RESEARCH ARTICLE

Argument on biometrics identity-based encryption schemes

Syh-Yuan Tan¹,², Zhe Jin² and Andrew Beng Jin Teoh³,4*

¹ Faculty of Engineering and Science, Tunku Abdul Rahman University, Perak 31900, Malaysia
² Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia
³ School of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea
⁴ Predictive Intelligence Research Cluster, Sunway University, Bandar Sunway, P.J. Selangor 46150, Malaysia

ABSTRACT

Recently, a few biometric identity-based encryption (BIO-IBE) schemes have been proposed. BIO-IBE leverages both fuzzy extractor and Lagrange polynomial to extract biometric feature as a user public key and as a preventive measure of collusion attack, respectively. In this paper, we reveal that BIO-IBE is not realistic whereby a query of fresh biometrics is needed for each encryption process. Moreover, the use of both fuzzy extractor and Lagrange polynomial in BIO-IBE simultaneously is a redundancy; it confers no advantage, but simply computational overhead. Therefore, we amend the progression of the BIO-IBE scheme by eliminating either Lagrange polynomial or fuzzy extractor to alleviate computational complexity. Subsequently, we demonstrate that the amendment does not compromise the security of the BIO-IBE scheme. Such amendments can be applied to other BIO-IBE schemes as well. Copyright © 2013 John Wiley & Sons, Ltd.

KEYWORDS

fuzzy extractor; biometric; identity-based encryption; Lagrange polynomial

*Correspondence

Andrew Teoh, School of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea.
E-mail: bjteoh@yonsei.ac.kr

1. INTRODUCTION

1.1. Backgrounds

The idea of fuzzy identity-based encryption (FIBE) was introduced by Sahai and Waters [1] as an extension 20 years after IBE [2]. The initial motivation of Sahai and Water’s FIBE (SW-FIBE) is to leverage noisy data such as biometric identity in encryption process. In other words, SW-FIBE can tolerate the errors between the enrollment identity ID and query identity ID'. This property adapted well to biometrics, which inherently suffers from distortion by the noises when biometric feature are captured each time. Although the scheme was not defined rigorously, SW-FIBE views ID as a set of extracted biometric features μ, and the distance between two IDs can be evaluated by using number of matches among their sets [1]. Decryption is deemed successful if the query biometric identity ID' of a user is close to the enrolled biometric identity, ID, specified on the ciphertext for certain distance such that |ID ∩ ID'| ≥ d where d is a threshold value.

As suggested by SW-FIBE, a naive way of constructing FIBE is to apply multiple user public identities (multi-ID) setting in IBE. In multi-ID IBE, in addition to generating IBE’s system parameters such as public and private keys, the setup algorithm requires to specify the threshold value d. So, a user will acquire his or her secret key uski for each of his or her identity IDi as a result of running key extraction algorithm. During encryption, the encrypter encrypts the plaintext with multiple identities. Decryption is possible only when the decrypter has at least d out of n uski corresponding to the identities in the ciphertext. It is clear that the multi-ID setting can be easily adopted by any IBE while preserving the existing security properties. However, this setting creates a security problem, namely collusion attack [1].

For an example, a multi-ID IBE algorithm fixes a threshold value d=3, and a ciphertext is generated for user C. Assume that the extracted biometric data of user A IDA = [1, 2, 3, 4, 5], user B IDB = [6, 7, 8, 9, 10], and user C IDc = [4, 5, 6, 11, 12]. Because each uski is bound with the ID independently, users A and B can combine their uski sets to generate new sets of secret keys corresponding to the identity IDA ∪ IDB = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. As |IDA ∪ IDB| = 3 = d, users A and B can decrypt the ciphertext...
of user C just by colluding their user secret keys. Therefore, the role of Lagrange polynomial (LP) in SW-FIBE is to bind elements \( \mu_i \in ID \) in key extraction to a randomly chosen secret polynomial (in \( usk \)) in order to avoid the collusion attack. With this protection, when the dishonest users A and B collude their \( usk \), they cannot decrypt the ciphertext as the \( usk \) obtained from polynomial interpolation is not the same as that of the victim C. Precisely, the \( i \)th component of a \( usk \) will be evaluated at the point \( \mu_i \in ID \) of the random LP and thus avoid the collusion attack.

