Recommended Paper

Provably-Secure Cancelable Biometrics Using 2-DNF Evaluation

Mitsuhiro Hattori[1,a)] Nori Matsuda[1] Takashi Ito[1]
Yoichi Shibata[1] Katsuyuki Takashima[1] Takeshi Yoneda[1]

Received: July 8, 2011, Accepted: January 13, 2012

Abstract: Biometric authentication has been attracting much attention because it is more user-friendly than other authentication methods such as password-based and token-based authentications. However, it intrinsically comprises problems of privacy and revocability. To address these issues, new techniques called cancelable biometrics have been proposed and their properties have been analyzed extensively. Nevertheless, only a few considered provable security, and provably secure schemes known to date had to sacrifice user-friendliness because users have to carry tokens so that they can securely access their secret keys. In this paper, we propose two cancelable biometric protocols each of which is provably secure and requires no secret key access of users. We use as an underlying component the Boneh-Goh-Nissim cryptosystem proposed in TCC 2005 and the Okamoto-Takashima cryptosystem proposed in Pairing 2008 in order to evaluate 2-DNF (disjunctive normal form) predicate on encrypted feature vectors. We define a security model in a semi-honest manner and give a formal proof which shows that our protocols are secure in that model. The revocation process of our protocols can be seen as a new way of utilizing the veiled property of the underlying cryptosystems, which may be of independent interest.

Keywords: biometric authentication, cancelable biometrics, provable security, homomorphic encryption

1. Introduction

User authentication has become an indispensable part of procedures in many application systems and on-line services. They are used for granting only legitimate users an access to the services. Several kinds of authentication methods including password-based authentication and token-based authentication have been used in real systems. Among them, biometric authentication has been attracting much attention in recent years because it is user-friendly (i.e., it does not force users to remember long passwords and to carry tokens). However, it intrinsically comprises problems of privacy and revocability. As for privacy, biometric data such as fingerprint, vein, face, iris, palm print, and retina can be regarded as sensitive information because it represents physical characteristics of individuals. Indeed, it is known that retina patterns may provide medical information about diabetes [26] and fingerprints may tell genetic information [32]. As for revocability, it is infeasible to revoke biometric data because only a limited number of biometric data can be used for substitution (suppose the case of iris; there are only two patterns). Also, once templates of biometric data stored on an authentication server are leaked out, they might be used in another authentication system for disguise. This will allow illegitimate users to gain illegal access to the services.

To address these issues, template protection schemes [20] have been considered extensively.

1.1 Related Work

Template protection schemes can be classified into two broad categories: biometric cryptosystem and feature transformation. A biometric cryptosystem such as a fuzzy commitment scheme [23], a fuzzy vault scheme [22], and a fuzzy extractor scheme [11] incorporates error-correcting codes to absorb intra-user deviation and appends noisy data called chaff to hide biometric features. Accuracy of authentication is therefore degraded by the chaff, and the security and the accuracy become a trade-off.

The feature transformation approach comprises several different schemes. Among them, cancelable biometrics [8], [19], [29], [34], [36], [37], [38], [39], [40] is the most successful one. Roughly speaking, cancelable biometrics uses a morphing function to hide information on biometric features. In enrollment, a biometric feature is morphed randomly depending on a user-specific parameter P and such morphed data is stored on the authentication server as a template. In authentication, a biometric feature for authentication is morphed using the same parameter P and the morphed one is compared to the template on the morphed domain. Cancelable biometrics was first proposed by Ratha [34] for fingerprint authentication. After that, several schemes were proposed for various modalities [8], [29], [36], [39], [40]. Those schemes, however, had no rigorous security analysis or security

[1] Mitsubishi Electric Corporation, Kamakura, Kanagawa 247–8501, Japan
[a)] Hattori.Mitsuhiro@eb.MitsubishiElectric.co.jp

The preliminary version of the paper was published at Multimeda, Distributed, Cooperative, and Mobile Symposium (DICOMO 2010), July 2010. The paper was recommended to be submitted to IPSJ journal via a chair of Computer Security Working Group (CSEC).

proof.

Provably secure cancelable biometrics was proposed by Takahashi and Hirata [19], [37], [38]. In their schemes, correlation-invariant random filtering is used as a morphing function, which is basically a kind of two-dimensional discrete Fourier transform (DFT) over a Galois field with a random parameter P. Due to the property of DFT and its inverse, the influence of the random filter is cancelled and only the desired metrics (the cyclic cross-correlation in their schemes) is recovered perfectly and revealed to the authentication server. In this way, both the accuracy and the provable security are satisfied. Moreover, their schemes satisfy both diversity (the property that cross-matching across databases is infeasible) and revocability. All of these four properties are desirable in template protection schemes [20], thus their schemes seem satisfiable.

However, user-friendliness may be degraded. In their schemes, the parameter P must be kept secret and only the legitimate user should have access to P in both enrollment and authentication. Therefore, users have to carry a tamper-proof token whenever they are authenticated. This may be alleviated slightly by building a proxy server which stores P and controls the user's access on it, but in that case, another user authentication process (e.g., password-based one) is required additionally.

In another line of research, Bringer and Chabanne proposed a biometric authentication protocol which satisfies both provable security and user-friendliness [6]. Their protocol, however, entails a high computational complexity in authentication, which could be an obstacle to practical use.

Other template protection approaches are known (e.g., ZeroBio [35]), but none of them satisfies all of the four properties as well as user-friendliness.

1.2 Our Contribution

In this paper, we propose two cancelable biometric protocols that satisfy all of the four properties as well as user-friendliness. Our protocols employ feature vectors of fixed length as a biometric feature and use the squared Euclidean distance of two vectors as similarity metrics. We define a security model in a semi-honest manner and give a formal proof which shows that our protocols are secure in that model. Also, we evaluate the data size and the computational cost of our protocols and reveal that both protocols are reasonable when biometric feature vectors are well-designed and thus the vector size is relatively short.

As an underlying component of our protocols, we use two public-key cryptosystems: the Boneh-Goh-Nissim cryptosystem [3] and the Okamoto-Takashima cryptosystem [30]. The Boneh-Goh-Nissim cryptosystem is an encryption scheme which enables 2-DNF (disjunctive normal form) evaluation on ciphertexts. The Okamoto-Takashima cryptosystem is not in nature 2-DNF homomorphic encryption, but it is possible to evaluate 2-DNF predicate by suitably modifying the algorithms. We employ these cryptosystems as a component of our protocols in order to achieve the "one-shot" authentication in a provably-secure manner, where one-shot implies that users have only to send their encrypted biometric data once, and leave all the remaining procedures to the authentication server [*1].

As we mentioned in the previous subsection, users should be free from possessing secret keys for the sake of user-friendliness. With this in mind, we pose in our system model a decryptor who possesses secret keys and decrypts ciphertexts. However, this may cause security breaches, because the decryptor may recover the original biometric data. Therefore, we carefully designed our protocol so that the decryptor is able to recover only the predefined metrics (squared Euclidean distance in our protocols). We show our techniques in the following sections.

From another point of view, our protocols can be seen as a new way of utilizing the veiled property of the underlying cryptosystems. In the revocation process of our protocols, ciphertexts and public keys are updated simultaneously without changing plaintexts nor secret keys. This is because the underlying cryptosystems have a property that, given a public key PK and a ciphertext C, one can find a new pair (PK', C') such that the secret key stays the same but C' encrypts the same plaintext under PK'. Although this veiled property seems useless in the original cryptosystems, it plays an important role in our protocols. We believe that our protocols are the first which unveiled and utilized the property.

1.3 Organization of the Paper

The rest of the paper is organized as follows. Section 2 describes the system model and desirable properties of cancelable biometric authentication systems. Our first protocol is given in Section 3 and the second one is given in Section 4. Section 5 evaluates our two protocols and discuss our contribution. Section 6 concludes the paper.

2. Preliminaries

In this section, we first describe our model of a biometric authentication system. Then we review the desirable properties of cancelable biometrics which were formalized by Jain, Nandakumar, and Nagar [20]. Finally, we give several notations used throughout the paper.

2.1 System Model

2.1.1 Entities

In our biometric authentication system, there are three kinds of entities: users $\mathcal{U}_1, \ldots, \mathcal{U}_N$, an authentication server $\mathcal{S}$, and a decryptor $\mathcal{D}$. We assume that $\mathcal{S}$ and $\mathcal{D}$ do not collude.

2.1.2 Processes

Our system comprises the following four processes:

The setup process: In this process, $\mathcal{D}$ generates a public key and a corresponding secret key.

The enrollment process: In this process, users register their biometric features into $\mathcal{S}$ as a template.

The authentication process: This process contains three consecutive procedures. Firstly, users send their authentication data to $\mathcal{S}$. Secondly, $\mathcal{S}$ computes (encrypted) similarity met-

[*1] The one-shot authentication cannot be achieved by using conventional homomorphic encryption schemes such as the ElGamal cryptosystem [12] or the Paillier cryptosystem [31], because a 2-DNF predicate cannot be evaluated without plaintexts and thus a two-way interaction is needed between a user and the authentication server.

rics and sends it to $\mathcal{D}$. Finally, $\mathcal{D}$ decrypts it, compares the result to the predefined threshold, and returns a binary result (accept or reject) to $\mathcal{S}$.

The revocation process: In this process, $\mathcal{S}$ revokes a template of a user and replace it with a new one.

2.1.3 Biometric Features and Metrics

Many kinds of biometric features and metrics are used in real systems to measure the similarity of biometrics. Most of such features and metrics are fully customized to each modality in order to improve the accuracy (see the seminal book [5] for details). In this paper, however, we employ generic feature vectors of fixed length as a biometric feature and use the squared Euclidean distance of two vectors as similarity metrics. This may seem a generic-but-useless approach. However, it has already been adopted in several modalities (e.g., as FingerCode in fingerprint recognition [21], as IrisCode in iris recognition [9], and as Competitive Code in palm print recognition [27]), and even in practical use [1], [10].

We assume that each feature vector consists of D elements of integers; e.g., $\vec{x} = (x_1, \ldots, x_D) \in \mathbb{Z}_n^D$. The squared Euclidean distance of two vectors $\vec{x}$ and $\vec{y}$ is defined by $d_{E^2}(\vec{x}, \vec{y}) = \sum_{i=1}^{D}(x_i - y_i)^2$, and it will be compared to the predefined threshold θ.

Note that both of our protocols can be applied straightforwardly to the binary feature vectors and the Hamming distance metrics.

2.2 Desirable Properties of Cancelable Biometrics

2.2.1 General Requirements

Cancelable biometrics are required to satisfy the following four properties [20].

Accuracy: In general, an error would occur in evaluating the similarity of the feature vectors in the cancelable biometric system, and the accuracy (false acceptance rate and false rejection rate) may be degraded from that of the original biometric system. It is important that the degree of accuracy degradation is small enough.

Diversity: It should be possible to produce a very large number of cancelable templates (to be used in different applications) from the same biometric feature. Furthermore, it should be impossible to match cancelable templates from different applications.

Revocability: It should be straightforward to revoke a compromised template and reissue a new one based on the same biometric feature.

Security: It should be infeasible to obtain any partial information on users' feature vectors from the data that appears in the protocol.

2.2.2 The Security Requirements and Assumptions of Our System

In our system, the general security requirement can further be classified into the following three requirements:

[Sec-1] **Security against the authentication server $\mathcal{S}$:** It is required that $\mathcal{S}$ cannot obtain extra information other than the binary result (accept or reject) of authentication.

