Securing Practical Packing-based Privacy-preserving Biometric Authentication

Takumi Matsunaga, Kentaroh Toyoda and Iwao Sasase
Dept. of Information and Computer Science, Keio University
3-14-1 Hiyoshi, Kohoku, Yokohama, Kanagawa, 223-8522 Japan,
Email: matsunaga@sasase.ics.keio.ac.jp

***Abstract*—Privacy-preserving biometric authentication is getting attractive. Recently, a not only privacy-preserving but also fast scheme has been proposed which is preferable for the real system. However, we point out that it has a flaw that an attacker can be authenticated with high probability. In this paper, we propose a secure practical packing-based privacy-preserving biometric authentication scheme which reduces the attack success rate. Our authentication scheme consists of three phases. At the first phase, the challenge feature vector is verified by the previous Hamming distance-based verification. In order to handle this vulnerability, we propose two more verifications which we call partial comparison and shift-then-inner product schemes. By combining three verifications, our scheme is secure against several attacking strategies. We evaluate the attack success rate, the true positive rate and the false positive rate with an iris dataset. As a result, we show that our proposed scheme can significantly reduce lower the attack success rate than the false positive rate while suppressively decreasing the true positive rate.**

I. Introduction

The authentication is utilized so as to prevent a malicious user from impersonating a legitimate user. Especially, the biometric authentication is getting attractive since users neither need to remember the password nor carry a physical object such as an IC card. The biometric authentication uses the bio-information, e.g., iris, fingerprint and palmprint. However, the bio-information is sensitive and must be securely stored so that the usable situation is limited to a local system, e.g., intra company. In order to get rid of this limitation and to make the scheme work in the cloud system, privacy-preserving biometric authentication schemes that use homomorphic encryption have been extensively proposed, e.g., [1]–[3]. Here, homomorphic encryption is the encryption scheme where arithmetic operations can be calculated over a ciphertext. We show authentication phase in privacy-preserving biometric authentication. Firstly, the bio-information is translated into a binary code which is called a feature vector by irreversible conversion [4]–[6]. Secondly, each element in a feature vector is encrypted with homomorphic encryption. A prover is successfully authenticated if and only if the Hamming distance between an enrolled and challenge feature vectors.

Although many authentication schemes have been proposed, a scheme proposed by Yasuda et al. is not only privacy-preserving but also fast which is an important feature required in authentication [7]. They proposed the practical packing method which translates a n-dimensional feature vector into a polynomial and encrypts it with homomorphic encryption. As a result, the number of encryption is reduced to one from n where a number of elements in a feature vector.

However, this scheme has a security issue that an attacker can be authenticated to arbitrary users with high probability. If $a_0 = 1$ and $b_0 = n/2$, the Hamming distance is $n/2 + n/2 - n = 0 < threshold$, so that an attacker can succeed in an attack. Unfortunately, the authentication system cannot check whether the elements of feature vectors are either 0 or 1 because they are encrypted. That is to say, the conventional scheme is strong against an attack with a binary vector, however, it is weak against an attack with a forged vector. Recently, In [8], Mandal et al. proposed a challenge-response authentication mechanism and add this scheme to [7]. Nevertheless, the scheme needs certificate authorities. Izu et al. proposed a scheme to detect whether each element satisfies either 0 or 1 [9]. However, the scheme cannot be adapted to the scheme [7] because the method calculates each element. Therefore, it is important to propose a fast and secure privacy-preserving biometric authentication scheme against such an attacker.

In this paper, we propose a secure privacy-preserving biometric authentication scheme which reduces the attack success rate. Our authentication scheme consists of three phases. At the first phase, the challenge feature vector is verified by the previous Hamming distance-based verification. In order to handle this vulnerability, we propose two more verifications which we call partial comparison and shift-then-inner product schemes. This modification is secure against the above attacker, though it is vulnerable against another forged feature vector. However, this attack can be avoided at the Hamming distance-based verification at first step. Since the attacker cannot attempt several attacking patterns in a time, attack success rate can be decreased. We evaluate the attack success rate, the true positive rate and the false positive rate with an iris dataset. We show that our proposed scheme can significantly reduce the attack success rate without increasing the false positive rate.

The main contributions of this paper are two folds: (1) the vulnerability and its remedy in the conventional scheme are shown. (2) we show that the proposed scheme can significantly reduce the attack success rate without increasing the false positive rate.