Another approach of integrating biometrics and cryptography, namely fuzzy extractor (FE), was reported by Dodis et al. [3]. FE allows error correction and generates strong cryptographic keys from noisy data such as biometrics. The cryptographic key in FE refers to the private key used in some encryption schemes, such as Advanced Encryption Standard (AES), instead of public key encryption schemes such as FIBE. Inspired by FE and FIBE, a new public key encryption scheme, namely biometric identity-based encryption (BIO-IBE) was proposed. Up to date, only few BIO-IBE schemes [4–7] appeared in the literature. In general, BIO-IBE uses FE to generate user public identity \( ID = \) Hash(\( b \)) where \( b \) is the set of extracted biometric features such that \( b = \{ \mu_1, \mu_2, \ldots, \mu_n \} \). To avoid collusion attack, BIO-IBE applies LP during encryption [4–6, 31] or during user secret key extraction [7].

BIO-IBE in [4] works slightly different in such a way that it applies LP (in addition to FE) during \( usk \) extraction and decryption. Because FE is applied, the errors of biometric features will be corrected, and thus, always the same user public identity \( ID \) is generated for BIO-IBE. The condition to activate the correction, however, is similar to FIBE where the error correction will be performed only if \( |b \cap b'| \geq d \) for a threshold value \( d \), where \( b \) is the enrolled biometric data and \( b' \) is the query biometric data.

The difference between the techniques of FIBE (which uses LP) and FE is that the former tolerates errors, whereas the latter corrects errors. However, it is a redundancy to use LP and FE together in BIO-IBE scheme where no errors exist for LP to tolerate with, because all the errors have been corrected by FE and vice versa. In fact, the basic construct of FE is an error correcting code that is using polynomials also. For instance, error correction codes such as BCH [11,12] codes are applied on the resulting points (of the biometric data \( b \) on LP) into a single-value \( ID \).

### 2. PRELIMINARIES

#### 2.1. Bilinear maps

**Definition 1.** A bilinear maps is a function \( e \) such that \( e : G \times G \rightarrow G_T \) where \( G \) and \( G_T \) are two multiplication cyclic groups of prime order \( p \). Let \( g \) be a generator of \( G \); the bilinear map \( e \) has the following properties:

1. **Bilinearity:** \( e(g^a, g^b) = e(g, g)^{ab} = e(g^a, g^b) \) for all \( a, b \in \mathbb{Z}_q \).
2. **Non-degeneracy:** \( e(g,g) \neq 1 \)
3. \( e \) is efficiently computable.

#### 2.2. Biometric performance metrics

**Definition 2.** False rejection rate

False rejection rate (FRR) refers to the probability that the system fails to detect a match between the input pattern and a matching template in the database. It measures the percent of valid inputs that are incorrectly rejected [9]. FRR can be formulated as follows:

\[
\text{FRR} = \frac{\text{number of rejected genuine users}}{\text{total number of genuine access}} \quad (1)
\]

**Definition 3.** False acceptance rate

False acceptance rate (FAR) refers to the probability that the system incorrectly matches the input pattern to a non-matching template in the database. It measures the percent of invalid inputs that are incorrectly accepted [10]. FAR can be formulated as follows:

\[
\text{FAR} = \frac{\text{number of accepted imposter}}{\text{total number of imposter access}} \quad (2)
\]

We note that as a cryptosystem, IBE requires zero risk of intrusion. Hence, FAR should be strictly controlled at 0%, whereas FRR can be within a certain degree of accepted inconvenient level. It is also noted that high FRR implies poor usability and otherwise.

### 1.3. Organization

The rest of the paper is organized as follows. We first provide the preliminaries and mathematical definitions in Section 2. Secondly, we show the redundancy of Sarier’s BIO-IBE [4,5,8] in Section 3, followed by the discussions in Section 4. The conclusion is provided in Section 5.

**1.2. Contribution**

In this paper, we revise the algorithm flow of BIO-IBE in order to rule out the unrealistic requirement of encryption. Furthermore, we reveal that the use of LP and FE together in BIO-IBE is inappropriate as it increases the algorithm complexity without gaining any benefit. As such, we propose a redundancy removal technique for a BIO-IBE [4] scheme and show that the security is not compromised. This technique can be applied on other BIO-IBE [5,6,8,31] schemes as well.
2.3. Lagrange coefficient

Definition 4. The Lagrange coefficient $\Delta_{i,j}$ is defined as

$$\Delta_{i,j}(x) = \prod_{j \in S, j \neq i} \frac{x - j}{i - j}$$  \hspace{1cm} (3)

where $i$ and $j$ are elements in the set $S$.

