete logarithm, which is assumed to be hard in general. However, in the proposed scheme, we just check if the plaintext is equal to 0, which is feasible by verifying the result of Dec_1 is 1 or not.

The modified Elgamal cryptosystem has been proved to be IND-CPA secure under the decisional Diffie-Hellman (DDH) assumption that states that solving the DDH problem is hard. It is easy to see that the modified Elgamal cryptosystem has the homomorphic property. From two ciphertexts $c = (c_1, c_2) \leftarrow \mathsf{Enc}(m)$ and $c' = (c'_1, c'_2) \leftarrow \mathsf{Enc}(m')$, it holds that $c \cdot c' := (c_1 c'_1, c_2 c'_2) = (g^{r+r'}, y^{r+r'} g^{m+m'})$ which is a ciphertext of $m + m'$.

This property is applicable in evaluating polynomials. An n-th degree polynomial $F(X)$ can be represented in the form of $F(X) = \sum_{i=0}^{n} a_i \cdot X^i$. As explained above, $\mathsf{Enc}(a_i \cdot x^i) = \mathsf{Enc}(a_i)^{x^i}$ holds for any x, i, and a_i. Therefore, from encrypted coefficients $\mathsf{Enc}(a_0), \cdots, \mathsf{Enc}(a_i)$ and any x, a ciphertext of the evaluated value $F(x)$ is computable by just multiplying $\mathsf{Enc}(a_i \cdot x^i)$ for every i, since it holds that $\mathsf{Enc}(F(x)) = \mathsf{Enc}(\sum_{i=0}^{n} a_i \cdot x^i) = \prod_{i=0}^{n} \mathsf{Enc}(a_i \cdot x^i)$.

2.2 Biometric Authentication and Fingerprint Minutiae

Biometric authentication is a technique that uses biometric characteristics such as fingerprints for authenticating individuals. Two biometric features extracted from the same biometric characteristic are, in most cases, different but close in some metric. Therefore, to verify if two biometric features are derived from the same individual, it is sufficient to check if they are close under that metric.

A client who would like to enroll himself extracts a biometric feature from his biometric characteristics using some devices such as sensors and cameras. The server stores a template that is generated from the biometric feature. To make the server authenticate a client, the client extracts a biometric feature again. Then, the server estimates the distance between the biometric feature to be authenticated and the biometric feature that has been generated and stored in a template to check if they have originated from the same biometric characteristic.

A fingerprint contains a number of ridges. Some of them abruptly end (called ridge endings), and others are divided into two ridges (called ridge bifurcations). Feature points such as ridge endings and ridge bifurcations are called *minutiae*. In general, a minutia is represented by its location (x, y) and orientation t. Different types (e.g., ridge endings or ridge bifurcations) are also used in some cases. We assume that the coordinate system is aligned every time biometric

characteristics are captured. We refer the readers to [14] and its references for information on pre-alignment techniques.

In authenticating an individual with minutiae, a set of minutiae is extracted from the fingerprint. Two fingerprints are considered to match if they have more than a threshold number of pairs of corresponding minutiae.

Two minutiae are said to correspond if their locations and orientations are close enough. That is, two minutiae $((x, y), t)$ and $((x', y'), t')$ correspond if both $d_2((x, y), (x', y')) := \sqrt{(x - x')^2 + (y - y')^2} \leq \Delta_d$ and $d_1(t, t') := |t - t'| \leq \Delta_t$ hold where Δ_d and Δ_t are predetermined thresholds, and d_1 and d_2 stand for the Euclidean distance in one and two dimensions, respectively. In this paper, each location and orientation are assumed to be represented by integers.

3 Secure Biometric Authentication Schemes

We provide formal definitions of the secure biometric authentication scheme in this section. First, the components and procedures of the scheme are explained. After that, we define its security in accordance with Hirano et al.'s definition [11].

3.1 Algorithms and Procedures

There are three kinds of entities, a server, clients, and a decryptor, in the model of secure biometric authentication scheme. A client uses his own biometric feature to enroll or authenticate himself. Clients are not required to have any secret information other than their own biometric characteristics. The enrolled information is stored by the server. Authentication is performed with the aid of the decryptor who has the secret key. The server decides the authentication result in accordance with whether the enrolled and authenticated biometric features are considered to have originated from the same biometric characteristic.

The procedures of the secure biometric authentication scheme include three phases, setup, enrollment, and authentication. The setup phase is done only once, and afterward the enrollment phase and the authentication phase are executed repeatedly by the clients in an arbitrary order. We now describe the procedures of each phase in detail (Fig. 1).

In the setup phase, the decryptor executes the setup algorithm. It takes as input the security parameter and a tuple of parameters, which includes information about the metrics for evaluating distance and the thresholds of acceptance to generate a public parameter and a secret key. The public parameter is published while the secret key is kept secret from other entities.

In the enrollment phase, a client who would like to register himself on the system runs the pseudonymous identifier encoder (PIE). It generates a protected template from the client's biometric feature. The protected template is sent to the server. The server sets identification data for the client, and the client is informed of the identification data. The protected template with the identification data is stored by the server. Depending on the application, the identification data is decided by the client and the client notifies the client of it.

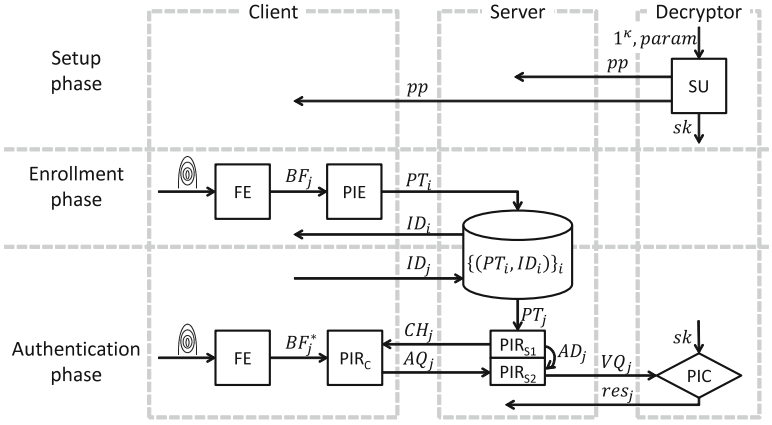

Fig. 1. Algorithms and procedures of secure biometric authentication scheme.

At the beginning of the authentication phase, a client who would like to be authenticated by the server shows his/her identification data to the server. The server selects the protected template that is associated with the identification data and interacts with the client through the pseudonymous identifier recoder (PIR) (for simplicity, we divide the PIR into three algorithms in the definition below). Finally, the server sends a verification query to the decryptor who runs the pseudonymous identifier comparator (PIC) with its secret key to determine the authentication result.

As mentioned in Sect. 2.2, the PIE and the PIR take as input a pre-aligned biometric feature extracted from a biometric characteristic. In Fig. 1, we denote the feature extraction algorithm by FE. This algorithm captures a biometric characteristic and outputs an appropriately aligned biometric feature. Since the pre-alignment technique is out of the range of the secure biometric authentication scheme, the feature extraction algorithm is not included in the tuple of the secure biometric authentication scheme.

The secure biometric authentication scheme is defined as follows. Note that the previous schemes [1,3,4,10,11] can be adapted to the formalization.

Definition 1. *A secure biometric authentication scheme is a tuple of six algorithms* $(\mathsf{SU}, \mathsf{PIE}, \mathsf{PIR}_{S,1}, \mathsf{PIR}_C, \mathsf{PIR}_{S,2}, \mathsf{PIC})$ *that satisfy the following:*

- $(pp, sk) \leftarrow \mathsf{SU}(1^\kappa, param)$ *on input security parameter* κ *and parameter* $param$*, outputs public parameter* pp *and secret key* sk*.*
- $PT \leftarrow \mathsf{PIE}(pp, BF)$ *on input public parameter* pp *and biometric feature* BF*, outputs protected template* PT*.*
- $(CH, AD) \leftarrow \mathsf{PIR}_{S,1}(pp, PT)$ *on input public parameter* pp *and protected template* pp*, outputs challenge* CH *and auxiliary data* AD*.*
- $AQ \leftarrow \mathsf{PIR}_C(pp, BF, CH)$ *on input public parameter* pp*, biometric feature* BF*, and challenge* CH*, outputs authentication query* AQ*.*

- $VQ \leftarrow \mathsf{PIR}_{S,2}(pp, AQ, AD)$ *on input public parameter* pp*, authentication query* AQ*, and auxiliary data* AD*, outputs verification query* VQ*.*
- $res \leftarrow \mathsf{PIC}(sk, VQ)$ *on inputs secret key* sk *and verification query* VQ*, outputs authentication result* $res \in \{Accept, Reject\}$*.*

When it is obvious from the context, we omit pp from the input of the algorithms.