The organization of the paper is as follows: we show the preliminaries in Section II. In Section III, we introduce the of conventional scheme. In Section IV, the proposed scheme is presented. In Section V, he performance evaluation is shown. In Section VI, we conclude the paper.

II. Preliminaries

In this section, we describe the system model, the attacker model and the basic construction of a homomorphic encryption used in the conventional and proposed schemes [10]. The homomorphic encryption scheme is public key encryption scheme and based on the polynomial LWE (Learning With Error) assumption [11].

A. System Model

We describe the system model assumed the conventional and proposed schemes. The model is valid when the feature vector is binary vector and the probability of any element that takes 1 is assumed to be $1/2$. We assume three entities, which are terminal $\mathcal{T}$, computation server $\mathcal{S}$, and decryption server $\mathcal{D}$. $\mathcal{T}$ is a device which can scan biometric information of a client, and also translates and encrypts bio-information. $\mathcal{S}$ stores encrypted bio-information and computes Hamming distance in the encrypted state. $\mathcal{D}$ decrypts a computation result and judges whether the authentication is successful.

B. Attacker Model

An attacker tries to pass the authentication by a forged feature vector which consists of not only 0 or 1 but also large numbers. In this system model, $\mathcal{S}$ cannot check the elements of feature vectors are either 0 or 1 because they are encrypted. An attacker can encrypt the forged feature vector, can send it to the computation server $\mathcal{S}$ and can pass the authentication only the biometric authentication.

C. Homomorphic Encryption

Parameters

- n: n is the degree of a cyclotomic polynomial $f(x) = x^n + 1$ and is an integer of 2-power, which defines the base ring $R := \mathbb{Z}[x]/\langle f(x)\rangle$, where $:=$, $\mathbb{Z}[x]$ and $\langle f(x)\rangle$ denote definition, polynomials over integer and subgroup generated by $f(x)$, respectively.
- q: q is a prime number and satisfies $q \equiv 1 \pmod{2n}$, which defines the base ring $R_q := R/qR = \mathbb{F}_q[x]/\langle f(x)\rangle$ for ciphertext space, where R_q and $\mathbb{F}_q[x]$ denote residue ring modulo q and polynomials over finite field whose order is q, respectively.
- t: t is an integer satisfying $t < q$ and is used for a plaintext space $R_t := \mathbb{F}_t[x]/\langle f(x)\rangle$.
- σ: σ is a parameter of a standard distribution to define a discrete Gaussian error distribution $\chi = D_{\mathbb{Z}^n,\sigma}$.

Key Generation The decryption server $\mathcal{D}$ selects an element $s \leftarrow \chi$, and then sample a random element $p_1 \leftarrow R_q$ and error $e \leftarrow \chi$. $s \leftarrow \chi$ denotes that s is element which is sampled from χ. $\mathcal{D}$ sets a public key $\mathsf{pk} := (p_0, p_1)$, where $p_0 = -(p_1 s + te)$ and a secret key $\mathsf{sk} := s$.

Encryption With $\mathsf{pk} = (p_0, p_1)$ and three randomly chosen elements $u, f, g \leftarrow \chi$, the terminal $\mathcal{T}$ encrypts a plaintext $m \in R_t$ as

$$\begin{aligned} \mathsf{Enc}_{\mathsf{pk}}(m) &= (c_0, c_1) \\ &:= (p_0 u + tg + m, p_1 u + tf) = \mathsf{ct}, \end{aligned} \quad (1)$$

where ct denotes a ciphertext and $\mathsf{Enc}_{\mathsf{pk}}(m)$ denotes that m is encrypted with pk.

Decryption With a ciphertext $\mathsf{ct} = (c_0, \ldots, c_\xi)$ (note that the homomorphic multiplication makes the ciphertext length longer) and a secret key $\mathsf{sk} = s$, $\mathcal{D}$ decrypts an encrypted message m as follows.

$$\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}) = [\acute{m}]_q \bmod t \in R_t, \quad (2)$$

where $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct})$ denotes that ct is decrypted with sk and $\acute{m} = \sum_{i=0}^{\xi} c_i s^i \in R_q$.