Let $q(.)$ be a random $(d-1)$-degree polynomial whose coefficients $c_1, c_2, \ldots, c_{d-1} \in \mathbb{Z}_q^*$ and $q(i_j) = s_{i_j}$; the polynomial can be reconstructed by having the knowledge of $d$-pair values of $(i_1, s_1)$ such that

$$q(.) = \sum_{i=0}^{d-1} s_i \Delta_{i,S}(\cdot)$$  \hspace{1cm} (4)

where $S = \{i_0, i_1, \ldots, i_{d-1}\}$. So, the value $d$ can be used as the threshold, which controls FRR and FAR. If the threshold $d$ is small enough, the impostor could be possibly accepted, and subsequently, the FAR will increase. In other words, a threshold $d$ is inversely proportional to FAR. Therefore, in order to be intrusion-free, FAR should be restricted to a very low level, such as $\approx 0\%$ with a proper threshold value $d$, which is determined experimentally. Consequently, with this selected threshold value, all impostor attempts will be blocked.

2.4. Fuzzy extractor

Definition 5. An FE consists of two efficient randomized algorithms $(Gen, Rep)$ (“generate” and “reproduce”) [3,5]:

(1) $Gen$ takes enroll biometric feature $w \in M$ as input where $M$ is a metric space with distance function $dis(.)$ whose threshold value is $(n - d)$ where $n$ is the size of $w$ and $d$ is the number of matched elements. It outputs a random string $R \in \{0,1\}^l$ and a helper string $PAR = b \oplus C_e(R)$ where $C_e$ is a one-to-one encoding function.

(2) $Rep$ takes a query biometric feature $w' \in M$ to compute $R' = C_d(b' \oplus PAR) = C_d(b \oplus b' \oplus C_e(R))$ and $R = R'$ if $dis(h,b') \leq (n - d)$. Here, $C_d$ is the decoding function that corrects the errors up to the threshold $(n - d)$.

Similar to FIBE, the threshold $(n - d)$ needs to be determined carefully to exclude all imposters.

2.5. Identity-based encryption model

Definition 6. An IBE scheme consists of four probabilistic polynomial time algorithms, namely setup, extract, encrypt, and decrypt [3]:

(1) $Setup(\cdot)$: Private key generator (PKG) runs this algorithm to generate its master public key $mpk$ and the master secret key $msk$. PKG publicizes $mpk$ and keeps $msk$ to itself.

(2) $Extract(msk, ID)$: SARK runs this algorithm to generate an identity $u$ which is bound to $ID$. SARK assigns $u$ to the user.

(3) $Encrypt(mpk, ID', M)$: A user runs this algorithm to generate a ciphertext $C$ which takes $mpk$ and public identity $ID'$, and the plaintext $M$ as input.

(4) $Decrypt(mpk, usk, C)$: A user decrypts the ciphertext $C$, which is encrypted under the identity $ID'$, and the result is either a plaintext $M$ or a Reject message.

2.6. Fuzzy identity-based encryption model

Definition 7. A FIBE scheme consists of four probabilistic polynomial time algorithms, namely setup, extract, encrypt, and decrypt [1]:

(1) $Setup(\cdot)$: PKG runs this algorithm to generate its master public key $mpk$, which contains an eligible error tolerance threshold $d$, and the master secret key $msk$. PKG publicizes $mpk$ and keeps $msk$ to itself.

(2) $Extract(msk, ID)$: Take $msk$ as input and an enrolled biometric identity $ID$; PKG runs this algorithm to generate a user secret key, $usk$, which is bound to $ID$. PKG assigns $usk$ to the user.

(3) $Encrypt(mpk, ID', M)$: A user runs this algorithm to generate a ciphertext $C$, which takes $mpk$, query biometric identity $ID'$, and the plaintext $M$ as input.

(4) $Decrypt(mpk, usk, C)$: A user decrypts the ciphertext $C$, which is encrypted under the query biometric identity $ID'$, and the result is either a plaintext $M$ or a Reject message.