For $(pp, sk) \leftarrow \mathsf{SU}(1^\kappa)$ and two biometric features BF_e and BF_a, let $(CH, AD) \leftarrow \mathsf{PIR}_{S,1}(\mathsf{PIE}(BF_e))$, $AQ \leftarrow \mathsf{PIR}_C(BF_a, CH)$, $(VQ) \leftarrow \mathsf{PIR}_{S,2}(pp, AQ, AD)$, and $res \leftarrow \mathsf{PIC}(sk, VQ)$. For correctness, we assume that if BF_e and BF_a are extracted from the same biometric characteristic, then $res = Accept$ holds; otherwise $res = Reject$.

3.2 Security

Adversarial clients may try to impersonate a legitimate user while an adversarial server and decryptor aim to obtain some information about the enrolled biometric features. We define these properties under the proposed framework described in the previous section. Note that the decryptor is assumed not to collude with any other entity. We formalize the following four security requirements: (a) hide biometric features from the server (which we call template protection against server), (b) hide biometric features from the decryptor (template protection against decryptor), (c) prevent impersonation (security for authentication), and (d) hide the distances between the enrolled and authenticated biometric features from the decryptor (security against hill-climbing attacks).

Hirano et al. [11] defined security that is specific to their scheme in the semi-honest model where the adversary is considered to corrupt some clients. We follow the definition of [11] about requirements (a) and (c) but we slightly modify the definitions to make them applicable to the proposed framework.

The major difference between the definitions in [11] and ours is to consider the security against an adversarial decryptor. Since the decryptor possesses the secret key, the decryptor is so powerful in the proposed framework that it can even obtain the biometric feature itself by mounting hill-climbing attack in some schemes (e.g., [11]). To capture such attacks by the decryptor, we newly define requirements (b) and (d). Here requirement (b) is defined in the semi-honest model similar to the definition of requirement (a). On the other hand, we newly define requirement (d) as the inability of the decryptor to obtain any information other than the authentication results.

Note that the definitions of requirements (a), (b), and (c) are in the semi-honest model similar to the definitions in [11] while we consider malicious adversaries in the definition of requirement (d). Below, let $(\mathsf{SU}, \mathsf{PIE}, \mathsf{PIR}_{S,1}, \mathsf{PIR}_C, \mathsf{PIR}_{S,2}, \mathsf{PIC})$ be a tuple that satisfies Definition 1.

Template protection against server. This security requirement captures an adversarial server that has templates and authentication queries from clients and try to obtain enrolled biometric features. We introduce a security game between challenger $\mathcal{C}$ and attacker $\mathcal{A}$ as follows (Fig. 2).

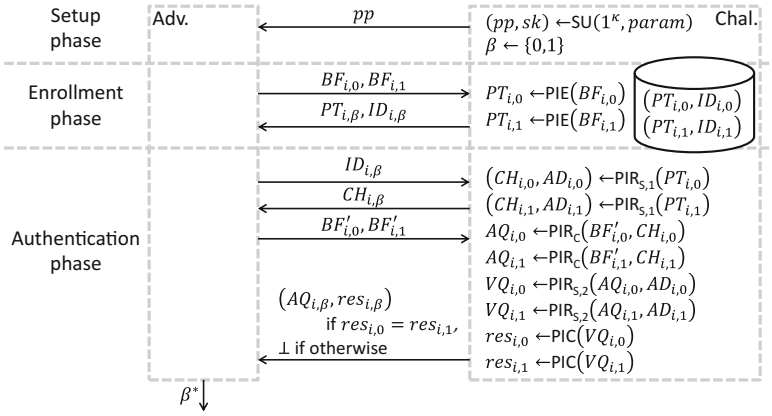

Fig. 2. Game for template protection against server.

Setup: $\mathcal{C}$ runs the setup algorithm to obtain (pp, sk) and chooses bit β randomly. pp is sent to $\mathcal{A}$.

Enrollment: As for the i-th query, $\mathcal{A}$ chooses and sends two biometric features $BF_{i,0}$ and $BF_{i,1}$ to $\mathcal{C}$. $\mathcal{C}$ runs $PT_{i,0} \leftarrow \mathsf{PIE}(BF_{i,0})$ and $PT_{i,1} \leftarrow \mathsf{PIE}(BF_{i,1})$ and selects two identification data $ID_{i,0}$ and $ID_{i,1}$ from $\mathcal{ID}$. $\mathcal{C}$ stores two pairs $(PT_{i,0}, ID_{i,0})$ and $(PT_{i,1}, ID_{i,1})$ and returns $(PT_{i,\beta}, ID_{i,\beta})$ to $\mathcal{A}$.

Authentication: For identification data ID_i from $\mathcal{A}$, $\mathcal{C}$ executes $(CH_{i,0}, AD_{i,0}) \leftarrow \mathsf{PIR}_{S,1}(PT_{i,0})$ and $(CH_{i,1}, AD_{i,1}) \leftarrow \mathsf{PIR}_{S,1}(PT_{i,1})$. Given $CH_{i,\beta}$ from $\mathcal{C}$, $\mathcal{A}$ chooses and sends to $\mathcal{C}$ two biometric features $BF'_{i,0}$ and $BF'_{i,1}$. Then, $\mathcal{C}$ runs $AQ_{i,0} \leftarrow \mathsf{PIR}_C(BF'_{i,0}, CH_i)$, $AQ_{i,1} \leftarrow \mathsf{PIR}_C(BF'_{i,1}, CH_i)$, $VQ_{i,0} \leftarrow \mathsf{PIR}_{S,2}(AQ_{i,0}, AD_i)$, $VQ_{i,1} \leftarrow \mathsf{PIR}_{S,2}(AQ_{i,1}, AD_i)$, $res_{i,0} \leftarrow \mathsf{PIC}(VQ_{i,0})$, and $res_{i,1} \leftarrow \mathsf{PIC}(VQ_{i,1})$, sequentially. $\mathcal{C}$ returns $(AQ_{i,\beta}, res_{i,\beta})$ to $\mathcal{A}$ if $res_{i,0} = res_{i,1}$ and $\perp$ otherwise.

Output: Finally, $\mathcal{A}$ outputs β^*.

Note that the enrollment and authentication phases can be repeated in an arbitrary order.

The advantage of $\mathcal{A}$ is defined as $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{TP,S}}(\kappa) := \Pr[\beta = \beta^*] - 1/2$. With this advantage, the security property is defined as follows.

Definition 2. *We say that a biometric authentication scheme satisfies template protection against server if for any PPT $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{TP,S}}(\kappa) \leq negl(\kappa)$.*

Template protection against decryptor. As in Fig. 3, to capture an adversarial decryptor obtaining enrolled biometric features, we slightly modify the game for template protection against server.

Definition 3. *We say that a biometric authentication scheme satisfies template protection against decryptor if for any PPT $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{TP,D}}(\kappa) \leq negl(\kappa)$.*

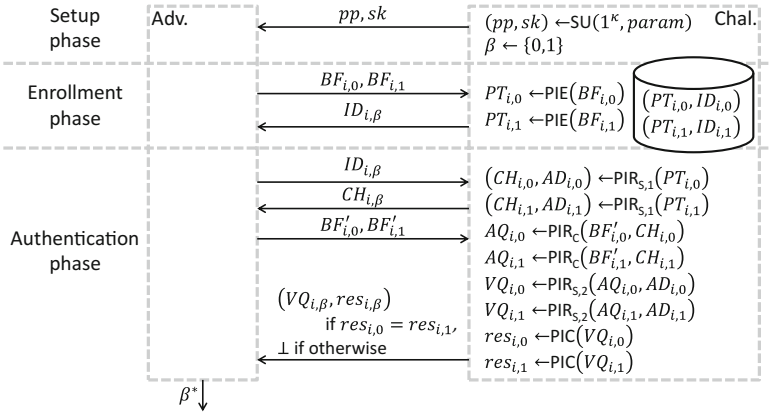

Fig. 3. Game for template protection against decryptor.