Homomorphic Operation Let $\mathsf{ct} = (c_0, \ldots, c_\xi)$, and $\mathsf{ct}' = (c'_0, \ldots, c'_\xi)$ be two ciphertexts. $\mathcal{S}$ computes the homomorphic addition $\dot{+}$ as follows.

$$\mathsf{ct} \dot{+} \mathsf{ct}' = (c_0 + c'_0, \ldots, c_\xi + c'_\xi) \in (R_q)^{\xi+1}. \quad (3)$$

In addition, $\mathcal{S}$ computes the homomorphic multiplication $\dot{\times}$ as follows.

$$\mathsf{ct} \dot{\times} \mathsf{ct}' = (\hat{c}_0, \ldots, \hat{c}_{2\xi}), \quad (4)$$

where each elements $\hat{c}_i$ is computed as follows.

$$\sum_{i=0}^{2\xi} \hat{c}_i z^i = \left(\sum_{i=0}^{\xi} c_i z^i\right) \cdot \left(\sum_{i=0}^{\xi} c'_i z^i\right) \in R_q[z]. \quad (5)$$

III. Conventional Scheme

In this section, we explain the outline and a security issue of the biometric authentication scheme proposed by Yasuda et al. [7]. Firstly, the algorithm of the authentication phase with the practical packing method is described. We then point out the security issue of this scheme.

A. Algorithm

1) Setup Phase: In setup phase, a decryption server $\mathcal{D}$ generates public and private key pair $(\mathsf{pk}, \mathsf{sk})$. In order for a terminal $\mathcal{T}$ and a computation server $\mathcal{S}$ to use biometric authentication, they receive pk from $\mathcal{D}$.

2) Enrollment Phase: In enrollment phase, a client enrolls its bio-information in the biometric authentication system. A client enrolls its bio-information with the terminal $\mathcal{T}$. $\mathcal{T}$ scans client's biometric information, e.g., iris, which is represented as a feature vector $\boldsymbol{A} = (a_0, a_1, \ldots, a_{n-1})$, where $a_i \in \{0, 1\}$ and $i \in [0, n-1]$. It then transforms $\boldsymbol{A}$ into a polynomial $\mathsf{pm}_1(\boldsymbol{A}) = \sum_{i=0}^{n-1} a_i x^i$ in ascending order, i.e., it translates a n-dimensional feature vector into a polynomial,in order to decrease the number of encryption from n to 2 or 3 for n-dimensional feature vector. After that, It gets a packed ciphertext $\mathsf{ct}_1^{pack}(\boldsymbol{A}) = \mathsf{Enc}_{\mathsf{pk}}(\mathsf{pm}_1(\boldsymbol{A}))$ by encrypting $\mathsf{pm}_1(\boldsymbol{A})$. The detailed calculation method is described in the appendix. It sends $\mathsf{ct}_1^{pack}(\boldsymbol{A})$ to the computation server $\mathcal{S}$. $\mathcal{S}$ enrolls $\mathsf{ct}_1^{pack}(\boldsymbol{A})$ as a client's template.

3) Authentication Phase: In authentication phase, a client attempts login with his/her bio-information. A client scans his/her bio-information with the terminal $\mathcal{T}$ which is denoted as $\boldsymbol{B} = (b_0, b_1, \ldots, b_{n-1})$, where $b_i \in \{0, 1\}$ and $i \in [0, n-1]$ from the bio-information for authentication. It translates $\boldsymbol{B}$ into a polynomial $\mathsf{pm}_2(\boldsymbol{B}) = -\sum_{i=0}^{n-1} b_i x^{n-i}$ in descending order,

and gets a packed ciphertext $\mathsf{ct}_2^{pack}(\boldsymbol{B}) = \mathsf{Enc}_{\mathsf{pk}}(\mathsf{pm}_2(\boldsymbol{B}))$ by encrypting $\mathsf{pm}_2(\boldsymbol{B})$. It sends $\mathsf{ct}_2^{pack}(\boldsymbol{B})$ to the computation server $\mathcal{S}$ to compute an encrypted Hamming distance ct^{pack} between the client's template $\mathsf{ct}_1^{pack}(\boldsymbol{A})$ and the received $\mathsf{ct}_2^{pack}(\boldsymbol{B})$. $\mathcal{S}$ sends ct^{pack} to the decryption server $\mathcal{D}$ to calculate Hamming distance $d_H(\boldsymbol{A},\boldsymbol{B}) = \sum_{i=0}^{n-1} a_i + \sum_{i=0}^{n-1} b_i - 2\sum_{i=0}^{n-1} a_i b_i$, by decrypting ct^{pack}. It judges that the authentication is successful if $d_H(\boldsymbol{A},\boldsymbol{B})$ is lower than threshold θ_{hd}.