3. SARIER’S BIO-IBE

We outline SARIER’s BIO-IBE [4] as follows before showing the redundancy of FE or LP in the scheme:

(1) $Setup(k)$: On input of $1^k$, generate a group $G$ of prime order $q$. Pick a random generator $g \in G$ and a random value $s \in Z_q^*$ to compute $P_{pub} = g^s$ and $e$ $(g,g)$. Define for FE the encoding function $C_e$ and the decoding function $C_d$ along with the feature extraction method $F_e$, which produces biometric feature $b$. Define three cryptographic hash functions $H: b \{0,1\}^* \rightarrow Z_q^*$, $H_2: G_T \{0,1\}^l \rightarrow Z_q^*$, and $H_3: G_T \{0,1\}^l$ where $MSg = \{0,1\}^l$ denotes the message space. Let $d$ be the threshold of FE and $t$ be the threshold of LP. The master public key is $mpk = (q, g, e, d, l, t, G, G_T, H_1, H_2, P_{pub}, e(g,g), C_e, C_d, F_e)$ and the master secret key is $msk = x$. 
(2) **Extract**(msk, b): An enrolled biometric data w are obtained from a biometric reader and the feature extractor \( F_c \). Each element \( \mu_i \in w \) is associated to a unique integer in \( \mathbb{Z}_n^* \). Take msk as input; let b be the binarized value of w; compute the biometric identity, \( ID = H(b) \), and the corresponding helper string \( PAR = b \oplus C_n(ID) \) using FE. Next, calculate \( D^D_{\mu_i} = g^{1/(x+H_i((w,ID)))} = g^{1/(x+b^m)} \) for each \( \mu_i \in w \) and assign \( \text{usk} = \{ D^D_{\mu_i} \} \) to the user.

(3) **Encrypt**(mpk, b’, M): Take mpk; query biometric data \( w' \), PAR, and a plaintext \( M \) as input; calculate \( ID' = \text{Rep}(b', \text{PAR})=ID \) where \( b' \) is the binarized value of \( w' \). Pick a random polynomial \( r(\cdot) \) of degree \( d - 1 \) over \( Z_q^* \) such that \( r(0)= s \in Z_q^* \). Compute the shares \( r(\mu_i) = r_i \in Z_q^* \) and \( L_i = P_{\text{pub}}'g^{H_i((\mu_i,ID'))} = g^{(x+b^m)} \) for \( \mu_i \in w' \). Set the ciphertext to \( C = \{ w'.L_i', W \} \) where \( V=H_2(e(g,g)^r) \) and \( W = M \oplus V \).

(4) **Decrypt**(mpk, usk, C): Given \( C = \{ w'.L_i', W \} \), select an arbitrary set \( S \subseteq w \cap w' \) such that \( |S|=d \). For every \( \mu_i \in S, h^D_{\mu_i} = h^D_{m(\mu_i)} \) and the plaintext \( M = W \) can be recovered with value of \( V \) as follows:

\[
V = H_2 \left( \prod_{\mu_i} e \left( L_i', D^D_{\mu_i} \right)^{\Delta_{w,s}(0)} \right) \\
= H_2 \left( \prod_{\mu_i} e \left( g^{r'}(x+b_m^\prime), g^{1/(x+b^m)} \right)^{\Delta_{w,s}(0)} \right) \\
= H_2 \left( \prod_{\mu_i} e(g,g)^{\Delta_{w,s}(0)} \right) \\
= H_2(e(g,g)^r) 
\]

In short, the aforementioned flow diagrams for Extract and Encrypt algorithms of BIO-IBE are shown in Figure 1.

### 3.1. Problems in algorithms flow

In the flow diagram of Sarier’s BIO-IBE, an impractical assumption is made whereby a fresh biometric reading \( w' \) must be obtained for every encryption. This means that the decrypter has to always stand by to provide a fresh biometric reading \( w' \) for the encrypter. The encrypter then acquires helper string \( PAR \) from the public storage and runs the reproduce algorithm Rep to compute \( ID' \) for use in the encryption process, as depicted in Figure 1. In short, in order to make this work, the decrypter has to be prepared with a biometric reader in hand as well as an Internet connection whenever the encrypter wants to generate a ciphertext to him or her.

To solve this problem, we suggest storing the enrolled biometric reading \( w \), which is used by Extract algorithm, together with the corresponding \( PAR \) in the public storage so that anyone can send a ciphertext without being restricted to the availability of the decrypter’s biometric reading. Besides, during the Decrypt algorithm, a decrypter must produce a query biometric data to claim the ownership of the corresponding \( \text{usk} \). The revised algorithm flow is shown in Figure 2.