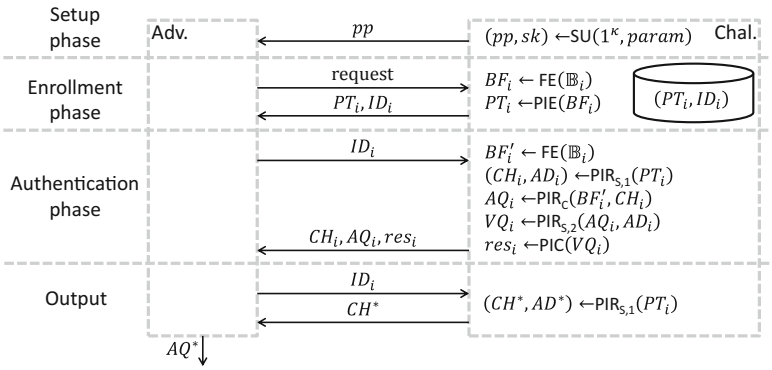

Fig. 4. Game for security for authentication.

Security for authentication. To capture an illegitimate client generating a valid authentication query, we define the following game (Fig. 4).

Setup: $\mathcal{C}$ runs the setup algorithm to obtain (pp, sk), and pp is sent to $\mathcal{A}$.

Enrollment: On the i-th request from $\mathcal{A}$, $\mathcal{C}$ chooses a biometric characteristic $\mathbb{B}_i$ and extracts a biometric feature $BF_i \leftarrow \mathbb{B}_i$. Also, $\mathcal{C}$ chooses identification data $ID_i \in \mathcal{ID}$. Then, $\mathcal{C}$ stores the pair (PT_i, ID_i) and also returns it to $\mathcal{A}$.

Authentication: For identification data ID_i from $\mathcal{A}$, $\mathcal{C}$ extracts a biometric feature from the i-th biometric characteristic $BF'_i \leftarrow \mathbb{B}_i$. With the protected template PT_i that is stored with ID_i, $\mathcal{C}$ executes $(CH_i, AD_i) \leftarrow \mathsf{PIR}_{S,1}(PT_i)$, $AQ_i \leftarrow \mathsf{PIR}_C(BF'_i, CH_i)$, $VQ_i \leftarrow \mathsf{PIR}_{S,2}(AQ_i, AD_i)$, and $res_i \leftarrow \mathsf{PIC}(VQ_i)$, sequentially, and returns (CH_i, AQ_i, res_i) to $\mathcal{A}$.

Output: For identification data ID_i from $\mathcal{A}$, $\mathcal{C}$ executes $(CH^*, AD^*) \leftarrow \mathsf{PIR}_{S,1}(PT_i)$ and returns CH^*. Finally, $\mathcal{A}$ outputs AQ_i^*.

Note that the enrollment and authentication phases can be repeated in an arbitrary order.

The advantage of $\mathcal{A}$ is defined as $\mathsf{Adv}^{\mathsf{Auth}}_{\mathcal{A}}(\kappa) := \Pr[\mathsf{PIC}(\mathsf{PIR}_{S,2}(AQ^*, AD^*)) = Accept]$. With this advantage, security for authentication is defined as follows.

Definition 4. *We say that a biometric authentication scheme is secure in the sense of authentication if for any PPT* $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{Auth}}_{\mathcal{A}}(\kappa) \leq negl(\kappa)$.

Security against hill-climbing attacks. It is preferable that the decryptor obtains as little information as possible. For example, distance is useful in guessing the enrolled biometric feature. Such guessing attacks are called hill-climbing attacks [19]. In the attacks, an attacker casts two queries and learns the distances between the queried biometric features and the enrolled one. From the distances, the attacker is able to learn which query is nearer to the enrolled one. Repeating this approach, he will successfully obtain some accepted queries. Therefore, it is preferable that the schemes do not to give out any information other than the authentication result (acceptance or rejection) to the decryptor. We define the following game (Fig. 5).

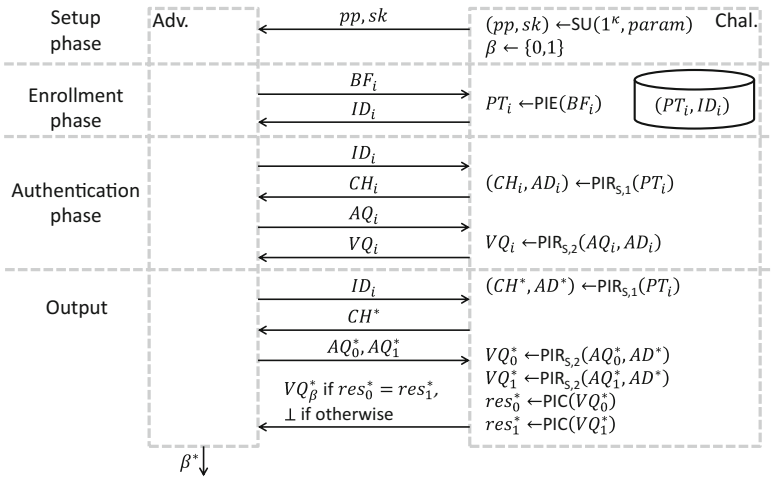

Fig. 5. Game for security against hill-climbing attacks.

Setup: $\mathcal{C}$ runs the setup algorithm to obtain (pp, sk) and chooses $\beta \in \{0, 1\}$ randomly. (pp, sk) is sent to $\mathcal{A}$.

Enrollment: On the i-th request from $\mathcal{A}$ for biometric feature BF_i, C runs $PT_i \leftarrow \mathsf{PIE}(pp, BF_i)$ and chooses index $ID_i \in \mathcal{ID}$. $\mathcal{C}$ stores pair (PT_i, ID_i) and returns ID_i to $\mathcal{A}$.

Authentication: For identification data ID_i from $\mathcal{A}, \mathcal{C}$ executes $(CH_i, AD_i) \leftarrow \mathsf{PIR}_{S,1}(PT_i)$ where PT_i is the protected templates that are stored with ID_i. Given CH_i, $\mathcal{A}$ chooses and sends to $\mathcal{C}$ an authentication query AQ_i. Then, $\mathcal{C}$ runs $VQ_i \leftarrow \mathsf{PIR}_{S,2}(AQ_i, AD_i)$ and returns VQ_i to $\mathcal{A}$.

Output: For identification data ID_i and two biometric features BF_0^* and BF_1^* from $\mathcal{A}$, $\mathcal{C}$ executes $(CH^*, AD^*) \leftarrow \mathsf{PIR}_{S,1}(PT_i)$, $AQ_0^* \leftarrow \mathsf{PIR}_C(BF_0^*, CH^*)$, $AQ_1^* \leftarrow \mathsf{PIR}_C(BF_1^*, CH^*)$, $VQ_0^* \leftarrow \mathsf{PIR}_{S,2}(AQ_0^*, AD^*)$, $VQ_1^* \leftarrow \mathsf{PIR}_{S,2}(AQ_1^*, AD^*)$, $res_0^* \leftarrow \mathsf{PIC}(VQ_0^*)$, and $res_1^* \leftarrow \mathsf{PIC}(VQ_1^*)$, sequentially. Then, $\mathcal{C}$ chooses $\beta \in \{0, 1\}$ randomly and returns VQ_β^* to $\mathcal{A}$ if $res_0^* = res_1^*$ and returns $\perp$ otherwise. Finally, $\mathcal{A}$ outputs β^*.

Note that the enrollment and authentication phases can be repeated in an arbitrary order.

The advantage of $\mathcal{A}$ is defined as $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Dist}}(\kappa) := \Pr[\beta = \beta^*] - 1/2$. With this advantage, security against hill-climbing attacks is defined as follows.

Definition 5. *We say that a biometric authentication scheme is secure against hill-climbing attacks if for any PPT* $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Dist}}(\kappa) \leq negl(\kappa)$.