[Sec-2] **Security against the decryptor $\mathcal{D}$:** It is required that $\mathcal{D}$ cannot obtain extra information other than the squared Euclidean distance of two feature vectors.

[Sec-3] **Security against eavesdroppers $\mathcal{E}$:** It is required that eavesdroppers $\mathcal{E}$ cannot obtain extra information other than the binary result (accept or reject) of authentication

In requirement [Sec-2], $\mathcal{D}$ is allowed to know the squared Euclidean distance because it is the indispensable sole input for authentication. Indeed, other cancelable biometrics allow disclosure of the similarity metrics; see requirement (ii-d) in Ref. [38] for example. A more specific definition of these security requirements will be given in Section 3.4.

We note here that, as with other privacy-preserving biometric authentication protocols [7], [13], [24], [25] and cancelable biometrics [34], [38], our focus is solely on privacy aspects of biometrics; thus authenticity aspects are outside of our scope. We therefore do not consider attacks on the authenticity, such as the impersonation attack and the replay attack, in this paper. Countermeasures to these attacks have been proposed by Hirano et al. [18] in which a challenge-response technique is applied for preventing impersonation.

Moreover, we consider the security of our system in the semi-honest model; thus we assume that both the server $\mathcal{S}$ and the decryptor $\mathcal{D}$ follow the protocols faithfully but will try to gain as much information as possible about the biometric features of the users. Also, we consider only passive eavesdroppers, i.e., those who do not interact with other entities but just observe all the data transmissions and analyze the biometric features of the users from the data. We pose these assumptions because general techniques have already been known [16] for converting any cryptographic protocols secure against passive adversaries in the semi-honest model into ones secure against active adversaries in the malicious model.

2.3 Notations

When S is a finite set, $y \xleftarrow{\mathsf{U}} S$ denotes that y is uniformly selected from S. When A is a random variable or distribution, $y \xleftarrow{\mathsf{R}} A$ denotes that y is randomly selected from A according to its distribution, and $A \to y$ denotes the event that A outputs y according to its distribution. $y := x$ denotes that x is substituted into y.

3. A Protocol Based on the Boneh-Goh-Nissim Cryptosystem

We describe in this section one of our protocols which is based on the Boneh-Goh-Nissim cryptosystem proposed in TCC 2005 [3]. Our protocol satisfies all the desirable properties: accuracy, diversity, revocability, and security.

3.1 Cryptographic Primitives

Let q_1 and q_2 be prime numbers and $n = q_1 q_2$. Let $\mathbb{G}$ and $\mathbb{G}_T$ be cyclic multiplicative groups of order n. If a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ satisfies the following two conditions, e is called a cryptographic bilinear map:

- For all $u, v \in \mathbb{G}$ and for all $a, b \in \mathbb{Z}$, $e(u^a, v^b) = e(u, v)^{ab}$.

- There exists a generator $g \in \mathbb{G}$ such that $e(g,g)$ is a generator of $\mathbb{G}_T$.

We can construct an algorithm $\mathcal{G}$ that takes as input a security parameter 1^λ and outputs a description of composite-order bilinear pairing groups $(q_1, q_2, \mathbb{G}, \mathbb{G}_T, e)$. Details can be found in Ref. [3].

Let $\mathbb{G}_{q_1}$ and $\mathbb{G}_{q_2}$ be two subgroups of $\mathbb{G}$ of order q_1 and q_2, respectively. It holds that for all $x_{q_1} \in \mathbb{G}_{q_1}$ and for all $y_{q_2} \in \mathbb{G}_{q_2}$,

$$e(x_{q_1}, y_{q_2}) = 1$$

where 1 is the identity element of $\mathbb{G}_T$.

As in the original Boneh-Goh-Nissim cryptosystem [3], our protocol is based on the following subgroup decision (SGD) assumption [*2].

Definition 1. *Suppose $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a cryptographic bilinear map. Let $n := q_1 q_2$, $g \xleftarrow{\cup} \mathbb{G}$, $g_{q_1} \xleftarrow{\cup} \mathbb{G}_{q_1}$, $Z := (n, g, g_{q_1}, e)$, $T \xleftarrow{\cup} \mathbb{G}_{q_1}$, and $R \xleftarrow{\cup} \mathbb{G}$. We say that the SGD assumption holds if for any PPT (probabilistic polynomial time) algorithm $\mathcal{A}$, $|\Pr[\mathcal{A}(Z,T) \to 1] - \Pr[\mathcal{A}(Z,R) \to 1]|$ is negligible.*

3.2 Protocol Description

We show our construction of four processes.

3.2.1 Setup Process

The setup process is as follows.

(S-1) $\mathcal{D}$ invokes $(q_1, q_2, \mathbb{G}, \mathbb{G}_T, e) \xleftarrow{R} \mathcal{G}(1^\lambda)$.

(S-2) $\mathcal{D}$ picks $\{g_k, u_k\}_{k=1,\ldots,N} \xleftarrow{\cup} \mathbb{G}$, where N is the number of users, and sets $\{h_k := u_k^{q_2},\ g_{T,k} := e(g_k, g_k)\}_{k=1,\ldots,N}$.

(S-3) $\mathcal{D}$ sets $PK := (n, \mathbb{G}, \mathbb{G}_T, e)$, $\{PK_k := (g_k, h_k, g_{T,k})\}_{k=1,\ldots,N}$, and $SK := q_1$, and makes PK and $\{PK_k\}_{k=1,\ldots,N}$ publicly available.

PK is the public parameter of the system and PK_k is the public key for the user $\mathcal{U}_k$. SK is the secret key of the system and thus stored secretly in $\mathcal{D}$.

3.2.2 Enrollment Process

The enrollment process of the user $\mathcal{U}_k$ is as follows.

(E-1) $\mathcal{U}_k$ picks $r_1, \ldots, r_D \xleftarrow{\cup} \mathbb{Z}_n$ and encrypts his biometric feature vector $\vec{x} = (x_1, \ldots, x_D) \in \mathbb{Z}_n^D$ under the public key PK_k as follows:

$$c_{\vec{x},1} := g_k^{x_1} h_k^{r_1},\ \ldots,\ c_{\vec{x},D} := g_k^{x_D} h_k^{r_D}.$$

(E-2) $\mathcal{U}_k$ sends $E_{PK_k}(\vec{x}) = (c_{\vec{x},1}, \ldots, c_{\vec{x},D})$ to $\mathcal{S}$.

(E-3) $\mathcal{S}$ stores a tuple $(k, PK_k, E_{PK_k}(\vec{x}))$ as a template.

3.2.3 Authentication Process

The authentication process of user $\mathcal{U}_k$ is as follows.

(A-1) $\mathcal{U}_k$ picks $s_1, \ldots, s_D \xleftarrow{\cup} \mathbb{Z}_n$ and encrypts his biometric feature vector $\vec{y} = (y_1, \ldots, y_D) \in \mathbb{Z}_n^D$ under the public key PK_k as follows:

$$c_{\vec{y},1} := g_k^{y_1} h_k^{s_1},\ \ldots,\ c_{\vec{y},D} := g_k^{y_D} h_k^{s_D}.$$

(A-2) $\mathcal{U}_k$ sends $E_{PK_k}(\vec{y}) = (c_{\vec{y},1}, \ldots, c_{\vec{y},D})$ to $\mathcal{S}$.

(A-3) $\mathcal{S}$ picks $u_1, u_2 \xleftarrow{\cup} \mathbb{Z}_n$, takes the corresponding tuple $(k, PK_k, E_{PK_k}(\vec{x}))$, and computes the *encrypted distance* Δ_k by

$$\Delta_k := e(g_k, h_k)^{u_1} e(h_k, h_k)^{u_2} \cdot \prod_{i=1}^{D} e(c_{\vec{x},i},\ c_{\vec{x},i}) \cdot e(c_{\vec{y},i},\ c_{\vec{y},i}) \cdot e(c_{\vec{x},i},\ c_{\vec{y},i})^{-2}.$$

(A-4) $\mathcal{S}$ sends the encrypted distance Δ_k to $\mathcal{D}$.

(A-5) $\mathcal{D}$ sets $\hat{g}_{T,k} := g_{T,k}^{q_1}$ and computes the discrete logarithm of $\Delta_k^{q_1}$ to the base $\hat{g}_{T,k}$, that is,

$$d_{E^2}(\vec{x}, \vec{y}) := \mathsf{DLog}_{\hat{g}_{T,k}} \Delta_k^{q_1},$$

where $d_{E^2}(\vec{x}, \vec{y})$ is the squared Euclidean distance of $\vec{x}$ and $\vec{y}$.

(A-6) $\mathcal{D}$ compares $d_{E^2}(\vec{x}, \vec{y})$ to the predefined threshold θ. If $d_{E^2}(\vec{x}, \vec{y}) < \theta$, $\mathcal{D}$ returns accept to $\mathcal{S}$; otherwise, $\mathcal{D}$ returns reject to $\mathcal{S}$.

3.2.4 Revocation Process

Suppose that the tuple

$$(k, PK_k = (g_k, h_k, g_{T,k}), E_{PK_k}(\vec{x}) = (c_{\vec{x},1}, \ldots, c_{\vec{x},D}))$$

leaked out from $\mathcal{S}$. The revocation process is as follows.

(R-1) $\mathcal{S}$ picks $\delta \xleftarrow{\cup} \mathbb{Z}_n$ and computes

$$g'_k := g_k^\delta,\ h'_k := h_k^\delta,\ g'_{T,k} := g_{T,k}^{\delta^2},$$
$$c'_{\vec{x},1} := c_{\vec{x},1}^\delta,\ \ldots,\ c'_{\vec{x},D} := c_{\vec{x},D}^\delta.$$

(R-2) $\mathcal{S}$ stores a new tuple $(k, PK'_k = (g'_k, h'_k, g'_{T,k}), E_{PK'_k}(\vec{x}) = (c'_{\vec{x},1}, \ldots, c'_{\vec{x},D}))$.

(R-3) $\mathcal{S}$ makes PK'_k publicly available.

3.3 Correctness

3.3.1 Authentication

We show that $\mathsf{DLog}_{\hat{g}_{T,k}} \Delta_k^{q_1}$ is exactly the squared Euclidean distance of two vectors $\vec{x}$ and $\vec{y}$.

$\Delta_k^{q_1}$ can be written as

$$\begin{aligned}
&\Delta_k^{q_1} \\
&= \left(e(g_k, h_k)^{u_1} e(h_k, h_k)^{u_2} \cdot \prod_{i=1}^{D} e(c_{\vec{x},i},\ c_{\vec{x},i}) e(c_{\vec{y},i},\ c_{\vec{y},i}) e(c_{\vec{x},i},\ c_{\vec{y},i})^{-2} \right)^{q_1} \\
&= e(g_k, h_k)^{u_1 q_1} e(h_k, h_k)^{u_2 q_1} \prod_{i=1}^{D} e(c_{\vec{x},i}, c_{\vec{x},i})^{q_1} e(c_{\vec{y},i}, c_{\vec{y},i})^{q_1} e(c_{\vec{x},i}, c_{\vec{y},i})^{-2q_1} \\
&= \prod_{i=1}^{D} e(g_k^{x_i} h_k^{r_i},\ g_k^{x_i} h_k^{r_i})^{q_1} \cdot e(g_k^{y_i} h_k^{s_i},\ g_k^{y_i} h_k^{s_i})^{q_1} \cdot e(g_k^{x_i} h_k^{r_i},\ g_k^{y_i} h_k^{s_i})^{-2q_1},
\end{aligned}$$

where the last equation holds since $e(g_k, h_k)^{q_1} = e(g_k, u_k)^{q_1 q_2} = 1$ and $e(h_k, h_k)^{q_1} = e(h_k, u_k)^{q_1 q_2} = 1$.