However, redundancy still exists in the algorithms although the flow is corrected. We will now show how to remove the redundancy in the following section and discuss the solution for privacy issue of public storage in Section 4.

### 3.2. Redundancy of fuzzy extractor

We show that when the FE is removed from Sarier’s BIO-IBE, the resulting scheme resembles Sakai–Kasahara FIBE scheme, which is the SK-IBE [14,29] embedded with LP in the Encrypt and Decrypt algorithms:

(1) **Setup**\( (k) \): Similar to the algorithm in Section 3 without FE parameters.

(2) **Extract**(msk, b): The involvement of FE is excluded. An enrolled biometric data \( w \) are obtained from the raw biometric information using a reader and the feature extractor \( F_c \). Each element \( \mu_i \in w \) is associated to a unique integer in \( \mathbb{Z}_n^* \). Take msk as input; let \( b \) be the string value of \( w \) such that \( b = \text{toString}(w) \); compute the biometric identity, \( ID = H(b) \), and calculate \( D^D_{\mu_i} = g^{1/(x+H_i((w,ID)))} = g^{1/(x+b^m)} \) for each \( \mu_i \in w \). PKG returns \( \text{usk} = \{ D^D_{\mu_i} \} \) to the user.

![Figure 1. Flow diagrams of Sarier’s BIO-IBE [4].](image-url)
amendments show that a unique ID does not affect the security because it only generates is expected to be publicly known. Therefore, the removal still can be tolerated by LP during

Hbles SK-IBE [14,29] whose ID is bound with FE: from BIO-IBE instead of FE, the resulting scheme resembles SK-IBE [14,29] whose ID is bound with FE:

3.3. Redundancy of Lagrange polynomial

We now show that on the other hand, if LP is removed from BIO-IBE instead of FE, the resulting scheme resembles SK-IBE [14,29] whose ID is bound with FE:

(3) Encrypt(mpk, b, M): Acquire w and b from the public storage; let \( ID = H(b) \) and exclude the involvement of FE. Pick a random polynomial \( \gamma(t) \) of degree \( t - 1 \) over \( \mathbb{Z}_q^* \) such that \( r(0) = s \in \mathbb{Z}_q^* \) Compute the shares \( r(\mu_i) = r_i \in \mathbb{Z}_q^* \) and \( L_s = P_{pub} g^{H(\mu_i, ID)} = g^{(x+h_0)} \) for \( \mu_i \in w \). Set the ciphertext to \( C = (w, L_s^0, W) \) where \( V = H_2(e(g, g)^t) \) and \( \text{W} = M \oplus V \).

(4) Decrypt(mpk, usk, C): Given a ciphertext \( C = (w, L_s^0, W) \), select an arbitrary set \( S \subseteq w \cap w^t \) such that \( |S| = t \) where \( w^t \) is the query biometric data. For every \( \mu_i \in S, H_s(\mu_i, ID) = h_{i}^{ID} \) and the plaintext \( M = W \oplus V \) can be recovered exactly the same to the algorithm in Section 3:

\[
V = H_2 \left( \prod_{\mu_i \in S} e \left( L_s^0, D_{\mu_i}^0 \right) \Delta_{\mu_i, 0}^{0(0)} \right)
\]

\[
= H_2 \left( \prod_{\mu_i \in S} \left( e \left( g^{(x+h_0)}, g^{1/(x+h_0)} \right) \Delta_{\mu_i, 0}^{0(0)} \right) \right)
\]

\[
= H_2 \left( \prod_{\mu_i \in S} e(g, g)^{\mu_i+h_0} \right)
\]

\[
= H_2(e(g, g)^t)
\]

If the query biometric data is obtained from the genuine user, together with the valid usk, the decryption is always success with maximum \( t \) errors tolerated by the LP.

Recall that the function of FE is to hide a secret random string \( R \) using biometric data \( w \). But from the original BIO-IBE scheme and its algorithm flow, we notice that \( R \) is replaced with a public computable hash value \( H(b) = ID \). Although this may create privacy issue, it fits the concept of BIO-IBE and FIBE in which \( ID \) is a public key and it is expected to be publicly known. Therefore, the removal of FE does not affect the security because it only generates a unique \( ID \) that is bound to the corresponding \( b \). The amendments show that \( H \) alone is sufficient for the job whereby the generation of unique \( ID \) still can be carried out by computing \( ID = H(b) \). Furthermore, the security proof in [4] does not involve FE but only the hash function \( H \). With the missing of error correction of FE, the errors of \( b \) still can be tolerated by LP during Encrypt and Decrypt.