4 Proposed Scheme

We propose a secure biometric authentication scheme that uses fingerprint minutiae. That is, the biometric features that the proposed scheme deals with consist of a set of minutiae. Acceptance is decided by the closeness of the minutiae as explained in Sect. 2.2. Note that it is easy to extend the scheme to deal with the types of minutiae.

4.1 Construction

Here, we propose a secure biometric authentication scheme. Prior to the in-depth description, we give an outline of the scheme.

In the enrollment phase, two polynomials are generated in accordance with the location and orientation of each minutia in the enrolled fingerprint. The polynomials are generated to satisfy the following condition: if a minutia that is close to the enrolled minutia is input into the polynomials, both of them result in 1. The template is a tuple of encrypted coefficients of the polynomials.

Later in the authentication phase, the polynomials are evaluated for every minutia of the fingerprint to be authenticated in the ciphertext domain. Then, the decryptor with the secret key checks to find out if the evaluated values of polynomials are 1 to determine if the authenticated minutia and the enrolled minutia are close enough to be corresponding minutiae.

As explained above, the proposed scheme proceeds in the same way for every pair of minutiae in the enrolled and authenticated fingerprints. Therefore, for simplicity, we explain the proposed scheme for matching a pair of minutiae. By just doing the same for every minutia pair, it is easy to extend the scheme to deal with fingerprints that consist of multiple minutiae.

Now we describe the algorithms of the proposed scheme ($\mathsf{SU}, \mathsf{PIE}, \mathsf{PIR}_{S,1}$, $\mathsf{PIR}_C, \mathsf{PIR}_{S,2}, \mathsf{PIC}$). In the following algorithms, $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ represents the modified Elgamal cryptosystem of which the plaintext domain is denoted by $\mathbb{Z}_p$. For simplicity, the homomorphic operations are described as $\mathsf{Enc}(m_1+m_2) = \mathsf{Enc}(m_1)\mathsf{Enc}(m_2)$ and $\mathsf{Enc}(m)^i = \mathsf{Enc}(im)$. In addition, we define a ciphertext transformation function Trans by $\mathsf{Trans}(r; (c_1, c_2)) = (c_1^r, c_2)$ where $r \in \mathbb{Z}_p$ and (c_1, c_2) is a ciphertext. Informally, without r, the decryptor cannot decrypt the transformed ciphertexts. It is easy to see that the additive homomorphic property also holds for transformed ciphertexts.

$\mathsf{SU}(1^\kappa, \mathit{param})$ where param includes the thresholds Δ_d and Δ_t:

1. Generates a public/secret key pair of the homomorphic encryption scheme as $(pk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$.
2. Outputs public parameter $pp := (pk, \mathit{param})$ and the secret key sk.

$\mathsf{PIE}(BF = ((x, y), t))$:

1. Generates two polynomials

$$F(X, Y) = \sum_i \sum_j a_{i,j} \cdot X^i \cdot Y^j, \text{ and} \tag{1}$$

$$G(T) = \sum_k b_k \cdot T^k \tag{2}$$

that satisfy $F(X, Y) = 1$ if $d_2((X, Y), (x, y)) \leq \Delta_d$ and $G(T) = 1$ if $d_1(T, t) \leq \Delta_t$ (the details of the polynomials including the values i, j, and k are provided below).

2. Encrypts $A_{i,j} \leftarrow \mathsf{Enc}(a_{i,j})$ and $B_k \leftarrow \mathsf{Enc}(b_k)$, and outputs the protected template $PT := (\{A_{i,j}\}_{i,j}, \{B_k\}_k)$.

$\mathsf{PIR}_{S,1}(PT = (\{A_{i,j}\}_{i,j}, \{B_k\}_k))$:

1. Chooses r_F, r_G, and r randomly from $\mathbb{Z}_p$ and computes $A'_{i,j} := \mathsf{Trans}(r; A_{i,j}^{r_F})$, $B'_k := \mathsf{Trans}(r; B_k^{r_G})$ for every i, j, and k.
2. Encrypts $R \leftarrow \mathsf{Enc}(-r_F - r_G)$.
3. Lets $CH := (\{A'_{i,j}\}_{i,j}, \{B'_k\}_k)$ and $AD := (R, r)$ and outputs (CH, AD).

$\mathsf{PIR}_C(CH = (\{A'_{i,j}\}_{i,j}, \{B'_k\}_k), BF^* = ((x^*, y^*), t^*))$:

1. Outputs $AQ := \prod_i \prod_j \left(A'^{(x^*)^i (y^*)^j}_{i,j}\right) \cdot \prod_k B'^{(t^*)^k}_k$.

$\mathsf{PIR}_{S,2}(AQ, AD = (R, r))$:

1. Chooses $r' \in \mathbb{Z}_p$ randomly and outputs $VQ := (\mathsf{Trans}(1/r; AQ) \cdot R)^{r'}$.

$\mathsf{PIC}(VQ)$:

1. If $\mathsf{Dec}_1(VQ) = 1$ then outputs *Accept*; otherwise outputs *Reject*.

Before explaining the details of the construction of the polynomials, let us confirm the correctness of the proposed scheme. Since it holds that $A'_{i,j} = (\mathsf{Enc}(a_{i,j}))^{r_F} = \mathsf{Enc}(r_F \cdot a_{i,j})$ and $B'_k = (\mathsf{Enc}(b_k))^{r_G} = \mathsf{Enc}(r_G \cdot b_k)$ for any i, j, and k, the challenge from the server is a tuple of encrypted coefficients of randomized polynomials $F'(X, Y) = r_F \cdot F(X, Y) = \sum_i \sum_j (r_F \cdot a_{i,j}) \cdot X^i \cdot Y^j$

and $G'(T) = r_G \cdot G(T) = \sum_k (r_G \cdot b_k) \cdot T^k$. They satisfy $F'(X, Y) = r_F$ and $G'(T) = r_G$ if minutiae $((x, y), t)$ and $((X, Y), T)$ correspond. The authentication query is a ciphertext of the sum of the evaluated value of the polynomials for the authenticated minutia as

$$\begin{aligned}\mathsf{Trans}(1/r; AQ) &= \prod_i \prod_j \mathsf{Enc}(r_F \cdot a_{i,j})^{(x^*)^i (y^*)^j} \cdot \prod_k \mathsf{Enc}(r_G \cdot b_k)^{(t^*)^k} \\ &= \mathsf{Enc}\left(F'(x^*, y^*) + G'(t^*)\right).\end{aligned}$$

Therefore, the verification query computed by the server is

$$\begin{aligned}VQ &= \left(\mathsf{Enc}\left(F'(x^*, y^*) + G'(t^*)\right) \cdot \mathsf{Enc}\left(-r_F - r_G\right)\right)^{r'} \\ &= \mathsf{Enc}\left(r'\left(r_F\left(F(x^*, y^*) - 1\right) + r_G\left(G(t^*) - 1\right)\right)\right).\end{aligned}$$

Thus, if the enrolled and authenticated minutiae correspond, VQ is a ciphertext of 0 that is $\mathsf{Dec}_1(VQ) = 1$ holds.

Description of polynomials. In the above scheme, polynomials $F(X, Y)$ and $G(T)$ that satisfy $F(X, Y) = 1$ if $d_2((X, Y), (x, y)) \leq \Delta_d$ and $G(T) = 1$ if $d_1(T, t) \leq \Delta_t$, respectively, are generated in accordance with enrolled minutia $\{((x, y), t)\}$. Such polynomials are constructible as

$$F(X, Y) = R_F \prod_{\ell \in L} \left\{(X - x)^2 + (Y - y)^2 - \ell\right\} + 1, \text{ and} \tag{3}$$

$$G(T) = R_G \prod_{\ell = -\Delta_t}^{\Delta_t} \left\{(T - t) - \ell\right\} + 1, \tag{4}$$

where R_F and R_G are randomly chosen and the set $L := \{d_2^2((X, Y), (X', Y')) \leq \Delta_d^2 \mid X, X', Y, Y' \in \mathbb{Z}\}$ consists of the possible values of the squared Euclidean distance that are smaller than the squared threshold Δ_d^2. In the above scheme, we describe the functions as an expanded form as $F(X, Y) = \sum_{i=0}^{2|L|} \sum_{j=0}^{2|L|} a_{i,j} \cdot X^i \cdot Y^j$ and $G(T) = \sum_{k=0}^{2\Delta_t + 1} b_k \cdot T^k$.