In the equation above, the first term can be written as

$$\begin{aligned}
&e(g_k^{x_i} h_k^{r_i},\ g_k^{x_i} h_k^{r_i})^{q_1} \\
&\quad = \left\{ e(g_k^{x_i},\ g_k^{x_i}) \cdot e(g_k^{x_i},\ h_k^{r_i}) \cdot e(h_k^{r_i},\ g_k^{x_i}) \cdot e(h_k^{r_i},\ h_k^{r_i}) \right\}^{q_1} \\
&\quad = e(g_k, g_k)^{x_i^2 q_1} \cdot e(g_k, h_k)^{2x_i r_i q_1} \cdot e(h_k, h_k)^{r_i^2 q_1} \\
&\quad = e(g_k, g_k)^{x_i^2 q_1} \cdot e(g_k, u_k)^{2x_i r_i q_2 q_1} \cdot e(u_k, u_k)^{r_i^2 q_2^2 q_1} \\
&\quad = e(g_k, g_k)^{x_i^2 q_1} \\
&\quad = \hat{g}_{T,k}^{x_i^2}.
\end{aligned}$$

[*2] Strictly speaking, our description of the SGD assumption is slightly different from that of Ref. [3] in that $\mathcal{A}$ is given an additional element g_{q_1}. However, it can be easily shown that both definitions are equivalent (with a factor of 2). See Section 3.3 of Ref. [4] for details.

In the same way, other two terms can be written as

$$e(g_k^{y_i} h_k^{s_i},\, g_k^{y_i} h_k^{s_i})^{q_1} = \hat{g}_{T,k}^{y_i^2},$$
$$e(g_k^{x_i} h_k^{r_i},\, g_k^{y_i} h_k^{s_i})^{-2q_1} = \hat{g}_{T,k}^{-2x_i y_i}.$$

Therefore,

$$\Delta_k^{q_1} = \prod_{i=1}^{D} \hat{g}_{T,k}^{x_i^2} \cdot \hat{g}_{T,k}^{y_i^2} \cdot \hat{g}_{T,k}^{-2x_i y_i} = \hat{g}_{T,k}^{\sum_{i=1}^{D}(x_i^2+y_i^2-2x_i y_i)} = \hat{g}_{T,k}^{d_{E^2}(\vec{x},\vec{y})}.$$

Thus, by computing the discrete logarithm of $\Delta_k^{q_1}$ to the base $\hat{g}_{T,k}$, we can obtain $d_{E^2}(\vec{x},\vec{y})$.

It might be odd that $\mathcal{D}$ must compute at step (A-5) the discrete logarithm, which is regarded as one of computationally expensive operations in computer science. However, in case that the exponent is supposed to lie in a certain interval T (say $0 \le m < T$, where m is the exponent), we can compute the exponent with the complexity $O(\sqrt{T})$ by using Pollard's lambda method [33].

In our biometric authentication protocol, $\mathcal{D}$ has only to know whether the squared Euclidean distance is under the predefined threshold θ. Therefore, the computational complexity of $\mathcal{D}$ is $O(\sqrt{\theta})$. If $\mathcal{D}$ could not obtain the exponent within $O(\sqrt{\theta})$ computation, $\mathcal{D}$ can conclude that the squared Euclidean distance exceeds θ and will return reject.

3.3.2 Revocation

We show that the updated public key PK'_k and the updated template $E_{PK'_k}(\vec{x})$ are correct.

The updated template $E_{PK'_k}(\vec{x})$ is represented as

$$\begin{aligned} E_{PK'_k}(\vec{x}) &= (c^{\delta}_{\vec{x},1}, \ldots, c^{\delta}_{\vec{x},D}) \\ &= \left((g_k^{x_1} h_k^{r_1})^{\delta}, \ldots, (g_k^{x_D} h_k^{r_D})^{\delta}\right) \\ &= \left((g_k^{\delta})^{x_1} (h_k^{\delta})^{r_1}, \ldots, (g_k^{\delta})^{x_D} (h_k^{\delta})^{r_D}\right) \\ &= \left(g_k'^{x_1} h_k'^{r_1}, \ldots, g_k'^{x_D} h_k'^{r_D}\right). \end{aligned}$$

This is exactly the template of $\vec{x}$ under a new public key $PK'_k = (g_k^{\delta},\ h_k^{\delta},\ g_{T,}^{\delta^2} = e(g_k^{\delta}, g_k^{\delta}))$. In this way, given an old public key PK_k and an old ciphertext $E_{PK_k}(\vec{x})$, $\mathcal{S}$ can construct a new pair $(PK'_k, E_{PK'_k}(\vec{x}))$ under the same secret key SK without knowing SK and $\vec{x}$. This is exactly the hidden nature of the underlying Boneh-Goh-Nissim cryptosystem [3].

3.4 Security

In this section, we give formal definitions of the security required in Section 2.2.2 and prove the security of our protocol. For clarity, we use hereafter the following notation: $\Delta_k(\vec{x},\vec{y})$ denotes the encrypted distance of $\vec{x}$ and $\vec{y}$.

3.4.1 Security Against the Server $\mathcal{S}$

Security against $\mathcal{S}$ is defined by the advantage of $\mathcal{S}$ in the following semantic security game between $\mathcal{S}$ and the challenger C.

Setup. C runs the setup process and gives $PK, \{PK_k\}_{k=1,\ldots,N}$ to $\mathcal{S}$. Then, C picks $\beta \xleftarrow{\cup} \{0,1\}$ and uses the same value throughout the game.

Query. $\mathcal{S}$ adaptively makes two types of queries in an arbitrary order:

- *Enrollment query.* $\mathcal{S}$ sends $(k^{(i)}, \vec{x}_0^{(i)}, \vec{x}_1^{(i)})$ to C, where $1 \le i \le N$. C returns the corresponding ciphertext $E_{PK_{k^{(i)}}}(\vec{x}_\beta^{(i)})$.
- *Authentication query.* This query consists of two consecutive procedures:
  (1) $\mathcal{S}$ sends $(k^{(j)}, \vec{y}_0^{(j)}, \vec{y}_1^{(j)})$ to C. C returns the corresponding ciphertext $E_{PK_{k^{(j)}}}(\vec{y}_\beta^{(j)})$.
  (2) After receiving $E_{PK_{k^{(j)}}}(\vec{y}_\beta^{(j)})$, $\mathcal{S}$ computes the encrypted distance $\Delta_{k^{(j)}}(\vec{x}_\beta^{(j)}, \vec{y}_\beta^{(j)})$ by using $E_{PK_{k^{(j)}}}(\vec{x}_\beta^{(j)})$ which must have been queried in the enrollment query, and sends $\Delta_{k^{(j)}}(\vec{x}_\beta^{(j)}, \vec{y}_\beta^{(j)})$ to C. C returns the authentication result (accept or reject).

  $\mathcal{S}$ can make a polynomial number of authentication queries, i.e. $1 \le j \le \mathrm{poly}(\lambda)$. The restriction is that if $\vec{x}_0^{(j)}$ and $\vec{y}_0^{(j)}$ are accepted vectors, then $\vec{x}_1^{(j)}$ and $\vec{y}_1^{(j)}$ must also be accepted ones, and vice versa.

Guess. $\mathcal{S}$ outputs a guess β' of β.

The advantage of $\mathcal{S}$ in the above game is defined as $\mathbf{Adv}_{\mathcal{S}}(\lambda) = \left|\Pr[\beta' = \beta] - \frac{1}{2}\right|$.

We justify here that the above security game indeed captures the real situation of the server $\mathcal{S}$.

First, we should note that we must define the semantic security game properly so that it captures *the multi-user setting*, since our system accommodates multiple users. Generally speaking, an adversary in the multi-user setting is more advantageous than one in the single-user setting in that he or she may obtain encryptions of some related messages under different public keys [*3]. In order to capture this kind of attacks, we employ the multi-user-based semantic security game given by Bellare, Boldyreva, and Micali [2], in which the adversary obtains not a single challenge ciphertext but *many challenge ciphertexts* from the challenger.

Next, our semantic security game should provide for $\mathcal{S}$ all the data retrieval access of the real protocol. Therefore, we defined the setup, the enrollment query and the authentication query.

Finally, we must prevent trivial attacks of $\mathcal{S}$. If there is no restriction on the authentication query, $\mathcal{S}$ can always win the semantic security game by throwing vectors $(k^{(j)}, \vec{y}_0^{(j)}, \vec{y}_1^{(j)})$ such that $\vec{x}_0^{(j)}$ and $\vec{y}_0^{(j)}$ are accepted and $\vec{x}_1^{(j)}$ and $\vec{y}_1^{(j)}$ are rejected. Thus, we must disallow such queries.

Definition 2. *We say that a biometric authentication protocol is semantically secure against the PPT server $\mathcal{S}$ if for all PPT adversarial servers $\mathcal{S}$, $\mathbf{Adv}_{\mathcal{S}}(\lambda)$ is a negligible function of λ.*

Theorem 1. *Our biometric authentication protocol in Section 3.2 is semantically secure against the PPT server $\mathcal{S}$ if the subgroup decision assumption holds.*

A proof sketch is given in Appendix A.1.

3.4.2 Security Against the Decryptor $\mathcal{D}$

Definition 3. *We say that a biometric authentication protocol is semantically secure against the unconditionally computable decryptor $\mathcal{D}$ if for all unconditionally computable adversarial decryptors $\mathcal{D}$, it is impossible to obtain any partial information on $\vec{x}$ and $\vec{y}$, except for $d_{E^2}(\vec{x},\vec{y})$.*

Theorem 2. *Our biometric authentication protocol in Section 3.2 is semantically secure against the unconditionally computable decryptor $\mathcal{D}$.*

[*3] An example of such adversarial attacks was given by Håstad [17] for the RSA encryption scheme. It is shown that the plaintext can be recovered from three ciphertexts in different moduli.

Proof. The proof is straightforward. In the authentication process, $\mathcal{D}$ receives $\Delta_k = \Delta_k(\vec{x}, \vec{y})$. This is the only data $\mathcal{D}$ can obtain with regard to $\vec{x}$ and $\vec{y}$, and as in Section 3.3, this is the ciphertext of $d_{E^2}(\vec{x}, \vec{y})$. Thus the theorem holds. □

3.4.3 Security Against Eavesdroppers $\mathcal{E}$

In this protocol, the security against eavesdroppers $\mathcal{E}$ is reduced to the security against the server $\mathcal{S}$. This is because all the transmissions $\mathcal{E}$ can observe are equally observed (or, in fact, organized) by $\mathcal{S}$, thus the adversarial capabilities of $\mathcal{E}$ are less than those of $\mathcal{S}$ [*4].

3.5 Four Properties

We briefly mention that our protocol satisfies the desired four properties.