B. Merits and Security Issue

1) Merits: This scheme has two merits. The one is fast calculation. If a n-dimensional feature vector is encrypted with other schemes, e.g., [1], [3], as many as n encryptions are needed. If contrast, if a n-dimensional feature vector is encrypted with the practical packing method, only a few encryption is needed. The other one is the authentication hardly succeeds even if a template is stolen. Let us consider an attacker that can steal $\mathsf{ct}_1^{pack}(\boldsymbol{A})$ from the computation server $\mathcal{S}$. When an attacker sends it to $\mathcal{S}$ instead of $\mathsf{ct}_2^{pack}(\boldsymbol{B})$ in the authentication phase, the authentication fails with very high probability since $\mathsf{ct}_1^{pack}(\boldsymbol{A})$ is computed with feature vector of ascending order, whereas $\mathsf{ct}_2^{pack}(\boldsymbol{B})$ is computed with that of descending order. In order to succeed the authentication, an attacker needs to use not $\mathsf{ct}_1^{pack}(\boldsymbol{A})$ but $\mathsf{ct}_2^{pack}(\boldsymbol{A})$.

2) Security Issue: We discuss the security issue of the conventional scheme. An attacker can acquire a public key pk used in the biometric authentication system and access the calculation server $\mathcal{S}$. Moreover, since the biometric authentication system needs to release the number of elements n, the attacker knows the number of elements n as well. Therefore, an attacker might be able to impersonate arbitrary users by sending a forged feature vector. More specifically, an attacker creates a forged vector $\tilde{\boldsymbol{B}} = (\tilde{b}_0, 0, \ldots, 0)$, where $\tilde{b}_0 = \beta$. Here, β is the expected summation value of $\boldsymbol{A}$. The attacker translates $\tilde{\boldsymbol{B}}$ into a polynomial $\mathsf{pm}_2(\tilde{\boldsymbol{B}})$, encrypts this polynomial to generate a ciphertext $\mathsf{ct}_2^{pack}(\tilde{\boldsymbol{B}})$ and sends this ciphertext to the calculation server $\mathcal{S}$. The decryption server $\mathcal{D}$ calculates the Hamming distance $d_H(\boldsymbol{A},\tilde{\boldsymbol{B}})$ as follows.

$$\begin{aligned} d_H(\boldsymbol{A},\tilde{\boldsymbol{B}}) &= \sum_{i=0}^{n-1} a_i + \sum_{i=0}^{n-1} \tilde{b}_i - 2\sum_{i=0}^{n-1} a_i\tilde{b}_i \\ &= \sum_{i=0}^{n-1} a_i + (1-2a_0)\beta \\ &= \begin{cases} \sum_{i=0}^{n-1} a_i - \beta & (a_0 = 1). \\ \sum_{i=0}^{n-1} a_i + \beta & (a_0 = 0). \end{cases} \end{aligned} \tag{6}$$

Here, $\sum_{i=0}^{n-1} a_i$ obeys a binomial distribution whose expected value is $n/2$ because each element is expected to take 0 or 1 equally. Hence, if an attacker sets $\tilde{b}_0 = \beta$ be $n/2$, and $a_0 = 1$, the attacker can succeed in the attack since $d_H(\boldsymbol{A},\tilde{\boldsymbol{B}})$ is approximately 0 and is always smaller than the threshold θ_{hd}. Therefore, the attacker can approximately succeed to be authenticated with the probability of $1/2$ by using $\tilde{\boldsymbol{B}}$. In this case, the calculation server $\mathcal{S}$ cannot check whether the elements of $\tilde{\boldsymbol{B}}$ is either 0 or 1 since $\tilde{\boldsymbol{B}}$ is encrypted.

IV. Proposed Scheme

Here, we propose a secure privacy-preserving biometric authentication scheme by introducing two modifications to the conventional scheme. Our authentication scheme consists of two phases. At the first phase, the challenge feature vector is verified by the previous Hamming distance-based authentication. In order to decrease the above attacker's successful rate, we propose two more verification phases which we call "partial comparison" and "shift-then-inner product". In partial comparison scheme, only randomly chosen elements are used for verification instead of entire vector every authentication. In contrast, shift-then-inner product scheme is that an enrolled feature vector is randomly shifted and then the inner product is calculated with a challenge one. Both schemes are secure against the above attacker, though it is vulnerable against an attacker who attempts with $\boldsymbol{B} = (0, 1, 0, 1, \ldots)$. However, this attack fails in the first HD-based detection. An attacker is hardly authenticated an attacker has to pass three verifications which have different features. Hence, we propose and explain the scheme which is strong against an attack with a forged vector. The above problem comes from the fact that the attacker sets a large number, e.g., $\beta = n/2$, in an attack vector.