It is easy to see that they satisfy the required properties. $F(X, Y) - 1 = R_F \prod_{\ell \in L} \left\{(X - x)^2 + (Y - y)^2 - \ell\right\} = 0$ holds for all (X, Y) that satisfy $d_2^2((X, Y), (x, y)) = (X - x)^2 + (Y - y)^2 = \ell$ for some $\ell \in L$. $G(T) - 1 = R_G \prod_{\ell = -\Delta_t}^{\Delta_t} \{(T - t) - \ell\} = 0$ holds for all T that satisfy $d_1(T, t) = T - t = \ell$ for some $\ell \in \{-\Delta_t, \cdots, \Delta_t\}$.

Recall that all locations are integers, the set L is a subset of and is smaller than the set $\{0, 1, \cdots, \Delta_d^2\}$. For example, there exists no tuple (X, X', Y, Y') that satisfies $d_2^2((X, Y), (X', Y') = 3$. Therefore, $3 \notin L$ holds. Set L is determined by Δ_d, for example, $L = \{0, 1, 2, 4, 5, 8, 9, 10, 13, 16, 17, 18, 20, 25\}$ when $\Delta_d = 5$. The number of terms in the derived polynomial F is computed to be $2|L|^2 + 3|L| + 1$.

4.2 Security

We show that the proposed scheme satisfies the security notions defined in Sect. 3.2. Detailed proofs are provided in Appendix A.

Template protection against server and template protection against decryptor. Under the IND-CPA security of the modified Elgamal cryptosystem, the proposed scheme can be proved to satisfy template protection against server and template protection against decryptor.

Theorem 1. *Under the IND-CPA security of the modified Elgamal cryptosystem, the proposed scheme satisfies template protection against server.*

Theorem 2. *Under the IND-CPA security of the modified Elgamal cryptosystem, the proposed scheme satisfies template protection against decryptor.*

Security for authentication. Under Assumption 1, we can prove Theorem 3.

Assumption 1. *For any PPT algorithm $\mathcal{A}$, it holds that*

$$\Pr\left[t = \mathsf{Dec}(t^*) \;\middle|\; \begin{array}{l}(pk, sk) \leftarrow \mathsf{Gen}(1^\kappa); s, t \leftarrow \{0,1\}^\kappa; \\ t^* \leftarrow \mathcal{A}(pk, \mathsf{Enc}(s), \mathsf{Enc}(st));\end{array}\right] \leq negl(\kappa).$$

Theorem 3. *Provided that the modified Elgamal cryptosystem satisfies Assumption 1, the proposed scheme satisfies security for authentication.*

Security against hill-climbing attacks. In the authentication phase, the decryptor checks if the decrypted value of the verification query is equal to 0. 0 implies that the distance does not exceed the threshold. Since non-0 values, which mean that the distance is greater than the threshold, are determined by the random values which the server chooses in every authentication, it looks random from the decryptor. Therefore, the decryptor can only know the authentication result but cannot guess the distance.

Theorem 4. *The proposed scheme satisfies security against hill-climbing attacks.*

5 Conclusion

We have defined the model of secure biometric authentication. A third party called a decryptor is employed in our model in addition to the normal entities in biometric authentication. Also, we have formally defined its security by adapting the security definition provided by Hirano et al. [11] to our model.

In the defined model, we have proposed a scheme that hides biometric features from the server and the decryptor. Moreover, no entity is able to obtain the distance between the enrolled and authenticated biometric features. Therefore, the proposed scheme is resistant to hill-climbing attacks [19]. Since the operations of the decryptor, key generation, and decryption are light enough, they can be implemented by hardware security modules (HSM). By utilizing the modified Elgamal cryptosystem [7], we have showed the security of the proposed scheme under the decisional Diffie-Hellman assumption.

Acknowledgment. We would like to appreciate Anja Lehmann and the anonymous reviewers for their valuable comments. The fourth author is supported by JSPS KAKENHI Grant Number 15K00193.

A Security of Proposed Scheme

In this section, we prove that the proposed scheme satisfies all security properties defined in Sect. 3.2. For simplicity, we show the proof of it where the adversary makes one enrollment query one authentication query. Also we prove three properties, template protection against server, security for authentication, and security against hill-climbing attacks for simpler but less secure version. The simpler version is without Trans. That is, it is different from the version in Sect. 4 in $\mathsf{PIR}_{S,1}$ and $\mathsf{PIR}_{S,2}$ as follows.

$\mathsf{PIR}_{S,1}(PT = (\{A_{i,j}\}_{i,j}, \{B_k\}_k))$:

1. Randomly generates r_F and r_G.
2. For every i, j, and k, computes $A'_{i,j} := A_{i,j}^{r_F}$, $B'_k := B_k^{r_G}$.
3. Encrypts $R \leftarrow \mathsf{Enc}(-r_F - r_G)$.
4. Lets $CH := (\{A'_{i,j}\}_{i,j}, \{B'_k\}_k)$ and $AD := R$ and outputs (CH, AD).

$\mathsf{PIR}_{S,2}(AQ, AD)$:

1. Randomly chooses r' and outputs $VQ := (AQ \cdot AD)^{r'}$.

Template protection against decryptor cannot be proved for the simpler version of the scheme. We explain the reason and that the scheme in Sect. 4 satisfies the property later.

A.1 Proof of Theorem 1 (Template Protection Against Server)

The modified Elgamal cryptosystem satisfies IND-CPA security under the DDH assumption.

Definition 6. *A public key encryption scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is indistinguishable against a chosen plaintext attack (CPA), or shortly IND-CPA-secure, if for any PPT algorithm* $\mathcal{A}$ *it holds that*

$$\mathsf{Adv}^{\mathsf{IND}}_{\mathcal{A}}(\kappa) := \Pr\left[\beta = \beta^* \;\middle|\; \begin{array}{l}(pk, sk) \leftarrow \mathsf{Gen}(1^\kappa); (m_0, m_1, \alpha) \leftarrow \mathcal{A}(pk);\\ \beta \leftarrow \{0,1\}; c \leftarrow \mathsf{Enc}(m_b); b^* \leftarrow \mathcal{A}(c, \alpha);\end{array}\right] \leq negl(\kappa).$$

The proof of Theorem 1 is provided using a sequence of games. We first present three games and after that discuss the relations among them.

Game 0. Game 0 is the game of security for template protection. We describe the game for the case of the proposed scheme as follows:

Setup: On input 1^κ and *param*, $\mathcal{C}$ runs $(pk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$ and chooses $\beta \in \{0,1\}$ at random. $(pk, param)$ is sent to $\mathcal{A}$.

Enrollment: $\mathcal{A}$ chooses and sends two minutiae $((x_0, y_0), t_0)$ and $((x_1, y_1), t_1)$ to $\mathcal{C}$. $\mathcal{C}$ generates two polynomials F_β and G_β according to $((x_\beta, y_\beta), t_\beta)$ as in Eqs. (3) and (4), and expands them as in Eqs. (1) and (2). $\mathcal{C}$ encrypts the coefficients of the polynomials as $A_{i,j} \leftarrow \mathsf{Enc}(a_{i,j})$ and $B_k \leftarrow \mathsf{Enc}(b_k)$ and sends the protected template $PT := (\{A_{i,j}\}_{i,j}, \{B_k\}_k)$ to $\mathcal{A}$.

Authentication: On a request from $\mathcal{A}$, $\mathcal{C}$ chooses r_F and r_G at random and let $A'_{i,j} \leftarrow \mathsf{Enc}(r_F a_{i,j})$ and $B'_k \leftarrow \mathsf{Enc}(r_G b_k)$. On receiving $CH = \left(\{A'_{i,j}\}_{i,j}, \{B'_k\}_k\right)$ from $\mathcal{C}$, $\mathcal{A}$ chooses two minutiae $((x'_0, y'_0), t'_0)$ and $((x'_1, y'_1), t'_1)$ and sends them to $\mathcal{C}$. If for two pairs of minutiae $(((x_0, y_0), t_0), ((x'_0, y'_0), t'_0))$ and $(((x_1, y_1), t_1), ((x'_1, y'_1), t'_1))$, one of them corresponds and the other is not, then $\mathcal{C}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ sends $AQ := \mathsf{Enc}(r_F \cdot F_\beta(x'_\beta, y'_\beta) + r_G \cdot G_\beta(t'_\beta))$ to $\mathcal{A}$.