Accuracy: Our protocol does not affect the accuracy, because in the authentication process, the decryptor $\mathcal{D}$ can recover the squared Euclidean distance which is also used in the original (non-cryptographic) biometric authentication.

Diversity: A very large number of cancelable templates can be produced in our protocol by changing random values such as $r_1, \ldots, r_D \xleftarrow{\cup} \mathbb{Z}_n$. Also, it is impossible to cross-match templates even within a single database. In our protocol, templates for users $\mathcal{U}_i$ and $\mathcal{U}_j$ are

$$E_{PK_i}(\vec{x}) = (g_i^{x_1} h_i^{r_1}, \ldots, g_i^{x_D} h_i^{r_D})$$
$$E_{PK_j}(\vec{x}) = (g_j^{x_1} h_j^{r'_1}, \ldots, g_j^{x_D} h_j^{r'_D}),$$

where $g_i, g_j \xleftarrow{\cup} \mathbb{G}$ and $h_i, h_j \xleftarrow{\cup} \mathbb{G}_{q_1}$. Therefore, two templates are totally independent. Thus the cross-matching is impossible.

Revocability: Shown in Section 3.3. Note that an intruder who obtained the former template $E_{PK_k}(\vec{x})$ and the new public key PK'_k cannot update his template into $E_{PK'_k}(\vec{x})$. This is because computing

$$E_{PK'_k}(\vec{x}) = (g_k^{\delta x_1} h_k^{\delta r_1}, \ldots, g_k^{\delta x_D} h_k^{\delta r_D})$$

from

$$E_{PK_k}(\vec{x}) = (g_k^{x_1} h_k^{r_1}, \ldots, g_k^{x_D} h_k^{r_D}),$$
$$PK'_k = (g_k^{\delta}, h_k^{\delta}, g_{T,k}^{\delta^2})$$

is the computational Diffie-Hellman problem.

Security: All of the desired security properties have been proved in Section 3.4.

4. A Protocol Based on the Okamoto-Takashima Cryptosystem

We describe our second protocol that is based on the Okamoto-Takashima cryptosystem proposed in Pairing 2008 [30]. An important feature of the cryptosystem is that it is constructed on dual pairing vector spaces (DPVS) which enables a higher dimensional vector treatment of bilinear pairing groups of *prime order*. Again, our protocol satisfies all the desirable properties: accuracy, diversity, revocability, and security.

[*4] This reduction holds since we consider the security only for the *passive* eavesdroppers. For the *active* adversaries, we must consider their security separately. See the assumptions of our system in Section 2.2.2 and the discussions therein.

4.1 Cryptographic Primitives

Let q be prime, $\mathbb{G}$ be a cyclic additive group of order q, and $\mathbb{G}_T$ be a cyclic multiplicative group of order q. If a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ satisfies the following two conditions, e is called a cryptographic bilinear map:

- For all $U, V \in \mathbb{G}$ and $s, t \in \mathbb{Z}$, $e(sU, tV) = e(U, V)^{st}$.
- There exists a generator $G \in \mathbb{G}$ such that $e(G, G)$ is a generator of $\mathbb{G}_T$.

A dual pairing vector space $(q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e)$ is defined as follows. We define an n-dimensional vector space $\mathbb{V}$ by $\overbrace{\mathbb{G} \times \cdots \times \mathbb{G}}^{n}$ and a canonical basis $\mathbb{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n)$ of $\mathbb{V}$ by $\boldsymbol{a}_i = (\overbrace{0, \ldots, 0}^{i-1}, G, \overbrace{0, \ldots, 0}^{n-i})$. For $\boldsymbol{x} = (x_1 G, \ldots, x_n G) \in \mathbb{V}$ and $\boldsymbol{y} = (y_1 G, \ldots, y_n G) \in \mathbb{V}$, we define the addition between $\boldsymbol{x}$ and $\boldsymbol{y}$ on $\mathbb{V}$ by $\boldsymbol{x} + \boldsymbol{y} = ((x_1 + y_1)G, \ldots, (x_n + y_n)G) \in \mathbb{V}$ and the scalar multiplication by $\alpha \boldsymbol{x} = (\alpha x_1 G, \ldots, \alpha x_n G) \in \mathbb{V}$ for any $\alpha \in \mathbb{Z}$. Then $(q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e)$ constitutes a dual pairing vector space.

The pairing $e : \mathbb{V} \times \mathbb{V} \to \mathbb{G}_T$ is defined by $e(\boldsymbol{x}, \boldsymbol{y}) = \prod_{i=1}^{n} e(G_i, H_i) \in \mathbb{G}_T$, where $\boldsymbol{x} = (G_1, \ldots, G_n) \in \mathbb{V}$ and $\boldsymbol{y} = (H_1, \ldots, H_n) \in \mathbb{V}$. e is also a cryptographic bilinear map, that is, $e(s\boldsymbol{x}, t\boldsymbol{y}) = e(\boldsymbol{x}, \boldsymbol{y})^{st}$ for any $s, t \in \mathbb{Z}$ and if $e(\boldsymbol{x}, \boldsymbol{y}) = 1$ for all $\boldsymbol{y} \in \mathbb{V}$, we have $\boldsymbol{x} = \boldsymbol{0}$.

Let $\phi_{i,j}$ be a linear transformation on $\mathbb{V}$ such that $\phi_{i,j}(\boldsymbol{a}_j) = \boldsymbol{a}_i$ and $\phi_{i,j}(\boldsymbol{a}_k) = \boldsymbol{0}$ if $j \neq k$. This transformation can easily be implemented on DPVS by $\phi_{i,j}(\boldsymbol{x}) = (\overbrace{0, \ldots, 0}^{i-1}, G_j, \overbrace{0, \ldots, 0}^{n-i})$, where $\boldsymbol{x} = (G_1, \ldots, G_n)$.

The DPVS generation algorithm $\mathcal{G}_{\mathsf{dpvs}}$ takes as input a security parameter 1^λ and a dimension n, and outputs a description of $(q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e)$.

The canonical basis $\mathbb{A}$ is changed to the basis $\mathbb{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \in \mathbb{V}$ using a uniformly chosen linear transformation $X = (\chi_{i,j}) \xleftarrow{\cup} GL(n, \mathbb{F}_q)$ such that $\boldsymbol{b}_i = \sum_{j=1}^{n} \chi_{i,j} \boldsymbol{a}_j$, $(i = 1, \ldots, n)$.

As an intractable problem of DPVS, we can consider the following vector decomposition problem: given $\boldsymbol{v} = v_1 \boldsymbol{b}_1 + \cdots + v_{\ell_1} \boldsymbol{b}_{\ell_1}$ and $\mathbb{B}$, compute $\boldsymbol{v}'$ such that $\boldsymbol{v}' = v_1 \boldsymbol{b}_1 + \cdots + v_{\ell_2} \boldsymbol{b}_{\ell_2}$ and $\ell_2 + 1 \leq \ell_1$. Although this is believed to be intractable, one can efficiently compute it if X is given. Indeed, the following Deco algorithm does it:

$$\mathsf{Deco}\ (\boldsymbol{v}, \langle \boldsymbol{b}_1, \ldots, \boldsymbol{b}_{\ell_2} \rangle, X, \mathbb{B}) :$$
$$(t_{i,j}) \leftarrow X^{-1},\ \boldsymbol{v}' \leftarrow \sum_{i=1}^{\ell_1} \sum_{j=1}^{\ell_2} \sum_{\kappa=1}^{\ell_1} t_{i,j} \chi_{j,\kappa} \phi_{\kappa,i}(\boldsymbol{v})$$
$$\mathsf{return}\ \boldsymbol{v}'.$$

As in the original Okamoto-Takashima cryptosystem, our protocol is based on the following subspace decision (SSD) assumption.

Definition 4. *Suppose $e : \mathbb{V} \times \mathbb{V} \to \mathbb{G}_T$ is a cryptographic bilinear map. Let $Z := (q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e, (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{\ell_1}))$, $\boldsymbol{v} \xleftarrow{\cup} \langle \boldsymbol{b}_{\ell_2+1}, \ldots, \boldsymbol{b}_{\ell_1} \rangle$, and $\boldsymbol{r} \xleftarrow{\cup} \mathbb{V}$. We say that the SSD assumption holds if for any PPT algorithm $\mathcal{A}$, $|\Pr[\mathcal{A}(Z, \boldsymbol{v}) \to 1] - \Pr[\mathcal{A}(Z, \boldsymbol{r}) \to 1]|$ is negligible.*

4.2 Protocol Description

We show our construction of four protocols.

4.2.1 Setup Process

The setup process is as follows.

(S-1) $\mathcal{D}$ invokes $(q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e) \xleftarrow{R} \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, n = 3)$.

(S-2) $\mathcal{D}$ picks

$$\left\{ X^{(k)} = \begin{pmatrix} \chi_{1,1}^{(k)} & \chi_{1,2}^{(k)} & \chi_{1,3}^{(k)} \\ \chi_{2,1}^{(k)} & \chi_{2,2}^{(k)} & \chi_{2,3}^{(k)} \\ \chi_{3,1}^{(k)} & \chi_{3,2}^{(k)} & \chi_{3,3}^{(k)} \end{pmatrix} \right\}_{k=1,\ldots,N} \xleftarrow{U} GL(3, \mathbb{F}_q),$$

where N is the number of users, and sets

$$\left\{ \begin{pmatrix} \boldsymbol{b}_1^{(k)} \\ \boldsymbol{b}_2^{(k)} \\ \boldsymbol{b}_3^{(k)} \end{pmatrix} = \begin{pmatrix} \chi_{1,1}^{(k)} & \chi_{1,2}^{(k)} & \chi_{1,3}^{(k)} \\ \chi_{2,1}^{(k)} & \chi_{2,2}^{(k)} & \chi_{2,3}^{(k)} \\ \chi_{3,1}^{(k)} & \chi_{3,2}^{(k)} & \chi_{3,3}^{(k)} \end{pmatrix} \begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \boldsymbol{a}_3 \end{pmatrix} \right\}_{k=1,\ldots,N}.$$

$\mathcal{D}$ then sets $\{\mathbb{B}^{(k)} := (\boldsymbol{b}_1^{(k)}, \boldsymbol{b}_2^{(k)}, \boldsymbol{b}_3^{(k)})\}_{k=1,\ldots,N}$ and computes $\{h_{T,k} := e(\boldsymbol{b}_1^{(k)}, \boldsymbol{b}_1^{(k)})\}_{k=1,\ldots,N}$.

(S-3) $\mathcal{D}$ sets $PK := (q, \mathbb{V}, \mathbb{G}_T, \mathbb{A}, e)$, $\{PK_k := (\mathbb{B}^{(k)}, h_{T,k})\}_{k=1,\ldots,N}$ and $\{SK_k := X^{(k)}\}_{k=1,\ldots,N}$, and makes PK and $\{PK_k\}_{k=1,\ldots,N}$ publicly available.

PK is the public parameter of the system and PK_k is the public key for the user $\mathcal{U}_k$. SK_k is the secret key of the user $\mathcal{U}_k$ and thus stored secretly in $\mathcal{D}$.

4.2.2 Enrollment Process

The enrollment process of the user $\mathcal{U}_k$ is as follows.