A. Ideas to Reduce the Attack Success Rate

1) Partial Comparison: In our scheme, $\mathcal{S}$ verifies an attempt feature vector by comparing the sum of partial elements of feature vectors in order to reduce attacker's authentication success rate. When the number of division is d, the number of the partial elements is denoted as n/d. In order to authenticate with randomly chosen n/d elements of feature vectors every authentication, $\mathcal{S}$ generates a masking vector $\boldsymbol{C'} = (c'_0, c'_1, \cdots, c'_{n-1})$, whose n/d elements are 1 and the others are 0. In this case, an attacker needs to set not $n/2$ but a smaller number so as to match the sum of partial elements of feature vectors. The lower number of elements is used for authentication, i.e., the larger d is, the lower attacker's success probability gets. However, if d is excessively large, e.g., $d = n$, an attacker can succeed in the authentication if the first element matches, i.e., an attacker can use $\tilde{\boldsymbol{B}} = (\beta, 0, \ldots, 0)$.

When we set $d = 2$, the sum of half elements of $\boldsymbol{A}$ is based on binomial distribution whose mean value is $(n/2)/2 = n/4$. On the other hand, the sum of half elements of $\tilde{\boldsymbol{B}}$ is either $\tilde{b}_0 = \beta$ or 0. For instance, if an attacker sets $\tilde{b}_0 = \beta = n/2$ so as to assume that the sum of all elements is used, i.e., $C' = C = (1, 1, \cdots, 1)$, the difference between the sum values is $n/2 - n/4 = n/4$ and the attack will fail with high probability. Accordingly, an attacker partitions $\tilde{b}_0 = \beta = n/2$ into $\tilde{b}_0 = \beta/2 = n/4$ and $\tilde{b}_1 = \beta/2 = n/4$. Therefore, attack success rate is reduced since an attacker needs both $a_0 = 1$ and $a_1 = 1$ so as to pass the succeed in the attack. The attacker is hard to succeed in an attack when it sets a smaller number, e.g., binary code. Moreover, the attack success rate can be reduced by $d = d_{attack} * 2$ if we assume that an attacker partitions β into d_{attack} elements, where d_{attack} is number of division by attacker.

2) Shift-then-Inner Product: $\mathcal{S}$ calculates randomly shifts an enrolled feature vector and calculates the inner product between shifted and challenged ones so as to make the attacker set a smaller number, e.g., binary code. In the case of a legitimate feature vector, e.g., binary code, the expected inner product value is $n/4$. On the other hand, if an attacker attempts with $\tilde{\boldsymbol{B}}$, the expected inner product value is 0. As a result, $\mathcal{D}$ can judge that challenged feature vector is a legitimate vector if it checks the inner product is about $n/4$ with threshold θ_{sl} and θ_{sh}, where θ_{sl} is lower limit, and θ_{sh} is upper limit. A shifted feature vector shows $\boldsymbol{B_s} = (b_{s+0 \bmod n}, b_{s+1 \bmod n}, \ldots)$, where s is shift value. In this case, an attacker needs to set more elements as non-zero values so as to match the inner product of a feature vector and shifted feature vector itself.

For instance, when an attacker attempts with $\tilde{\boldsymbol{B}} = (\beta, 0, 0, \ldots, 0)$, the inner product of attack vector $\tilde{\boldsymbol{B}}$ and shifted attack vector $\tilde{\boldsymbol{B}}_{\boldsymbol{s}} = (0, 0, \beta, \ldots, 0)$ is 0.

B. Algorithm

In this section, we only explain the method of authentication phase in our proposed scheme since setup and enrollment phases are same as the conventional scheme.