Output: Finally, $\mathcal{A}$ outputs β^*.

We define the advantage of the adversary in game 0 as $\mathsf{Adv}^0_{\mathcal{A}}(\kappa) := \Pr[\beta = \beta^*] - 1/2$. Apparently $\mathsf{Adv}^0_{\mathcal{A}}(\kappa)$ is the same as $\mathsf{Adv}^{\mathsf{TP,S}}_{\mathcal{A}}(\kappa)$ for the scheme.

Game 1. To make game 1, we only modify the way to generate AQ of game 0. The difference is as follows: Before generating AQ, $\mathcal{C}$ additionally chooses a random bit β' and let $AQ := \mathsf{Enc}(r_F \cdot F_\beta(x'_\beta, y'_\beta) + r_G \cdot G_\beta(t'_\beta))$ if $\beta' = 0$ and $AQ := \mathsf{Enc}(r_F \cdot F_{1-\beta}(x'_{1-\beta}, y'_{1-\beta}) + r_G \cdot G_{1-\beta}(t'_{1-\beta}))$ otherwise. $F_{1-\beta}$ and $G_{1-\beta}$ are the functions generated according to the minutia $((x_{1-\beta}, y_{1-\beta}), t_{1-\beta})$ which was obtained in the enrollment query through Eqs. (3) and (4) with the same random values R_F and R_G that are used in making F_β and G_β.

Obviously, if $\beta' = 0$, game 1 runs exactly the same way as the game 0. We define the advantage of the adversary in game 1 as $\mathsf{Adv}^1_{\mathcal{A}} := \Pr[\beta = \beta^*] - 1/2$.

Game 2. We further modify the method of generating AQ to define game 2 as follows: Before generating AQ, $\mathcal{C}$ additionally chooses two random bit β' and let $AQ := \mathsf{Enc}(r_F \cdot F_\beta(x'_\beta, y'_\beta) + r_G \cdot G_\beta(t'_\beta))$ if $\beta' = 0$ and $AQ := \mathsf{Enc}(r_F \cdot F_{\beta''}(x'_{\beta''}, y'_{\beta''}) + r_G \cdot G_{\beta''}(t'_{\beta''}))$ otherwise.

If $\beta' = 0$ or $\beta'' = \beta$, this game runs exactly the same way as the game 0. We define the advantage of the adversary in game 2 as $\mathsf{Adv}^2_{\mathcal{A}} := \Pr[\beta'' = \beta^*] - 1/2$. We note that this probability is not taken over β but β''.

In order to prove Theorem 1, we show the following lemmas. The first lemma says that the advantage of game 2 is negligible under the IND-CPA security of the underlying cryptosystem.

Lemma 1. *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an IND-CPA secure public key encryption scheme. Then for any PPT algorithm* $\mathcal{A}$ *it holds that* $\mathsf{Adv}^2_{\mathcal{A}}(\kappa) \leq negl(\kappa)$.

Proof. We assume that there exists an PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}^2_{\mathcal{A}}$ is non-negligible and show that PKE is not IND-CPA secure. In order to show it we construct an adversary $\mathcal{A}'$ of IND-CPA game as follows:

First, given an public key pk, $\mathcal{A}'$ input $(pk, param)$ into $\mathcal{A}$. When $\mathcal{A}$ queries two minutiae $((x_0, y_0), t_0)$ and $((x_1, y_1), t_1)$ to $\mathcal{C}$ for enrollment, $\mathcal{A}'$ chooses $\beta \in \{0, 1\}$ randomly, generates two polynomials F_β and G_β, and returns PT as in the description of the game 0.

On an authentication query from $\mathcal{A}$, $\mathcal{A}'$ chooses $\beta' \in \{0, 1\}$ and r_F and r_G randomly. If $\beta' = 0$, $\mathcal{A}'$ returns $AQ := \mathsf{Enc}(r_F \cdot F_\beta(x'_\beta, y'_\beta) + r_G \cdot G_\beta(t'_\beta))$ to $\mathcal{A}$. Otherwise $\mathcal{A}'$ outputs $m_0 := r_F \cdot F_0(x'_0, y'_0) + r_G \cdot G_0(t'_0))$ and $m_1 := r_F \cdot F_1(x'_1, y'_1) + r_G \cdot G_1(t'_1))$ as its challenge. Given a ciphertext c, $\mathcal{A}'$ returns it to $\mathcal{A}$. Finally, $\mathcal{A}'$ outputs $\beta^* \in \{0, 1\}$ that $\mathcal{A}$ outputs.

From the description of IND-CPA game, c is the ciphertext of $m_{\beta''}$ where β'' is a randomly chosen bit. By the assumption, the probability that the output β^* of $\mathcal{A}$ is equal to β'' is non-negligibly greater than 1/2. Thus it is straightforward to see that $\mathsf{Adv}^{\mathsf{IND}}_{\mathcal{A}} \geq negl(\kappa)$.

Next we show that the advantage of game 1 is negligible from IND-CPA security of the underlying cryptosystem.

Lemma 2. *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an IND-CPA secure public key encryption scheme. Then for any PPT algorithm* $\mathcal{A}$*, it holds that* $\mathsf{Adv}^1_{\mathcal{A}}(\kappa) \leq negl(\kappa)$.

To prove this lemma, we use the well-known result that for an IND-CPA secure public key encryption scheme sequences of ciphertexts are also indistinguishable.

Lemma 3. *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public key encryption scheme.* PKE *is n-IND-CPA secure if*

$$\mathsf{Adv}^{n,\mathsf{IND}}_{\mathcal{A}}(\kappa) := \Pr\left[\beta = \beta^* \middle| \begin{array}{l} (pk, sk) \leftarrow \mathsf{Gen}(1^\kappa); \\ ((m_0^1, \ldots, m_0^n), (m_1^1, \ldots, m_1^n), \alpha) \leftarrow \mathcal{A}(pk); \\ \beta \leftarrow \{0,1\}; \\ c^1 \leftarrow \mathsf{Enc}(m_b^1); \cdots; c^n \leftarrow \mathsf{Enc}(m_b^n); \\ b^* \leftarrow \mathcal{A}((c^1, \ldots, c^n), \alpha); \end{array}\right] \leq negl(\kappa)$$

for any PPT algorithm $\mathcal{A}$ *and* $n \in \mathbb{Z}$*. Then* PKE *is IND-CPA secure if and only if* PKE *is n-IND-CPA secure.*

We refer to [9] for the formal proof. Instead of directly showing Lemma 2 from IND-CPA security of the underlying cryptosystem, we prove it from n-IND-CPA security as follows:

Proof. We assume that there exists an PPT algorithm $\mathcal{A}$ such that $\mathsf{Adv}^2_{\mathcal{A}}$ is non-negligible and show that PKE is not n-IND-CPA secure. In order to show it we construct an adversary $\mathcal{A}'$ of n-IND-CPA game as follows:

First, given an public key pk, $\mathcal{A}'$ input $(pk, param)$ into $\mathcal{A}$. When $\mathcal{A}$ queries two minutiae $((x_0, y_0), t_0)$ and $((x_1, y_1), t_1)$ to $\mathcal{C}$ for enrollment, $\mathcal{A}'$ generates four polynomials F_0, F_1, G_0, and G_1 by following Eqs. (3) and (4) where R_F is the same for F_0 and F_1 and R_G is the same for G_0 and G_1. Expanding them as in Eqs. (1) and (2), $\mathcal{A}'$ outputs two tuples of plaintexts where the first one consists of the coefficients of F_0 and G_0, and the second F_1 and G_1. when $\mathcal{A}'$ obtains the ciphertexts of one of the tuples, $\mathcal{A}'$ returns them to $\mathcal{A}$.

On a authentication query from $\mathcal{A}$, $\mathcal{A}'$ chooses $\alpha \in \{0,1\}$ and r_F and r_G randomly. $\mathcal{A}'$ returns $AQ := \mathsf{Enc}(r_F F_\alpha(x'_\alpha, y'_\alpha) + r_G G_\alpha(t'_\alpha))$ to $\mathcal{A}$. Finally, $\mathcal{A}'$ outputs $\beta^* \in \{0,1\}$ that $\mathcal{A}$ outputs.