(E-1) $\mathcal{U}_k$ picks $\{r_{2,i}, r_{3,i}\}_{i=1,\ldots,D} \xleftarrow{U} \mathbb{F}_q$ and encrypts his biometric feature vector $\vec{x} = (x_1, \ldots, x_D) \in \mathbb{F}_q^D$ under the public key PK_k as follows:

$$\left\{ \boldsymbol{c}_{\vec{x},i} := x_i \boldsymbol{b}_1^{(k)} + r_{2,i} \boldsymbol{b}_2^{(k)} + r_{3,i} \boldsymbol{b}_3^{(k)} \right\}_{i=1,\ldots,D}.$$

(E-2) $\mathcal{U}_k$ sends $E_{PK_k}(\vec{x}) = (\boldsymbol{c}_{\vec{x},1}, \ldots, \boldsymbol{c}_{\vec{x},D})$ to $\mathcal{S}$.

(E-3) $\mathcal{S}$ stores a tuple $(k, PK_k, E_{PK_k}(\vec{x}))$ as a template.

4.2.3 Authentication Process

The authentication process of the user $\mathcal{U}_k$ is as follows.

(A-1) $\mathcal{U}_k$ picks $\{s_{2,i}, s_{3,i}\}_{i=1,\ldots,D} \xleftarrow{U} \mathbb{F}_q$ and encrypts his biometric feature vector $\vec{y} = (y_1, \ldots, y_D) \in \mathbb{F}_q^D$ under the public key PK_k as follows:

$$\left\{ \boldsymbol{c}_{\vec{y},i} := y_i \boldsymbol{b}_1^{(k)} + s_{2,i} \boldsymbol{b}_2^{(k)} + s_{3,i} \boldsymbol{b}_3^{(k)} \right\}_{i=1,\ldots,D}.$$

(A-2) $\mathcal{U}_k$ sends $E_{PK_k}(\vec{y}) = (\boldsymbol{c}_{\vec{y},1}, \ldots, \boldsymbol{c}_{\vec{y},D})$ to $\mathcal{S}$.

(A-3) $\mathcal{S}$ picks $u_2, u_3, \{t_{1,i}, t_{2,i}, t_{3,i}\}_{i=1,\ldots,D} \xleftarrow{U} \mathbb{F}_q$, takes the corresponding tuple $(k, PK_k, E_{PK_k}(\vec{x}))$, and computes

$$\left\{ \boldsymbol{c}_{\Delta,i} := (\boldsymbol{c}_{\vec{x},i} - \boldsymbol{c}_{\vec{y},i}) + t_{1,i} \boldsymbol{b}_1^{(k)} + t_{2,i} \boldsymbol{b}_2^{(k)} + t_{3,i} \boldsymbol{b}_3^{(k)} \right\}_{i=1,\ldots,D},$$
$$\boldsymbol{c}_\Delta := \sum_{i=1}^{D} \left(2t_{1,i}(\boldsymbol{c}_{\vec{x},i} - \boldsymbol{c}_{\vec{y},i}) + t_{1,i}^2 \boldsymbol{b}_1^{(k)} \right) + u_2 \boldsymbol{b}_2^{(k)} + u_3 \boldsymbol{b}_3^{(k)}.$$

(A-4) $\mathcal{S}$ sends the components of the encrypted distance $(\boldsymbol{c}_{\Delta,1}, \ldots, \boldsymbol{c}_{\Delta,D}, \boldsymbol{c}_\Delta)$ to $\mathcal{D}$.

(A-5) $\mathcal{D}$ computes $\{z_i \boldsymbol{b}_1^{(k)} := \mathsf{Deco}(\boldsymbol{c}_{\Delta,i}, \langle \boldsymbol{b}_1^{(k)} \rangle, X^{(k)}, \mathbb{B})\}_{i=1,\ldots,D}$ and also $z\boldsymbol{b}_1^{(k)} := \mathsf{Deco}(\boldsymbol{c}_\Delta, \langle \boldsymbol{b}_1^{(k)} \rangle, X^{(k)}, \mathbb{B})$. Then $\mathcal{D}$ computes $d_{E^2}(\vec{x}, \vec{y}) := \mathsf{DLog}_{h_{T,k}} Z$, where

$$Z := \frac{\prod_{i=1}^{D} e(z_i \boldsymbol{b}_1^{(k)}, z_i \boldsymbol{b}_1^{(k)})}{e(z\boldsymbol{b}_1^{(k)}, \boldsymbol{b}_1^{(k)})}.$$

(A-6) $\mathcal{D}$ compares $d_{E^2}(\vec{x}, \vec{y})$ to the predefined threshold θ. If $d_{E^2}(\vec{x}, \vec{y}) < \theta$, $\mathcal{D}$ returns accept to $\mathcal{S}$; otherwise, $\mathcal{D}$ returns reject to $\mathcal{S}$.

4.2.4 Revocation Process

Suppose that the tuple

$$(k,\ PK_k = (\mathbb{B}^{(k)}, h_{T,k}),\ E_{PK_k}(\vec{x}) = (\boldsymbol{c}_{\vec{x},1}, \ldots, \boldsymbol{c}_{\vec{x},D}))$$

leaked out from $\mathcal{S}$. The revocation process is as follows.

(R-1) $\mathcal{S}$ picks $\delta \xleftarrow{U} \mathbb{F}_q$ and computes

$$\begin{pmatrix} \boldsymbol{b}_1'^{(k)} \\ \boldsymbol{b}_2'^{(k)} \\ \boldsymbol{b}_3'^{(k)} \end{pmatrix} := \delta \begin{pmatrix} \boldsymbol{b}_1^{(k)} \\ \boldsymbol{b}_2^{(k)} \\ \boldsymbol{b}_3^{(k)} \end{pmatrix}, \begin{pmatrix} \boldsymbol{c}'_{\vec{x},1} \\ \vdots \\ \boldsymbol{c}'_{\vec{x},D} \end{pmatrix} := \delta \begin{pmatrix} \boldsymbol{c}_{\vec{x},1} \\ \vdots \\ \boldsymbol{c}_{\vec{x},D} \end{pmatrix}, \ h'_{T,k} := h_{T,k}^{\delta^2}.$$

(R-2) $\mathcal{S}$ stores a new tuple $(k, PK'_k := (\mathbb{B}'^{(k)}, h'_{T,k}), E_{PK'_k}(\vec{x}) = (\boldsymbol{c}'_{\vec{x},1}, \ldots, \boldsymbol{c}'_{\vec{x},D}))$.

(R-3) $\mathcal{S}$ makes PK'_k publicly available.

4.3 Correctness

4.3.1 Authentication

We show that $\mathsf{DLog}_{h_{T,k}} Z$ is exactly the squared Euclidean distance of two vectors $\vec{x}$ and $\vec{y}$.

The numerator of Z can be written as

$$\prod_{i=1}^{D} e(z_i \boldsymbol{b}_1^{(k)}, z_i \boldsymbol{b}_1^{(k)}) = e(\boldsymbol{b}_1^{(k)}, \boldsymbol{b}_1^{(k)})^{\sum_{i=1}^{D} z_i^2} = h_{T,k}^{\sum_{i=1}^{D} z_i^2} = h_{T,k}^{\sum_{i=1}^{D} ((x_i - y_i) - t_{1,i})^2}$$

and the denominator of Z can be written as

$$e(z\boldsymbol{b}_1^{(k)}, \boldsymbol{b}_1^{(k)}) = e(\boldsymbol{b}_1^{(k)}, \boldsymbol{b}_1^{(k)})^z = h_{T,k}^z = h_{T,k}^{\sum_{i=1}^{D} (2t_{1,i}(x_i - y_i) + t_{1,i}^2)}.$$

Thus,

$$\begin{aligned} Z &= \frac{\prod_{i=1}^{D} e(z_i \boldsymbol{b}_1^{(k)}, z_i \boldsymbol{b}_1^{(k)})}{e(z\boldsymbol{b}_1^{(k)}, \boldsymbol{b}_1^{(k)})} \\ &= h_{T,k}^{\sum_{i=1}^{D} ((x_i - y_i) - t_{1,i})^2 - \sum_{i=1}^{D} (2t_{1,i}(x_i - y_i) + t_{1,i}^2)} \\ &= h_{T,k}^{d_{E^2}(\vec{x}, \vec{y})}. \end{aligned}$$

By computing the discrete logarithm of Z to the base $h_{T,k}$, we can obtain the squared Euclidean distance $d_{E^2}(\vec{x}, \vec{y})$.

4.3.2 Revocation

We show that the updated public key PK'_k and the updated template $E_{PK'_k}(\vec{x})$ are correct.

The updated template $E_{PK'_k}(\vec{x})$ is represented as

$$\begin{aligned} E_{PK'_k}(\vec{x}) &= (\delta \boldsymbol{c}_{\vec{x},1}, \ldots, \delta \boldsymbol{c}_{\vec{x},D}) \\ &= (\delta x_i \boldsymbol{b}_1^{(k)} + \delta r_{2,i} \boldsymbol{b}_2^{(k)} + \delta r_{3,i} \boldsymbol{b}_3^{(k)})_{i=1,\ldots,D} \\ &= (x_i (\delta \boldsymbol{b}_1^{(k)}) + r_{2,i} (\delta \boldsymbol{b}_2^{(k)}) + r_{3,i} (\delta \boldsymbol{b}_3^{(k)}))_{i=1,\ldots,D} \\ &= (\boldsymbol{c}'_{\vec{x},1}, \ldots, \boldsymbol{c}'_{\vec{x},D}). \end{aligned}$$

This is exactly the template of $\vec{x}$ under the new public key $PK'_k = (\mathbb{B}'^{(k)}, h'_{T,k})$.

As with the Boneh-Goh-Nissim cryptosystem, this nature is a hidden property of the underlying Okamoto-Takashima cryptosystem.

Table 1 Difference of the entities and security target between Bringer et al.'s protocol [6] and our protocols.

	Bringer et al.'s protocol [6]	Our protocols
Entities	Users $1,\ldots,N$ ($\mathcal{U}_1,\ldots,\mathcal{U}_N$)	Users $1,\ldots,N$ ($\mathcal{U}_1,\ldots,\mathcal{U}_N$)
	Database ($\mathcal{DB}$)	Server ($\mathcal{S}$)
	Service provider ($\mathcal{SP}$)	Decryptor ($\mathcal{D}$)
Security target	Relationship between the user's identity and its biometric information	User's biometric information *per se*

4.4 Security

4.4.1 Security Against the Server $\mathcal{S}$

Theorem 3. *Our biometric authentication protocol in Section 4.2 is semantically secure against the PPT server $\mathcal{S}$ if the SSD assumption holds.*

As with the proof of Theorem 1, this theorem can be proved by using a combination of the standard hybrid argument, thus details are omitted.

4.4.2 Security Against the Decryptor $\mathcal{D}$

Theorem 4. *Our biometric authentication protocol in Section 4.2 is semantically secure against the unconditionally computable decryptor $\mathcal{D}$.*

The proof is given in Appendix A.2.

4.4.3 Security Against an Eavesdropper $\mathcal{E}$

As in Section 3.4.3, the security against $\mathcal{E}$ is reduced to the security against $\mathcal{S}$.

4.5 Four Properties

We briefly mention that our protocol satisfies the desired four properties.

Accuracy: Our protocol does not affect the accuracy, because in the authentication process, the decryptor $\mathcal{D}$ can recover the squared Euclidean distance which is also used in the original (non-cryptographic) biometric authentication.