The computation server $\mathcal{S}$ calculates the Hamming distance $d_H(\boldsymbol{A}, \boldsymbol{B})$ as well as the conventional scheme. $\mathcal{S}$ first verifies the attempted $\boldsymbol{B}$ by calculating $d_H(\boldsymbol{A}, \boldsymbol{B})$. If $d_H(\boldsymbol{A}, \boldsymbol{B}) > \theta_{hd}$, the authentication fails. Otherwise, $\mathcal{S}$ verifies $\boldsymbol{B}$ by partial comparison and shift-then-inner product scheme to detect whether $\boldsymbol{B}$ is an attack vector or not. In partial comparison scheme, it prepares a masking vector $\boldsymbol{C'}$ whose n/d elements are 1 and the others are 0, and a shift parameter $-x^s$ so as to get partial elements of $\boldsymbol{A}$ and $\boldsymbol{B}$, and inner product of $\boldsymbol{B}$ and $\boldsymbol{B_s}$ respectively. $\mathcal{S}$ then calculates $p_{sum}(\boldsymbol{A}, \boldsymbol{B})$, which is the difference between the sum of partial elements of feature vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ by $\mathsf{ct}_1^{pack}(\boldsymbol{A}) \dot{\times} \mathsf{ct}_2^{pack}(\boldsymbol{C'}) \dot{+} (-\mathsf{ct}_1^{pack}(\boldsymbol{C'}) \dot{\times} \mathsf{ct}_2^{pack}(\boldsymbol{B}))$. By using a masking vector $\boldsymbol{C'}$, $\mathcal{S}$ can extract the sum of only randomly chosen elements which are part of $c'_i = 1$, where c'_i are elements of $\boldsymbol{C'}$. In shift-then-inner product scheme, it also calculates $\boldsymbol{B} \cdot \boldsymbol{B_s}$, which is the inner product of $\boldsymbol{B}$ and $\boldsymbol{B_s}$ by $\mathsf{ct}_2^{pack}(\boldsymbol{B}) \dot{\times} (\mathsf{ct}_2^{pack}(\boldsymbol{B}) \dot{\times} \mathsf{ct}_2^{pack}(-x^s))$, and then sends $d_H(\boldsymbol{A}, \boldsymbol{B})$, the difference $p_{sum}(\boldsymbol{A}, \boldsymbol{B})$ and the inner product $\boldsymbol{B} \cdot \boldsymbol{B_s}$ to the decryption server $\mathcal{D}$. By using a shift parameter $-x^s$, $\mathsf{pm}_2(\boldsymbol{B}) \times (-x^s) = \sum_{i=0}^{n-1} b_i x^{n-i+s \bmod n}$ which is decrypted $\mathsf{ct}_2^{pack}(\boldsymbol{B}) \dot{\times} \mathsf{ct}_2^{pack}(-x^s)$, so that $\mathcal{S}$ can extract a shifted feature vector. $\mathcal{D}$ judges that the authentication is successful if $p_{sum}(\boldsymbol{A}, \boldsymbol{B})$ is lower than θ_{psum} and $\theta_{sl} \leq \boldsymbol{B} \cdot \boldsymbol{B_s} \leq \theta_{sh}$.

C. Security Analysis

We discuss the security aspect of our scheme. There are three aspects, which are confidentiality, privacy, and security. We then discuss the confidentiality which is defined as the property that information is not disclosed to the unauthorized entity, i.e., the computation server $\mathcal{S}$. The computer server $\mathcal{S}$ cannot learn any information about feature vectors as long as the secret key sk is leaked out since all feature vectors are encrypted. Therefore we can use the cloud as $\mathcal{S}$ for outsourcing storage and computation resources.

We then discuss the privacy aspect. Privacy is the property to protect the bio-information. Information about the original data cannot be deduced even if the feature vectors are revealed since the original data are transformed into feature vectors by irreversible conversion. Moreover, the feature vectors are encrypted. Therefore, the bio-information, i.e., identity theft is very hard.

Finally, security is discussed. Security is the degree of protection from attack. We discuss an attacker that can steal $\mathsf{ct}_1^{pack}(\boldsymbol{A})$. When an attacker sends it to $\mathcal{S}$ instead of $\mathsf{ct}_2^{pack}(\boldsymbol{B})$ in the authentication phase, the authentication fails with very high probability due to packing methods. With respect to the attack success rate, we discuss it in Section V.