Now let β be a bit that the challenger chooses. That is, the tuple that $\mathcal{A}'$ obtains are the ciphertexts of the coefficients of F_β and G_β. Then, if $\beta = \alpha$ $AQ := \mathsf{Enc}(r_F \cdot F_\beta(x'_\beta, y'_\beta) + r_G \cdot G_\beta(t'_\beta))$, otherwise $AQ := \mathsf{Enc}(r_F \cdot F_{1-\beta}(x'_{1-\beta}, y'_{1-\beta}) + r_G \cdot G_{1-\beta}(t'_{1-\beta}))$,

From the assumption, the probability that the output β^* of $\mathcal{A}$ is the same as β is non-negligibly greater than 1/2. It implies that the cryptosystem does not satisfy n-IND-CPA security.

Next we show a relation among the games as Lemma 4. By combining Lemmas 1, 2, and 4, Theorem 1 is proved.

Lemma 4. *Let* $\mathsf{Adv}^2_{\mathcal{A}}(\kappa) \leq negl(\kappa)$ *then for any PPT algorithm* $\mathcal{A}'$ *it holds that* $|\mathsf{Adv}^0_{\mathcal{A}'}(\kappa) - \mathsf{Adv}^1_{\mathcal{A}'}(\kappa)| \leq negl(\kappa)$.

Proof. Recall that the differences among the games are only the ways to generate AQ. In game 2, if $\beta = \beta''$, $AQ = \mathsf{Enc}(r_F \cdot F_\beta(x'_\beta, y'_\beta) + r_G \cdot G_\beta(t'_\beta))$. In this case, the behavior of game 2 is the same as that of game 0. For the case where $\beta \neq \beta''$, $AQ = \mathsf{Enc}(r_F F_\beta(x'_\beta, y'_\beta) + r_G G_\beta(t'_\beta))$ if $\beta' = 0$ and $AQ = \mathsf{Enc}(r_F F_{1-\beta}(x'_{1-\beta}, y'_{1-\beta}) + r_G G_{1-\beta}(t'_{1-\beta}))$ otherwise. This is the same as game 1. Thus, if the advantage of adversary in game 2 is negligible, the difference between advantages of adversary in games 0 and 1 is also negligible.

A.2 Proof of Theorem 2 (Template Protection Against Decryptor)

In the simpler version of the scheme, the malicious decryptor who has the secret key can determine the bit β by decrypting the challenge. Therefore, we introduce the transform function Trans. Transformed ciphertexts cannot be decrypted by the secret key. This property is proved under the DDH assumption. The complete proof will be appeared in the final version of this manuscript.

A.3 Proof of Theorem 3 (Security for Authentication)

Here, we prove Theorem 3 under Assumption 1. Note that this assumption holds for the modified Elgamal cryptosystem under the computational Diffie-Hellman (CDH) assumption.

Lemma 5. *Under the CDH assumption, Assumption 1 holds for the modified Elgamal cryptosystem.*

Proof. Let $\mathcal{A}$ be a PPT algorithm that does not satisfy Assumption 1 for the modified Elgamal cryptosystem, and we show that there exists a PPT algorithm $\mathcal{A}'$ that breaks the CDH assumption. We construct $\mathcal{A}'$ as follows:

For tuple (G, p, g, g_1, g_2) where $g_1 = g^a$ and $g_2 = g^b$ for some $a, b \in \mathbb{Z}_p$ that is input into $\mathcal{A}'$, let $h_0 := g_1 = g^a$, $h_1 := g = h_0^{1/a}$ and $h_2 := g_2 = h_0^{b/a}$. $\mathcal{A}'$ chooses $x, r, r' \in \mathbb{Z}_p$, and let $y := h_0^x$. Then, $\mathcal{A}'$ inputs tuple $((p, g, y), (h_0^r, h_1 y^r), (h_0^{r'}, h_2 y^{r'}))$ into $\mathcal{A}$. Here, for the key pair $pk = (p, h_0, y)$ and $sk = x$, $(h_0^r, h_1 y^r)$ and $(h_0^{r'}, h_2 y^{r'})$ are ciphertexts of $1/a$ and b/a, respectively. Therefore, $\mathcal{A}$ outputs a ciphertext $(c_1, c_2) := (h_0^{r''}, h_0^b y^{r''})$ of b for some $r'' \in \mathbb{Z}_p$ with non-negligible probability. Now it is easy to see which $\mathcal{A}'$ outputs $c_2 / c_1^x = h_0^b = g^{ab}$, which breaks the CDH assumption.

Now we prove Theorem 3.

Proof. We first describe the game of security for authentication for the proposed scheme between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows:

Setup: On input of 1^κ and *param*, $\mathcal{C}$ runs $(pk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$ and chooses bit β at random. $(pk, param)$ is sent to $\mathcal{A}$.

Enrollment: On request from $\mathcal{A}$, $\mathcal{C}$ chooses a minutia $((x, y), t)$ and computes two polynomials F and G in accordance with it as in Eqs. (1), (2), (3), and (4). Then, $\mathcal{C}$ encrypts the coefficients of the polynomials as $A_{i,j} \leftarrow \mathsf{Enc}(a_{i,j})$ and $B_k \leftarrow \mathsf{Enc}(b_k)$ and returns the protected template $PT := (\{A_{i,j}\}_{i,j}, \{B_k\}_k)$ to $\mathcal{A}$.

Authentication: On request from $\mathcal{A}$, $\mathcal{C}$ chooses r_F and r_G at random and computes $A'_{i,j} \leftarrow \mathsf{Enc}(r_F \cdot a_{i,j})$ and $B'_k \leftarrow \mathsf{Enc}(r_G \cdot b_k)$. Then, $\mathcal{C}$ lets $CH := \left(\{A'_{i,j}\}_{i,j}, \{B'_k\}_k\right)$, $AQ := \mathsf{Enc}(-r_F - r_G)$ and sends back tuple $(CH, AQ, Accept)$ to $\mathcal{A}$.

Output: On request from $\mathcal{A}$, $\mathcal{C}$ chooses r'_F and r'_G randomly and sends back $CH^* := \left(\{A''_{i,j}\}_{i,j}, \{B''_k\}_k\right)$ to $\mathcal{A}$ where $A''_{i,j} \leftarrow \mathsf{Enc}(r'_F \cdot a_{i,j})$ and $B''_k \leftarrow \mathsf{Enc}(r'_G \cdot b_k)$. Finally, $\mathcal{A}$ outputs AQ^*.

The advantage of $\mathcal{A}$ that can be described as $\mathsf{Adv}^{\mathsf{Auth}}_{\mathcal{A}}(\kappa) := \Pr[\mathsf{Dec}(AQ^*) = r'_F + r'_G]$ is assumed to be non-negligible. We construct algorithm $\mathcal{A}'$ that breaks Assumption 1 for the underlying cryptosystem as follows:

On input of $(pk, \mathsf{Enc}(s), \mathsf{Enc}(st))$, $\mathcal{A}'$ chooses minutia $((x, y), t)$. For the location of the minutia, $\mathcal{A}'$ computes polynomial F as in Eqs. (1) and (3) and encrypts the coefficients of the polynomials as $A_{i,j} \leftarrow \mathsf{Enc}(a_{i,j})$. Also, $\mathcal{A}'$ generates the polynomial G as in Eqs. (2) and (4) where $R_G := s$. That is, if we write polynomial $\tilde{G}$ as $\tilde{G}(T) = \prod_{\ell=-\Delta_t}^{\Delta_t} \{(T - t) - \ell\} + 1 = \sum_k \tilde{b}_k \cdot T^k$, it holds that $b_k = s \cdot \tilde{b}_k$ for any k. Therefore, $\mathcal{A}'$ encrypts the coefficients as $B_k := \mathsf{Enc}(s)^{\tilde{b}_k} = \mathsf{Enc}(s \cdot \tilde{b}_k) = \mathsf{Enc}(b_k)$. Then, $\mathcal{A}'$ sends the protected template $PT := (\{A_{i,j}\}_{i,j}, \{B_k\}_k)$ to $\mathcal{A}$.