Diversity: A very large number of cancelable templates can be produced in our protocol by changing random values. Also, it is impossible to cross-match templates even within a single database. The reason is similar to that of Section 3.5 and omitted here.

Revocability: As in Section 3.5, an intruder cannot update his template because it is the computational Diffie-Hellman problem.

Security: All of the desired security properties have been proved in Section 4.4.

5. Evaluation of Our Protocols

In this section, we evaluate our protocols and discuss our contribution. In Section 5.1, we compare one of our protocols, the BGN-based protocol proposed in Section 3, to the conventional one, and discuss the things that can be done with our protocols. Then, we show the result of comparison between our two protocols in Section 5.2.

5.1 Comparison to the Conventional Protocol

Since our goal is to construct biometric authentication protocols which satisfy both security and user-friendliness, we take as a conventional protocol the Bringer and Chabanne's protocol [6] which satisfies both of them, and compare the data size and the computational complexity of their protocol to those of our BGN-based protocol and discuss the efficiency.

Note here that, the system model and the security target of Bringer et al.'s protocol are slightly different from ours. Prior to the comparison, we clarify the differences in **Table 1**. The most significant difference is the security target. Although our security target is the user's biometric information *per se*, Bringer et al.'s target is the relationship between a user's identity and its biometric information, since they regard biometric information as public data. Another difference is the entities in the system. In the Bringer et al.'s protocol, the database ($\mathcal{DB}$) stores templates and accepts authentication requests from users, and the service provider ($\mathcal{SP}$) decrypts ciphertexts. For details of their protocol, refer to Ref. [6].

Based on such differences, we give the results of the comparison of Bringer et al's protocol and our protocol in **Table 2**. In the Bringer et al's protocol, the public key size is $O(1)$, while that of our protocol is $O(N)$. This is due to the fact that the same public key can be shared among users in their protocol. On the other hand, the computational complexity of $\mathcal{U}_k$, $\mathcal{DB}$ and $\mathcal{SP}$ are $O(D\log N)$, $O(DN)$ and $O(D\log N)$, respectively, in their protocol, while that of $\mathcal{U}_k$, $\mathcal{S}$ and $\mathcal{D}$ are $O(D)$, $O(D)$ and $O(1)$, respectively, in our protocol. Especially, $O(DN)$ exponentiation operations of $\mathcal{DB}$ in their protocol could be an obstacle to practical use. As an example, consider the case of the iris recognition system in airport control in UAE [1]. In this system, $D = 2{,}048$ and $N = 840{,}751$, which leads to $DN = O(10^{10})$ exponentiation operations in the authentication. Our protocol reduces this to $D = 2{,}048$ exponentiation and pairing operations. This kind of computational complexity improvement, while not degrading the security and user-friendliness, is the contribution of our work that has not been done ever.

5.2 Comparison Between Our Protocols

We evaluate the data size and the computational complexity between our two protocols.

Table 3 summarizes the result of the evaluation. In Table 3, the template size denotes the size of $\{E_{PK_k}(\vec{x})\}_{k=1,\ldots,N}$ and the complexity denotes the computational complexity in the authentication process. Note that the authentication data size is identical to the template size, and the computational complexity of a user in the enrollment process is identical to that of a user in the authentication process.

In our protocols, the public-key size depends on the number of users and the template size depends on the number of users and the feature vector size. The computational complexity of the au-

Table 2 Data size and computational complexity of the Bringer et al.'s protocol [6] (based on Lipmaa's PIR protocol [28]) and our protocol (BGN-based one).

Data size / Complexity	Bringer et al.'s protocol	Our protocol (BGN-based one)
Public key	$3\|\mathbb{Z}_n\|$	$2N\|\hat{\mathbb{G}}\| + N\|\hat{\mathbb{G}}_T\|$
Secret key	$3\|\mathbb{Z}_n\|$	$\|\mathbb{Z}_n\|$
Template	$DN\|\mathbb{Z}_n\|$	$DN\|\hat{\mathbb{G}}\|$
Authentication Data	$[\mathcal{U}_k \to \mathcal{DB}] : O(D\log^2 N)\|\mathbb{Z}_n\|$	$[\mathcal{U}_k \to \mathcal{S}] : D\|\hat{\mathbb{G}}\|$
	$[\mathcal{DB} \to \mathcal{SP}] : O(D)\|\mathbb{Z}_n\|$	$[\mathcal{S} \to \mathcal{D}] : \|\hat{\mathbb{G}}_T\|$
Enrollment	$[\mathcal{U}_k] : O(D)\dot{\mathsf{e}}$	$[\mathcal{U}_k] : 2D\hat{\mathsf{e}}$
Authentication	$[\mathcal{U}_k] : O(D\log N)\dot{\mathsf{e}}$	$[\mathcal{U}_k] : 2D\hat{\mathsf{e}}$
	$[\mathcal{DB}] : O(DN)\dot{\mathsf{e}}$	$[\mathcal{S}] : (D+2)\hat{\mathsf{e}} + (3D+2)\hat{\mathsf{p}}$
	$[\mathcal{SP}] : O(D\log N)\dot{\mathsf{e}}$	$[\mathcal{D}] : 2\hat{\mathsf{e}} + O(\sqrt{\theta})$

$n = pq$: a product of two prime numbers p and q.
$|\mathbb{Z}_n|$: the data size of an element of $\mathbb{Z}_n$.
$|\hat{\mathbb{G}}|, |\hat{\mathbb{G}}_T|$: the data size of an element of a bilinear group of order n.
$\dot{\mathsf{e}}, \hat{\mathsf{e}}$: an exponentiation in $\mathbb{Z}_n$ and $\hat{\mathbb{G}}$ (or $\hat{\mathbb{G}}_T$), respectively.
$\hat{\mathsf{p}}$: a pairing operation in $\hat{\mathbb{G}}$ (or $\hat{\mathbb{G}}_T$).

Table 3 Data size and computational complexity of the two protocols.

Data size / Complexity	Protocol in Section 3 (BGN-based protocol)	Protocol in Section 4 (OT-based protocol)
Public key	$2N\|\hat{\mathbb{G}}\| + N\|\hat{\mathbb{G}}_T\|$	$9N\|\mathbb{G}\| + N\|\mathbb{G}_T\|$
Secret key	$\|\mathbb{Z}_n\|$	$9N\|\mathbb{F}_q\|$
Template	$DN\|\hat{\mathbb{G}}\|$	$3DN\|\mathbb{G}\|$
Encrypted distance	$\|\hat{\mathbb{G}}_T\|$	$3(D+1)\|\mathbb{G}\|$
User $\mathcal{U}_k$	$2D\hat{\mathsf{e}}$	$9D\mathsf{e}$
Server $\mathcal{S}$	$(D+2)\hat{\mathsf{e}} + (3D+2)\hat{\mathsf{p}}$	$(15D+6)\mathsf{e}$
Decryptor $\mathcal{D}$	$2\hat{\mathsf{e}} + O(\sqrt{\theta})$	$9(D+1)\mathsf{e} + 3(D+1)\mathsf{p} + O(\sqrt{\theta})$

$|\hat{\mathbb{G}}|, |\hat{\mathbb{G}}_{q_1}|, |\hat{\mathbb{G}}_T|$: the data size of an element of *composite order* bilinear group.
$|\mathbb{G}|, |\mathbb{G}_T|$: the data size of an element of *prime order* bilinear group.
$|\mathbb{Z}_n|, |\mathbb{F}_q|$: the data size of an element of $\mathbb{Z}_n$ and $\mathbb{F}_q$, respectively.
$\hat{\mathsf{e}}, \hat{\mathsf{p}}$: an exponentiation and a pairing operation in $\hat{\mathbb{G}}$ (or $\hat{\mathbb{G}}_T$), respectively.
e, p : an exponentiation and a pairing operation in $\mathbb{G}$ (or $\mathbb{G}_T$), respectively.

thentication process depends on the feature vector size; therefore, our protocols are reasonable when biometric feature vectors are well-designed and the vector size is relatively short.

Apparently, our first protocol (BGN-based protocol) is attractive both in the data size and in the complexity. However, the first protocol is based on *composite order* bilinear groups while the second protocol (OT-based protocol) is based on *prime order* bilinear groups. The bit length of prime order bilinear groups is typically 160–200 bits and that of composite order bilinear groups is at least 1,024 bits (or preferably 2,048 bits) since it must be infeasible to factor. This makes the pairing operation dozens of times slower (e.g., 50 times slower [14]) and less attractive.

Another difference is that in the first protocol, the server $\mathcal{S}$ must compute the pairing, while in the second protocol the decryptor $\mathcal{D}$ does that.

6. Conclusion

In this paper, we have proposed two biometric authentication protocols each of which satisfies all of the required properties of cancelable biometrics as well as user-friendliness. The first protocol is based on the Boneh-Goh-Nissim cryptosystem [3] which enables almost directly the 2-DNF evaluation on ciphertexts, and the second one is based on the Okamoto-Takashima cryptosystem [30] which allows us to evaluate the 2-DNF predicate by suitably modifying the algorithms. We have proved the security of our protocols based on the assumptions of the underlying cryptosystems. The evaluation of the two protocols has shown that the first protocol is preferable in the case that the authentication server has more computational power than the decryptor, whereas the second one is preferable in the opposite case or when prime order bilinear groups are desirable.

A future direction of our work will be to increase the security from the semi-honest model to the malicious model without severely degrading the efficiency.

Acknowledgments We would like to thank the anonymous reviewers for their invaluable comments.

Reference

[1] Al-Raisi, A.N. and Al-Khouri, A.M.: Iris recognition and the challenge of homeland and border control security in UAE, *Telematics and Informatics*, Vol.25, No.2, pp.117–132 (2008).
[2] Bellare, M., Boldyreva, A. and Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements, *EUROCRYPT 2000, LNCS*, Preneel, B. (Ed.), Vol.1807, pp.259–274, Springer, Heidelberg (2000).
[3] Boneh, D., Goh, E.-J. and Nissim, K.: Evaluating 2-DNF formulas on ciphertexts, *TCC 2005, LNCS*, Kilian, J. (Ed.), Vol.3378, pp.325–341, Springer, Heidelberg (2005).
[4] Boneh, D., Sahai, A. and Waters, B.: Fully collusion resistant traitor tracing with short ciphertexts and private keys, *EUROCRYPT*