V. Evaluation

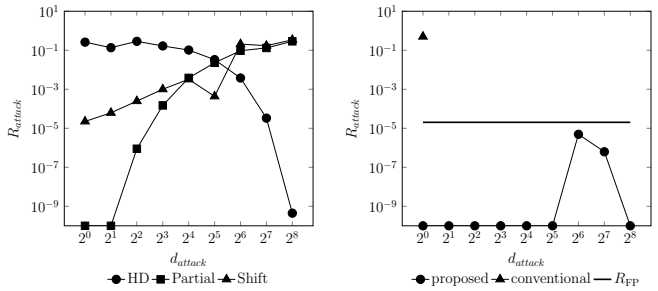
In order to show the efficiency of our scheme, we evaluate the attack success rate R_{attack}, the true positive rate R_{TP} and the false positive rate R_{FP} with an iris dataset. R_{attack} denotes the ratio of the total number of attacker's successful authentication to the total number of impersonation attempts. R_{TP} denotes the ratio of the total number of legitimate user's successful authentication to the total number of login attempts by him/herself. R_{FP} denotes the ratio of the total number of mistakenly authenticated to the total number of login attempts by other person. The dataset is IITD Iris Database version 1.0 which consists of as many as $2,240$ iris images collected from 224 students and staff at IIT Delhi, New Delhi, India [12]. For each person, totally 10 images (5 images from each eye) are captured. These images are first translated to feature vectors and then are masked with data in [13]. We use mask data [14] to exclude non-iris pixels.

We show the simulation parameters in TABLE I. d_{attack} denotes the number of partition in attacker, i.e., an attacker partitions β into d_{attack} elements. The attacker can know d value and can set d_{attack} according to d. We heuristically define the threshold of hamming distance θ_{hd} considering high true positive and low false positive. Moreover, θ_{psum}, θ_{sl} and θ_{su} are also heuristically set to achieve the highest R_{TP}. The value of elements β with an attack vector $\tilde{\boldsymbol{B}}$ is $(n/2)/d_{attack}$.

We assume three possible types of attacker. Attacker 1 tries to impersonate a legitimate feature vector by using the same number of elements expect 0. In order to do that, the attacker 1 generates $\tilde{\boldsymbol{B}} = (\tilde{b}_0, \tilde{b}_1, \ldots, \tilde{b}_{n-1})$, whose d_{attack} elements are $\tilde{b}_i = \beta_{d_{attack}}$, where i is d_{attack} numbers from 0 to $n-1$, and the other elements are 0, and then tries authentication. Attacker 2 tries to impersonate a legitimate feature vector by using various number of elements since setting a big number increases attack success rate in the case of Hamming distance while setting a small number increases that of partial comparison and random shift. In order to do that, the attacker 2 generates $\tilde{\boldsymbol{B}} = (\beta/2, \beta/4, \beta/8, \ldots, 1, 0, \ldots, 0)$. Attacker 3 tries to impersonate a legitimate user by using only both extremely big and small numbers to a feature vector. In order to do that, the attacker 3 generates $\tilde{\boldsymbol{B}} = (\beta/2, 1, \ldots, 1, 0, \ldots, 0)$. However, it is an open question whether more sophisticated attack exists.

TABLE I. SIMULATION PARAMETER

Name	Data
Number of elements n	2048
Number of partitions d_{attack}	2^k $(0 \leq k \leq 8)$
Threshold of hamming distance θ_{hd}	697
Threshold of partial sum θ_{psum}	$697/d$
Threshold of lower inner product θ_{sl}	460
Threshold of upper inner product θ_{su}	565
Value of elements β	1024
Value of elements β_d	$1024/d$
Number of elements β_d	d



(a) three schemes individually ($d = d_{attack}/2$). (b) combining three schemes ($d = 128$).

Fig. 1. R_{attack} versus d_{attack} against attacker 1.

A. Attack Success Rate R_{attack} versus d_{attack} against Attacker 1

Fig. 1(a) shows the attack success rate R_{attack} versus d_{attack} in the case of $d = d_{attack}/2$. We evaluate R_{attack} with $d = d_{attack}/2$ since R_{attack} is decreased when $d > d_{attack}$. In this figure, "HD", "partial" and "Shift" denote that the three types of schemes with only Hamming distance, i.e., the conventional scheme, only partial comparison and only shift-then-inner product, respectively. From Fig. 1(a), we can see that as d_{attack} for HD gets larger, R_{attack} gets lower. This is because the attacker sets smaller numbers β_d to an element of $\tilde{\boldsymbol{B}}$ when d_{attack} gets larger. We then discuss the result of Partial. From this figure, we can see that as d_{attack} gets smaller, R_{attack} gets lower. When $d_{attack} \geq 128$, the sum of partial enrolled vector and that of partial attack vector is almost the same due to the sum of less elements. For example, if $d = n$, an attacker can succeed in the authentication by matching only one element. We then discuss the result of Shift. From this figure, we can see that as d_{attack} gets smaller, R_{attack} gets lower. In the case of larger d_{attack}, the inner product gets closer to $n/4$ since the elements of an attack vector is small. As a result, an attacker has to select neither too large nor too small d_{attack} so as to succeed in the attack with higher probability, in this case, the best strategy for an attacker is to set $d_{attack} = 64$.