On authentication query from $\mathcal{A}$, $\mathcal{A}'$ chooses r_F and r_G at random and computes $A'_{i,j} \leftarrow \mathsf{Enc}(r_F \cdot a_{i,j})$ and $B'_k := \mathsf{Enc}(s)^{r_G \cdot \tilde{b}_k} = \mathsf{Enc}(r_G \cdot b_k)$. Then, $\mathcal{C}$ lets $CH := \left(\{A'_{i,j}\}_{i,j}, \{B'_k\}_k\right)$, $AQ := \mathsf{Enc}(-r_F - r_G)$ and sends back tuple $(CH, AQ, Accept)$ to $\mathcal{A}$.

On output query from $\mathcal{A}$, $\mathcal{A}'$ chooses r'_F randomly and computes $A''_{i,j} \leftarrow \mathsf{Enc}(r'_F \cdot a_{i,j})$ and $B''_k := \mathsf{Enc}(st)^{\tilde{b}_k} = \mathsf{Enc}(t \cdot b_k)$. Given $CH^* := \left(\{A''_{i,j}\}_{i,j}, \{B''_k\}_k\right)$, $\mathcal{A}$ outputs AQ^*. Finally, $\mathcal{A}'$ outputs $t^* := AQ^* \cdot \mathsf{Enc}(-r'_F)$.

Note that the random values r_G and r'_G that the challenger chooses in the game for authentication is set to be s and t, respectively, by $\mathcal{A}'$. Therefore, from the assumption, AQ^* is a ciphertext of $r'_F + r'_G = r'_F + t$ with non-negligible probability. This means that the output of $\mathcal{A}'$ is a ciphertext of t with non-negligible probability. That is, $\mathcal{A}'$ breaks Assumption 1.

A.4 Proof of Theorem 4 (Security Against Hill-Climbing Attacks)

Proof. We first describe the game of security against hill-climbing attacks for the proposed scheme between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows:

Setup: On input of 1^κ and *param*, $\mathcal{C}$ runs $(pk, sk) \leftarrow \mathsf{Gen}(1^\kappa)$ and chooses bit β at random. $((pk, param), sk)$ is sent to $\mathcal{A}$.

Enrollment: On request from $\mathcal{A}$ for a minutia $((x,y),t)$, $\mathcal{C}$ computes two polynomials F and G in accordance with it as in Eqs. (1)–(4). $\mathcal{C}$ encrypts the coefficients of the polynomials as $A_{i,j} \leftarrow \mathsf{Enc}(a_{i,j})$ and $B_k \leftarrow \mathsf{Enc}(b_k)$ and returns protected template $PT := (\{A_{i,j}\}_{i,j}, \{B_k\}_k)$ to $\mathcal{A}$.

Authentication: On request from $\mathcal{A}$, $\mathcal{C}$ chooses r_F and r_G at random and lets $A'_{i,j} \leftarrow \mathsf{Enc}(r_F \cdot a_{i,j})$ and $B'_k \leftarrow \mathsf{Enc}(r_G \cdot b_k)$. On receiving $CH = \left(\{A'_{i,j}\}_{i,j}, \{B'_k\}_k\right)$ from $\mathcal{C}$, $\mathcal{A}$ sends back AQ. Then, $\mathcal{C}$ chooses r at random and computes $VQ := (AQ \cdot \mathsf{Enc}(-r_F - r_G))^r$ and sends it to $\mathcal{A}$.

Output: $\mathcal{A}$ requests $\mathcal{C}$ for two minutiae $((x'_0, y'_0), t'_0)$ and $((x'_1, y'_1), t'_1)$. If for two pairs of minutiae $(((x,y),t),((x'_0,y'_0),t'_0))$ and $(((x,y),t),((x'_1,y'_1),t'_1))$, one of them corresponds and the other does not, then $\mathcal{C}$ returns $\perp$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ chooses bit β and r'_F, r'_G, and r' randomly and returns $VQ^* \leftarrow \mathsf{Enc}(r(r'_F \cdot F(x'_\beta, y'_\beta) + r'_G \cdot G(t'_\beta) - r'_F - r'_G))$ to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs β^*.

The advantage of adversary $\mathcal{A}$ is defined as $\mathsf{Adv}^{\mathsf{Dist}}_{\mathcal{A}}(\kappa) := \Pr[\beta = \beta^*] - 1/2$. The adversary may obtain information related to β only in the output phase. If the correspondences of two pairs of minutiae are different, $\mathcal{A}$ does not obtain any information in the output phase. Also, if both pairs correspond, VQ^* that $\mathcal{A}$ obtains in the output phase is the ciphertext of 0 no matter which bit is chosen as β. In these two cases, we can say that $\mathcal{A}$ does not obtain any information on β. Thus, $\mathcal{A}$ has no chance to distinguish the bit β. The remaining case is where both pairs do not correspond. In this case, although $\mathcal{A}$ who has the secret key sk obtains VQ^*, which is related to β, it is randomized by a new random value r that is used only once. Therefore, $\mathcal{A}$ cannot guess β in this case as well.

References

1. Barbosa, M., Brouard, T., Cauchie, S., de Sousa, S.M.: Secure biometric authentication with improved accuracy. In: Mu, Y., Susilo, W., Seberry, J. (eds.) ACISP 2008. LNCS, vol. 5107, pp. 21–36. Springer, Heidelberg (2008)
2. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
3. Bringer, J., Chabanne, H.: An authentication protocol with encrypted biometric data. In: Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 109–124. Springer, Heidelberg (2008)
4. Bringer, J., Chabanne, H., Izabachène, M., Pointcheval, D., Tang, Q., Zimmer, S.: An application of the goldwasser-micali cryptosystem to biometric authentication. In: Pieprzyk, J., Ghodosi, H., Dawson, E. (eds.) ACISP 2007. LNCS, vol. 4586, pp. 96–106. Springer, Heidelberg (2007)
5. Campisi, P. (ed.): Security and Privacy in Biometrics. Springer, London (2013)
6. Dodis, Y., Reyzin, L., Smith, A.: Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 523–540. Springer, Heidelberg (2004)
7. Elgamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theor. **31**(4), 469–472 (1985). IEEE

8. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient private matching and set intersection. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 1–19. Springer, Heidelberg (2004)
9. Goldreich, O.: The Foundations of Cryptography - Basic Applications, vol. 2. Cambridge University Press, Cambridge (2004)
10. Hattori, M., Matsuda, N., Ito, T., Shibata, Y., Takashima, K., Yoneda, T.: Provably-secure cancelable biometrics using 2-DNF evaluation. J. Inf. Process. **20**(2), 496–507 (2012). IPSJ
11. Hirano, T., Hattori, M., Ito, T., Matsuda, N.: Cryptographically-secure and efficient remote cancelable biometrics based on public-key homomorphic encryption. In: Sakiyama, K., Terada, M. (eds.) IWSEC 2013. LNCS, vol. 8231, pp. 183–200. Springer, Heidelberg (2013)
12. Juels, A., Sudan, M.: A fuzzy vault scheme. In: ISIT 2002, p. 408. IEEE (2002)
13. Juels, A., Wattenberg, M.: A fuzzy commitment scheme. In: CCS 1999, pp. 28–36. ACM (1999)
14. Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition, 2nd edn. Springer Publishing Company, Incorporated, London (2009)
15. Nandakumar, K., Jain, A.K., Pankanti, S.: Fingerprint-based fuzzy vault: implementation and performance. IEEE Trans. Inf. Forensics Secur. **2**(4), 744–757 (2007). IEEE
16. Rathgeb, C., Uhl, A.: A survey on biometric cryptosystems and cancelable biometrics. EURASIP J. Inf. Secur. **2011**(1), 3 (2011). Springer International Publishing AG
17. Shahandashti, S.F., Safavi-Naini, R., Ogunbona, P.: Private fingerprint matching. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP 2012. LNCS, vol. 7372, pp. 426–433. Springer, Heidelberg (2012)
18. Simoens, K., Bringer, J., Chabanne, H., Seys, S.: A framework for analyzing template security and privacy in biometric authentication systems. IEEE Trans. Inf. Forensics Secur. **7**(2), 833–841 (2012). IEEE
19. Uludag, U., Jain, A.K.: Attacks on biometric systems: a case study in fingerprints. In: Delp, E.J., Wong, P.W. (eds.) Proceedings of SPIE 5306, pp. 622–633. SPIE (2004)