2006, LNCS, Vaudenay, S. (Ed.), Vol.4004, pp.573–592, Springer, Heidelberg (2006).
[5] Boulgouris, N.V., Plataniotis, K.N. and Micheli-Tzanakou, E. (Eds.): *Biometrics: Theory, methods, and applications*, Wiley-IEEE Press (2009).
[6] Bringer, J. and Chabanne, H.: An authentication protocol with encrypted biometric data, *Africacrypt 2008, LNCS*, Vaudenay, S. (Ed.), Vol.5023, pp.109–124, Springer, Heidelberg (2008).
[7] Bringer, J., Chabanne, H. and Kindarji, B.: Anonymous identification with cancelable biometrics, *ISPA 2009*, pp.494–499 (2009).
[8] Connie, T., Teoh, A.B.J., Goh, M. and Ngo, D.: PalmHashing: A novel approach for cancelable biometrics, *Inf. Process. Let.*, Vol.93, pp.1–5 (2005).
[9] Daugman, J.: High confidence personal identification by rapid video analysis of iris texture, *ICCST 1992*, pp.50–60 (1992).
[10] Daugman, J.: Results from 200 billion iris cross-comparisons, Technical Report UCAM-CL-TR-635, University of Cambridge (2005).
[11] Dodis, Y., Reyzin, L. and Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data, *EUROCRYPT 2004, LNCS*, Cachin, C. and Camenisch, J.L. (Eds.), Vol.3027, pp.523–540, Springer, Heidelberg (2004).
[12] ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms, *CRYPTO 84, LNCS*, Blakley, G.R. and Chaum, D. (Eds.), Vol.196, pp.10–18, Springer, Heidelberg (1984).
[13] Erkin, Z., Franz, M., Guajardo, J., Katzenbeisser, S., Lagendijk, I. and Toft, T.: Privacy-preserving face recognition, *PETS 2009, LNCS*, Goldberg, I. and Atallah, M.J. (Eds.), Vol.56725, pp.235–253, Springer, Heidelberg (2009).
[14] Freeman, D.M.: Converting pairing-based cryptosystems from composite-order groups to prime-order groups, *EUROCRYPT 2010, LNCS*, Gilbert, H. (Ed.), Vol.6110, pp.44–61, Springer, Heidelberg (2010).
[15] Goldreich, O.: *Foundations of cryptography*, Vol.1 Basic Tools, Cambridge University Press (2001).
[16] Goldreich, O.: *Foundations of cryptography*, Vol.2 Basic Applications, Cambridge University Press (2004).
[17] Hastad, J.: Solving simultaneous modular equations of low degree, *SIAM Journal of Computing*, Vol.17, pp.336–341 (1988).
[18] Hirano, T., Mori, T., Hattori, M., Ito, T. and Matsuda, N.: Homomorphic encryption based countermeasure against active attacks in privacy-preserving biometric authentication, *Technical Report of IEICE*, Vol.110, No.443, pp.7–14 (2011).
[19] Hirata, S. and Takahashi, K.: Cancelable biometrics with perfect secrecy for correlation-based matching, *ICB 2009, LNCS*, Tistarelli, M. and Nixon, M.S. (Eds.), Vol.5558, pp.868–878, Springer, Heidelberg (2009).
[20] Jain, A.K., Nandakumar, K. and Nagar, A.: Biometric template security, *EURASIP Journal on Advances in Signal Processing*, Vol.2008, pp.1–17 (2008).
[21] Jain, A.K., Prabhakar, S., Hong, L. and Pankanti, S.: Filterbank-based fingerprint matching, *IEEE Trans. Image Processing*, Vol.9, No.5, pp.846–859 (2000).
[22] Juels, A. and Sudan, M.: A fuzzy vault scheme, *ISIT 2002*, p.408 (2002).
[23] Juels, A. and Wattenberg, M.: A fuzzy commitment scheme, *ACM CCS 1999*, pp.28–36 (1999).
[24] Kerschbaum, F., Atallah, M.J., M'Raïhi, D. and Rice, J.R.: Private fingerprint verification without local storage, *ICBA 2004, LNCS*, Zhang, D. and Jain, A.K. (Eds.), Vol.3072, pp.387–394, Springer, Heidelberg (2004).
[25] Kikuchi, H., Nagai, K., Ogata, W. and Nishigaki, M.: Privacy-preserving similarity evaluation and application to remote biometrics authentication, *MDAI 2008, LNAI*, Torra, V. and Narukawa, Y. (Eds.), Vol.5285, pp.3–14, Springer, Heidelberg (2008).
[26] Kohner, E., Hamilton, A., Saunders, S., Sutcliffe, B. and Bulpitt, C.: The retinal blood flow in diabetes, *Diabetologia*, Vol.541, pp.27–33 (1975).
[27] Kong, A.W.-K. and Zhang, D.: Competitive coding scheme for palmprint verification, *ICPR 2004*, Vol.1, pp.520–523 (2004).
[28] Lipmaa, H.: An oblivious transfer protocol with log-squared communication, *ISC 2005, LNCS*, Zhou, J., Lopez, J., Deng, R.H. and Bao, F. (Eds.), Vol.3650, pp.314–328, Springer, Heidelberg (2005).
[29] Maiorana, E., Campisi, P., Ortega-Garcia, J. and Neri, A.: Cancelable biometrics for HMM-based signature recognition, *BTAS 2008*, pp.1–6 (2008).
[30] Okamoto, T. and Takashima, K.: Homomorphic encryption and signatures from vector decomposition, *Pairing 2008, LNCS*, Galbraith, S.D. and Paterson, K.G. (Eds.), Vol.5209, pp.57–74, Springer, Heidelberg (2008).
[31] Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes, *EUROCRYPT 99, LNCS*, Stern, J. (Ed.), Vol.1592, pp.223–238, Springer, Heidelberg (1999).
[32] Penrose, L.: Dermatoglyphic topology, *Nature*, Vol.205, pp.544–546 (1965).
[33] Pollard, J.M.: Monte Carlo methods for index computation (mod p), *Mathematics of Computation*, Vol.32, No.143, pp.918–924 (1978).
[34] Ratha, N.K., Connell, J.H. and Bolle, R.M.: Enhancing security and privacy in biometrics-based authentication systems, *IBM Systems Journal*, Vol.40, No.3, pp.614–634 (2001).
[35] Sakashita, T., Shibata, Y., Yamamoto, T., Takahashi, K., Ogata, W., Kikuchi, H. and Nishigaki, M.: A proposal of efficient remote biometric authentication protocol, *IWSEC 2009, LNCS*, Takagi, T. and Mambo, M. (Eds.), Vol.5824, pp.212–227, Springer, Heidelberg (2009).
[36] Savvides, M., Kumar, B.V. and Khosla, P.: Cancelable biometric filters for face recognition, *ICPR 2004*, Vol.3, IEEE Computer Society, pp.922–925 (2004).
[37] Takahashi, K. and Hirata, S.: Generating provably secure cancelable fingerprint templates based on correlation-invariant random filtering, *BTAS 2009*, pp.1–6 (2009).
[38] Takahashi, K. and Hirata, S.: Cancelable biometrics with provable security and its application to fingerprint verification, *IEICE Trans. Fundamentals*, Vol.E94-A, No.1, pp.233–244 (2011).
[39] Teoh, A.B.J., Toh, K.-A. and Yip, W.K.: 2^N discretisation of BioPhasor in cancellable biometrics, *ICB 2007, LNCS*, Lee, S.-W. and Li, S.Z. (Eds.), Vol.4642, pp.435–444, Springer, Heidelberg (2007).
[40] Zuo, J., Ratha, N. and Connell, J.: Cancelable iris biometric, *ICPR 2008*, pp.1–4 (2008).

Appendix

A.1 A Proof Sketch of Theorem 1

We give a proof sketch of Theorem 1.

We prove the theorem by using a standard hybrid argument [15]. We consider a sequence of games defined below.

Definition 5. *Let* Game_0 *denote the original game and* Q_1 *denote the number of enrollment queries* ($1 \le Q_1 \le N$). Game_{1,ℓ_1} *is defined as a sequence of hybrid games in the following way:*

- *For the first* ℓ_1 *enrollment queries, on receiving* $(k^{(i)}, \vec{x}_0^{(i)}, \vec{x}_1^{(i)})$, *the challenger returns a fake ciphertext* $\left(g_{k^{(i)}}^{R_1}, \ldots, g_{k^{(i)}}^{R_D}\right)$, *where* $R_1, \ldots, R_D \xleftarrow{\cup} \mathbb{Z}_n$, *instead of a real ciphertext* $\left(g_{k^{(i)}}^{x_{\beta,1}^{(i)}} \cdot h_{k^{(i)}}^{r_1}, \ldots, g_{k^{(i)}}^{x_{\beta,D}^{(i)}} \cdot h_{k^{(i)}}^{r_D}\right)$.
- *For the last* $Q_1 - \ell_1$ *queries, the challenger returns a real ciphertext.*

Definition 6. *Let* Q_2 *denote the number of authentication queries.* Game_{2,ℓ_2} *is defined as a sequence of hybrid games in the following way:*

- *For all the enrollment queries, the challenger returns a fake ciphertext* $\left(g_{k^{(i)}}^{R_1}, \ldots, g_{k^{(i)}}^{R_D}\right)$ (*as in Definition 5*).
- *For the first* ℓ_2 *authentication queries, on receiving* $(k^{(j)}, \vec{y}_0^{(j)}, \vec{y}_1^{(j)})$, *the challenger returns a fake ciphertext* $\left(g_{k^{(j)}}^{S_1}, \ldots, g_{k^{(j)}}^{S_D}\right)$, *where* $S_1, \ldots, S_D \xleftarrow{\cup} \mathbb{Z}_n$, *instead of a real ciphertext* $\left(g_{k^{(j)}}^{y_{\beta,1}^{(j)}} \cdot h_{k^{(j)}}^{s_1}, \ldots, g_{k^{(j)}}^{y_{\beta,D}^{(j)}} \cdot h_{k^{(j)}}^{s_D}\right)$.
- *For the last* $Q_2 - \ell_2$ *queries, the challenger returns a real ciphertext.*

It is obvious that $\mathsf{Game}_{1,0}$ is identical to Game_0 and $\mathsf{Game}_{2,0}$ is identical to $\mathsf{Game}_{1,N}$. Also, the advantage of $\mathcal{S}$ in Game_{2,Q_2} is 0 because all the ciphertexts $\mathcal{S}$ receives contain no any partial information on β and all the authentication results $\mathcal{S}$ receives are independent of β (due to the restriction on authentication queries). Therefore, by using the standard hybrid argument, the theorem holds if we could prove the following two lemmas.

Lemma 1. *Assume that the SGD assumption holds. Then, for any*

PPT adversarial server $\mathcal{S}$, the difference of advantage in winning Game_{1,ℓ_1} *and* $\mathsf{Game}_{1,\ell_1+1}$ *is negligible.*

Lemma 2. *Assume that the SGD assumption holds. Then, for any PPT adversarial server $\mathcal{S}$, the difference of advantage in winning* Game_{2,ℓ_2} *and* $\mathsf{Game}_{2,\ell_2+1}$ *is negligible.*

Both Lemma 1 and Lemma 2 can be proved almost in the same way. Therefore, we show here the proof of Lemma 1.

Prior to that, we further define a sequence of games below.

Definition 7. *Define a sequence of hybrid games* $\mathsf{Game}_{1,\ell_1,m}(0 \le m \le D)$ *in the following way:*

- *For the first ℓ_1 enrollment queries, the challenger returns a fake ciphertext* $\left(g_{k^{(i)}}^{R_1}, \ldots, g_{k^{(i)}}^{R_D}\right)$.
- *For the $i = (\ell_1 + 1)$-th enrollment query, the challenger returns a (slightly modified) fake ciphertext*

$$(\overbrace{g_{k^{(i)}}^{R_1}, \ldots, g_{k^{(i)}}^{R_m}}^{m}, \overbrace{g_{k^{(i)}}^{x^{(i)}_{\beta,m+1}} h_{k^{(i)}}^{r_{m+1}}, \ldots, g_{k^{(i)}}^{x^{(i)}_{\beta,D}} h_{k^{(i)}}^{r_D}}^{D-m}).$$

- *For the last $Q_1 - (\ell_1 + 1)$ queries, the challenger returns a real ciphertext.*

It is obvious that $\mathsf{Game}_{1,\ell_1,0}$ is identical to Game_{1,ℓ_1} and $\mathsf{Game}_{1,\ell_1,D}$ is identical to $\mathsf{Game}_{1,\ell_1+1}$.