3.3. Redundancy of Lagrange polynomial

We now show that on the other hand, if LP is removed from BIO-IBE instead of FE, the resulting scheme resembles SK-IBE [14,29] whose ID is bound with FE:

(1) Setup(\( k \)): This is similar to the algorithm in Section 3 but without the LP parameters.

(2) Extract(usk, b): Enrolled biometric data \( w \) are obtained from the raw biometric information using a reader and the feature extractor \( F_e \). Each element \( \mu_i \) \( w \) is associated to a unique integer in \( \mathbb{Z}_q^* \). Take \( \text{msk} \) as input; let \( b \) be the binarized value of \( w \); compute the biometric identity, \( ID = H(b) \), and the corresponding helper string \( \text{PAR} = b \oplus C_\text{i}(ID) \) using FE. Next, calculate \( \text{usk} = D_{\text{ID}} = g^{1/(x+h_0)} \) such that \( \text{PAR} = b \oplus C_\text{i}(ID) \) using FE. Next, calculate \( \text{usk} = D_{\text{ID}} = g^{1/(x+h_0)} \) where \( \text{PAR} = b \oplus C_\text{i}(ID) \). PKG returns \( \text{usk} \) to the user.

(3) Encrypt(mpk, b, M): Take mpk and plaintext \( M \) as input, acquire \( w \) and \( b \) from the public storage, and pick a random \( s \in \mathbb{Z}_q^* \). Compute the value \( L = P_{pub} g^{H(\mu_i, ID)} = g^{(x+h_0)} \) where \( ID = H(b) \). Set the ciphertext to \( C = (w, L_s^0, W) \) where \( V = H_2(e(g, g)^t) \) and \( \text{W} = M \oplus V \).

(4) Decrypt(mpk, usk, C): Given \( C = (w, L_s^0, W) \), compute \( ID = \text{Rep}(b', \text{PAR}) = ID \) where \( b' \) is the query biometric data of the decrypter and \( \text{dis}(b, b') \leq (n - d) \). The plaintext \( M = W \oplus V \) can be obtained similar to the algorithm in Section 3:

\[
V = H_2(e(L_s^0, D_{\text{ID}}))
\]

\[
= H_2 \left( e \left( g^{1/(x+h_0)}, g^{1/(x+h_0)} \right) \right)
\]

\[
= H_2(e(g, g)^t)
\]

If the query biometric reading is obtained from the genuine user, together with the valid usk, the decryption is success with a maximum of \( d \) errors corrected by the FE.

The SK-IBE scheme presented in [29] is the stronger version (chosen ciphertext attack secure) of the original version [14] (chosen plaintext attack secure) using Fujisaki-Okamoto transformation [30]. This transformation requires extra two hash functions to perform the security upgrade. Notice that if these extra two hash functions in [29] are removed, its Encrypt and Decrypt algorithms are the same as the one in this section. Although such removal yields a chosen plaintext attack secure SK-IBE only, we argue that the original BIO-IBE is also proven secure under the same security level [4]. The only difference is that SK-IBE of [29] views user public identity as a string value, whereas BIO-IBE views it as a list of string values here. Thus, the removal of LP does not affect
BIO-IBE’s security as $b$ of every user is corrected and resulted in a single value by FE, which revert the BIO-IBE back to its building block, SK-IBE. Some may worry that the binding of $b$ to ID in FE instead of binding to $usk$ will threaten the security of Decrypt algorithm. Note that publishing $ID$ only creates privacy issue, which is not related to the cryptographic security of the underneath encryption scheme. The decryption will fail as long as $usk$ is out of the reach of attackers.