Lemma 3. *Assume that the SGD assumption holds. Then, for any PPT adversarial server $\mathcal{S}$, the difference of advantage in winning* $\mathsf{Game}_{1,\ell_1,m}$ *and* $\mathsf{Game}_{1,\ell_1,m+1}$ *is negligible.*

Proof. We assume that there exists an adversarial server $\mathcal{S}$ that distinguishes between $\mathsf{Game}_{1,\ell_1,m}$ and $\mathsf{Game}_{1,\ell_1,m+1}$ with a non-negligible advantage. We show that there exists a simulator $\mathcal{B}$ which uses $\mathcal{S}$ to solve the SGD problem with a non-negligible advantage. Suppose $\mathcal{B}$ is given a challenge instance (Z, X) of the SGD problem, where $Z = (n, g, g_{q_1}, e)$ and X is either $T \xleftarrow{\cup} \mathbb{G}_{q_1}$ or $R \xleftarrow{\cup} \mathbb{G}$. $\mathcal{B}$ answers this by using $\mathcal{S}$ in a black-box way as follows.

Setup. $\mathcal{B}$ picks $\{a_k, b_k\}_{k=1,\ldots,N} \xleftarrow{\cup} \mathbb{Z}_n$. Then $\mathcal{B}$ sets $\{g_k := g^{a_k},\ h_k := g_{q_1}^{b_k},\ g_{T,k} := e(g_k, g_k)\}_{k=1,\ldots,N}$. $\mathcal{B}$ sends to $\mathcal{S}$ public keys $PK := (n, \langle g\rangle, \langle e(g,g)\rangle, e)$ and $\{PK_k := (g_k, h_k, g_{T,k})\}_{k=1,\ldots,N}$. Finally, $\mathcal{B}$ flips $\beta \xleftarrow{\cup} \{0,1\}$.

Query. As for the enrollment query, $\mathcal{S}$ sends $(k^{(i)}, \vec{x}_0^{(i)}, \vec{x}_1^{(i)})$ to $\mathcal{B}$. On receiving the query, $\mathcal{B}$ first records $(k^{(i)}, \vec{x}_0^{(i)}, \vec{x}_1^{(i)})$ in a "query table" (a table for recording all the enrollment queries). Then, for the i-th query, $\mathcal{B}$ answers in the following way.

- $1 \le i \le \ell_1$: $\mathcal{B}$ picks $R_1, \ldots, R_D \xleftarrow{\cup} \mathbb{Z}_n$ and returns a fake ciphertext $\left(g_{k^{(i)}}^{R_1}, \ldots, g_{k^{(i)}}^{R_D}\right)$.
- $i = \ell_1 + 1$: $\mathcal{B}$ picks $R_1, \ldots, R_m \xleftarrow{\cup} \mathbb{Z}_n$ and $r_{m+2}, \ldots, r_D \xleftarrow{\cup} \mathbb{Z}_n$, and returns a slightly modified fake ciphertext

$$(\overbrace{g_{k^{(i)}}^{R_1}, \ldots, g_{k^{(i)}}^{R_m}}^{m}, g_{k^{(i)}}^{x^{(i)}_{\beta,m+1}} X, \overbrace{g_{k^{(i)}}^{x^{(i)}_{\beta,m+2}} h_{k^{(i)}}^{r_{m+2}}, \ldots, g_{k^{(i)}}^{x^{(i)}_{\beta,D}} h_{k^{(i)}}^{r_D}}^{D-(m+1)}).$$

- $\ell_1 + 2 \le i \le N$: $\mathcal{B}$ generates a real ciphertext $E_{PK_{k^{(i)}}}(\vec{x}_\beta^{(i)})$ and returns it.

As for the authentication query, on receiving $(k^{(j)}, \vec{y}_0^{(j)}, \vec{y}_1^{(j)})$, $\mathcal{B}$ first returns a real ciphertext $E_{PK_{k^{(j)}}}(\vec{y}_\beta^{(j)})$. Then, on receiving an encrypted distance, $\mathcal{B}$ refers to the query table and picks $\vec{x}_\beta^{(j)}$. If $d_{E^2}(\vec{x}_\beta^{(j)}, \vec{y}_\beta^{(j)}) < \theta$ then return accept; otherwise return reject.

Guess. $\mathcal{S}$ outputs $\beta' \in \{0,1\}$. If $\beta' = \beta$, then $\mathcal{B}$ outputs 1; otherwise, $\mathcal{B}$ outputs 0.

In the simulation above, suppose that $X = T$, i.e. X is a random element in subgroup $\mathbb{G}_{q_1}$. In this case, $\mathcal{B}$ simulates $\mathsf{Game}_{1,\ell_1,m+1}$. Suppose that $X = R$, i.e. X is a random element in the group $\mathbb{G}$. In this case, $\mathcal{B}$ simulates $\mathsf{Game}_{1,\ell_1,m}$. Since $\mathcal{S}$ is assumed to have a non-negligible advantage in distinguishing $\mathsf{Game}_{1,\ell_1,m}$ and $\mathsf{Game}_{1,\ell_1,m+1}$, $\mathcal{B}$ can utilize this advantage directly for distinguishing the SGD instance. Thus the lemma holds. □

A.2 Proof of Theorem 4

Since $\mathcal{D}$ has the secret key SK_k and is assumed to be unconditionally computable, $\mathcal{D}$ can compute from $(\boldsymbol{c}_{\Delta,1}, \ldots, \boldsymbol{c}_{\Delta,D}, \boldsymbol{c}_\Delta)$ their discrete log values as follows:

$$\begin{aligned} z_1 &= (x_1 - y_1) - t_{1,1}, \\ &\vdots \\ z_D &= (x_D - y_D) - t_{1,D}, \\ z &= \sum_{i=1}^{D} (2t_{1,i}(x_i - y_i) + t_{1,i}^2). \end{aligned}$$

Now we consider the distribution of the tuple $(z_1, \ldots, z_D, z)$. Let $\mathbf{D}(\vec{x}, \vec{y})$ be the real distribution and $\mathbf{R}(d_{E^2}(\vec{x}, \vec{y}))$ be a random distribution as follows:

$$\begin{aligned} \mathbf{D}(\vec{x}, \vec{y}) := \{&(z_1, \ldots, z_D, z) \mid t_{1,1}, \ldots, t_{1,D} \xleftarrow{\cup} \mathbb{F}_q, \\ &\{z_i := (x_i - y_i) + t_{1,i}\}_{i=1,\ldots,D},\ z := \textstyle\sum_{i=1}^{D} (2t_{1,i}(x_i - y_i) + t_{1,i}^2)\}, \end{aligned}$$

$$\begin{aligned} \mathbf{R}(d_{E^2}(\vec{x}, \vec{y})) := \{&(z'_1, \ldots, z'_D, z') \mid z'_1, \ldots, z'_D \xleftarrow{\cup} \mathbb{F}_q, \\ &z' := \textstyle\sum_{i=1}^{D} z_i^2 - d_{E^2}(\vec{x}, \vec{y})\}. \end{aligned}$$

z in $\mathbf{D}(\vec{x}, \vec{y})$ can be written as

$$\begin{aligned} z &= \sum_{i=1}^{D} (2t_{1,i}(x_i - y_i) + t_{1,i}^2) \\ &= \sum_{i=1}^{D} ((x_i - y_i) + t_{1,i})^2 - \sum_{i=1}^{D} (x_i - y_i)^2 \\ &= \sum_{i=1}^{D} z_i^2 - d_{E^2}(\vec{x}, \vec{y}). \end{aligned}$$

Therefore, $\mathbf{D}(\vec{x}, \vec{y})$ is represented as

$$\begin{aligned} \mathbf{D}(\vec{x}, \vec{y}) := \{&(z_1, \ldots, z_D, z) \mid t_{1,1}, \ldots, t_{1,D} \xleftarrow{\cup} \mathbb{F}_q, \\ &\{z_i := (x_i - y_i) + t_{1,i}\}_{i=1,\ldots,D},\ z := \textstyle\sum_{i=1}^{D} z_i^2 - d_{E^2}(\vec{x}, \vec{y}). \end{aligned}$$

This distribution is, in fact, identical to $\mathbf{R}(d_{E^2}(\vec{x}, \vec{y}))$, because $z_i := (x_i - y_i) + t_{1,i}$ ($t_{1,i} \xleftarrow{\cup} \mathbb{F}_q$) and z'_i distribute identically. Since the distribution $\mathbf{R}(d_{E^2}(\vec{x}, \vec{y}))$ does not contain any information on $\vec{x}$ or $\vec{y}$ except for $d_{E^2}(\vec{x}, \vec{y})$, $\mathcal{D}$ cannot obtain any partial information on $\vec{x}$ and $\vec{y}$ except for $d_{E^2}(\vec{x}, \vec{y})$.

Editor's Recommendation

The paper proposes a novel biometric authentication protocol which allows revoke identity without revealing confidential biometric information. The proposed scheme allows evaluating 2-DNF (disjunctive normal form) predicate on encrypted feature

vectors in secure way, based on Boneh-Goh-Nissim cryptosystem the Okamoto-Takashima cryptosystem. The fundamental scheme, comparison of encrypted vectors without decrypting, is general enough and thus can be applied to a variety of targets. The paper is outstanding in terms of quality and gives insights to broad range of readers.

(Chairman of SIGCSEC Hiroaki Kikuchi)

Mitsuhiro Hattori received his B.E and M.E. degrees from Kyoto University, Japan, in 2003 and 2005, respectively. He joined Mitsubishi Electric Corporation in 2005, and since then he has been engaged in the research and development of information security. He was awarded the DICOMO 2010 best paper prize. He is a member of IEICE and IPSJ.

Nori Matsuda received his B.E and M.E. degrees from Chuo University, Japan, in 1995 and 1997, respectively. He joined Mitsubishi Electric Corporation in 1997, and since then he has been engaged in the research and development of information security. He was awarded the SCIS 1996 paper prize. He is a member of IEICE.

Takashi Ito received his B.E. and M.E. degrees from the University of Tokyo, Japan, in 2000 and 2002, respectively. He joined Mitsubishi Electric Corporation in 2002, and since then he has been engaged in the research and development of information security. He is a member of IPSJ.

Yoichi Shibata received his B.I, M.I. and Ph.D. degrees from Shizuoka University, Japan, in 2003, 2005 and 2008, respectively. He joined Mitsubishi Electric Corporation in 2008, and since then he has been engaged in the research and development of information security. He is a member of IPSJ.

Katsuyuki Takashima received his B.S., M.S. and Ph.D. degrees from Kyoto University, Japan, in 1993, 1995 and 2009, respectively. He is presently engaged in the research and development of information security at Mitsubishi Electric Corporation. He was awarded the SCIS 2000 paper prize and 2003 annual award of JSIAM papers. He is a member of IEICE, IPSJ, JSIAM and IACR.

Takeshi Yoneda received his B.E, M.E. and Ph.D. degrees from Keio University, Japan, in 1989, 1991 and 1994, respectively. He joined Mitsubishi Electric Corporation in 1994, and since then he has been engaged in the research and development of information security. He is a member of IPSJ and ACM.