4. DISCUSSION

Because the main ingredient of error encoding code is LP, to ease our calculation, we assume that the complexity of $Gen$ and $Rep$ are at most the same as computing a polynomial and the Lagrange coefficient, respectively. Referring to the mathematical operations timing from [22,23], let $d=t=100$ with $lw=n>100$ and set the complexity of BIO-IBE as 1; we can calculate the complexity ratio of IBE-FE’s algorithms with respect to BIO-IBE as follows: Setup:

$$R = \frac{1}{R} = \frac{T_e + T_p}{T_e + T_p}$$

Extract:

$$R = \frac{1}{R} = \frac{n(T_e + T_a + T_h + T_f + T_{poly})}{n(T_e + T_a + T_h + T_f + T_{poly})}$$

Let $x = (T_e + T_a) \approx T_{poly}$,

$$R \approx \frac{x + 0 + 0 + x + x}{n(x + 0 + 0) + x + x}$$

Encrypt:

$$R \approx \frac{1}{R} = \frac{n(2T_e + T_m + T_{poly} + T_h + T_{e} + F_e + T_{Lag})}{2(T_e + T_h + T_m + T_a + T_e)}$$

Let $x = F_e \approx T_{poly} \approx T_e \approx T_{Lag}$,

$$R \approx \frac{2(T_e + 0) + 0 + 0 + x}{n(2x + 0) + x + 0 + 0 + x + x}$$

Decrypt:

$$R \approx \frac{1}{R} = \frac{r(T_e + T_p) + T_h + (r - 1)T_m + T_{Lag}}{T_e + T_h + T_p + F_e + T_{Lag}}$$

Let $x = F_e \approx T_e \approx T_{Lag}$ and $T_p \approx 2T_e$,

$$R \approx \frac{r(0 + 2x) + 0 + (r - 1)0 + x}{x + 0 + 2x + x + x}$$

$$R \approx \frac{x}{(2r+1)x} = 0.02$$

Note that these ratios are applicable on BIO-IBE of SK-IBE [14,29] instance only. However, the ratios will remain the same despite the implementation platform in practice, and it would be a good indicator for the amount of complexity reduced. By repeating the similar calculations for IBE-LP, the results are summarized in Table I.

Table I shows that the complexity of the original BIO-IBE is greatly reduced especially when LP is removed. The puzzle left now is to decide whether to remove LP or FE if the privacy issue of biometric data is a concern.

Recall that biometric data is used as public key in BIO-IBE, and thus, if a user’s biometric data is known through the public storage, it does not affect the operation of BIO-IBE. In fact, BIO-IBE does not work at all if one’s biometric data is not publicized. However, in this paper, for the completeness of security, we do consider the protection of biometric data in the public storage. This problem is resolvable using biometric template protection techniques, which can be classified into four categories, namely biometric salting, non-invertible transform, key binding, and key generation [13]. Biometric salting and non-invertible transform can be generalized as feature transformation, and their instances are reported in literature such as [15,17,18], whereas key binding and key generation fall into the category bio-cryptography, and the corresponding examples in literature are [16,19,20].

In order to further reduce users’ privacy invasion concern, the preferable type of user biometric should be those that can be acquired easily such as fingerprint, palmprint, face, and voice. In short, the employment of template protection techniques should be included as a basic requirement of BIO-IBE and FIBE.

From Table I, we can see that IBE + FE is faster than IBE + LP in all algorithms. Furthermore, FE itself is already a template protector whereby, given the public parameter $PAR$, it is infeasible to reverse engineer $PAR$ to the biometric template $w$. On the basis of these facts, IBE + FE should be a better choice than IBE + LP even when the privacy issue is considered. However, as mentioned in Section 3.2, all the secret parameters of FE have been made public, for the purpose of fuzzily generating a public key $ID$, thus rendering the template protection feature of FE useless.

Similar to IBE + FE, IBE + LP does not come with any protection mechanism for their biometric template but this can be solved by applying template protection techniques. Because FE and LP are based on polynomials, the constraints on the techniques are the same to that of
Table I. Complexity of BIO-IBE, IBE + LP, and IBE + FE.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>BIO-IBE</th>
<th>IBE + LP</th>
<th>IBE + FE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of ( u_{sk} )</td>
<td>( n</td>
<td>G</td>
<td>)</td>
</tr>
<tr>
<td>Size of ( C )</td>
<td>( n</td>
<td>G</td>
<td>+ 1 )</td>
</tr>
<tr>
<td>Complexity</td>
<td>( T_o + T_p )</td>
<td>( T_o + T_p )</td>
<td>( T_o + T_p )</td>
</tr>
<tr>
<td>Setup</td>
<td>( nT_o + (n + 1)T_h + nT_p + T_{poly} )</td>
<td>( nT_o + T_p )</td>
<td>( T_o + T_p )</td>
</tr>
<tr>
<td>Extract</td>
<td>( nT_o + (n + 1)T_h + nT_p + T_{poly} )</td>
<td>( nT_o + T_p )</td>
<td>( T_o + T_p )</td>
</tr>
<tr>
<td>Encrypt</td>
<td>( nT_o + (n + 1)T_h + T_p + T_{poly} )</td>
<td>( 2T_o + nT_h + T_p + T_{poly} )</td>
<td>( 2T_o + nT_h + T_p + T_{poly} )</td>
</tr>
<tr>
<td>Decrypt</td>
<td>( nT_o + (n + 1)T_h + T_p + T_{poly} )</td>
<td>( nT_o + T_p )</td>
<td>( T_o + T_p )</td>
</tr>
<tr>
<td>Ratio</td>
<td>( nT_o + (n + 1)T_h + nT_p + T_{poly} )</td>
<td>( nT_o + T_p )</td>
<td>( T_o + T_p )</td>
</tr>
<tr>
<td>Setup</td>
<td>( 1 )</td>
<td>( 1 )</td>
<td>( 1 )</td>
</tr>
<tr>
<td>Extract</td>
<td>( 1 )</td>
<td>( n + 1/n + 2 )</td>
<td>( 1/n )</td>
</tr>
<tr>
<td>Encrypt</td>
<td>( 1 )</td>
<td>( 2n + 1/2n + 3 )</td>
<td>( 2/2n + 3 )</td>
</tr>
<tr>
<td>Decrypt</td>
<td>( 1 )</td>
<td>( 1 )</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Timing notation:
- \( T_{exp} \): exponentation in \( G \);
- \( T_{pair} \): pairing in \( G \);
- \( T_{add} \): addition in \( Z_q \);
- \( T_{inv} \): inversion in \( Z_q \);
- \( T_{mul} \): multiplication in \( G \);
- \( T_{ex} \): exponentiation in \( G \);
- \( T_{hash} \): hash to \( Z_q \);
- \( T_{lag} \): compute Lagrange coefficient;
- \( T_{poly} \): compute \( t-1 \) degree polynomial \( r(x) \) in \( Z_q \).

Table II. Properties of BIO-IBE, IBE + LP, and IBE + FE.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>BIO-IBE</th>
<th>IBE + LP</th>
<th>IBE + FE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avoid collision attack</td>
<td>Bind ( w ) to FE + LP</td>
<td>Bind ( w ) to LP</td>
<td>Bind ( w ) to FE</td>
</tr>
<tr>
<td>Redundancy</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Require fresh ( w ) to encrypt?</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Constraint of biometrics data</td>
<td>Ordered, fixed length, binary/integer</td>
<td>Ordered, fixed length, binary/integer</td>
<td>Ordered, fixed length, binary/integer</td>
</tr>
<tr>
<td>Algorithm complexity</td>
<td>Highest</td>
<td>Medium</td>
<td>Lowest</td>
</tr>
<tr>
<td>Privacy issue of ( w )</td>
<td>Not a concern</td>
<td>Protect ( w ) with feature transformation</td>
<td></td>
</tr>
</tbody>
</table>

FIBE [21]. The resulting biometric trait \( w \) from the template protection scheme must be ordered and have a fixed length in which the elements are either in integers or binary form. The instances that can satisfy these requirements are [24–28]. The index of each \( \mu_i \in w \) will be used to generate the random polynomial \( r() \) during encryption, whereas the binarized value \( b \) will be fed into \( H \) to compute \( H(b) = ID \).

Note that the addition of the template protection scheme will increase the overall complexity of every BIO-IBE/FIBE scheme. Thus, the true comparison should be based on the BIO-IBE/FIBE scheme without considering template protection add-on as indicated in Table I. The properties of each BIO-IBE are as depicted in Table II, and obviously, IBE + FE is the best choice among all.

5. CONCLUSION

We showed that BIO-IBE schemes in the literature [4,5,8] are not well defined yet by fixing the unpractical algorithm flow. In addition, redundancy occurred with the adoption of both FE and LP. The former corrects errors whereas the latter tolerates errors in user biometric data. When they are used together, conflict happens in either case where there is no error to be corrected or there is no error to be tolerated. The new algorithm flow and the redundancy removal technique can be applied in the similar way on other BIO-IBE schemes as well.

ACKNOWLEDGEMENT

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (no. 2011-8-1095).

REFERENCES