We evaluate R_{attack} for the proposal which combines HD, Partial and Shift and compare it with the conventional scheme. In this evaluation, we set $d = d_{attack} * 2 = 128$ since the attack success rate R_{attack} can be decreased by using $d > d_{attack} = 64$. When d gets excessively larger, the attacker can easily succeed in the authentication so as to decrease elements with partial comparison. Fig. 1(b) shows the attack success rate R_{attack} versus d_{attack} in the case of $d = 128$. There is only one plot ($d_{attack} = 2^0$) of conventional scheme since the conventional scheme does not use partial feature vectors. We compare R_{attack} with the false positive rate R_{FP} since if R_{attack} is lower than R_{TP}, it is better for an attacker to use another person's feature vector $\boldsymbol{B}$ for impersonation. Therefore, if R_{attack} is lower than R_{TP}, we show that our proposed scheme is secure system. From Fig. 1(b), R_{attack} of the proposed scheme is always lower than R_{FP}. As a result, our proposed scheme can make the attack success rate R_{attack} lower than the false positive rate R_{FP}.

TABLE II. R_{attack} AGAINST ATTACKER 2 AND ATTACKER 3.

Scheme	Value
R_{attack} of attacker 2	$10^{-6.05}$
R_{attack} of attacker 3	$10^{-7.35}$
R_{FP}	$10^{-4.69}$

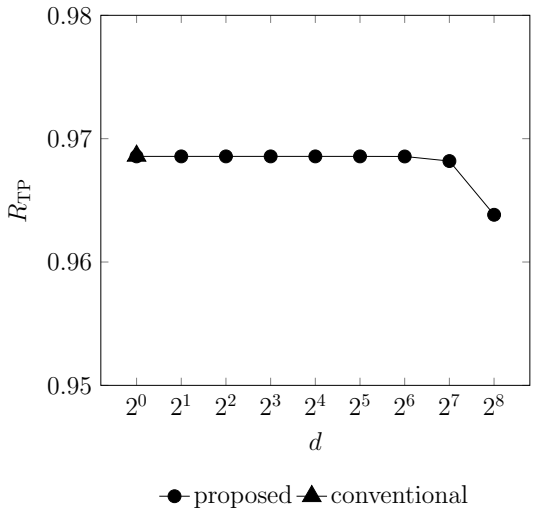


Fig. 2. R_{TP} versus d.

B. Attack Success Rate R_{attack} against Attacker 2 and Attacker 3

TABLE II shows the attack success rate R_{attack} against attacker 2 and attacker 3. We compare R_{attack} with the false positive rate R_{FP} so as to show that an attack scheme with $\tilde{\boldsymbol{B}}$ is less effective than the other person's feature vector $\boldsymbol{B}$. From TABLE II, R_{attack} is lower than R_{FP}. As a result, our proposed scheme can make the attack success rate R_{attack} lower than the false positive rate R_{FP}. Therefore, our proposed scheme is effective since an attack scheme with the other person's vector $\boldsymbol{B}$ is higher attack success rate compared with the attack scheme with an attack vector $\tilde{\boldsymbol{B}}$ we assume.

C. True Positive Rate versus d

Fig. 2 shows the true positive rate R_{TP} versus d. There is only one plot ($d = 2^0$) of the conventional scheme since the

conventional scheme does not use partial feature vector. From Fig. 2, we can see that R_{TP} is a constant value when $d \leq 2^6$. In contrast, when $d > 2^6$, R_{TP} gets worse. If d is lower, R_{TP} does not decrease since our scheme can compare the sum of more elements. On the other hand, if d is larger, $p_{sum}(\boldsymbol{A}, \boldsymbol{B})$ is larger than θ_{psum} even if $\boldsymbol{A}$ and $\boldsymbol{B}$ are same person's feature vectors since our scheme compares the sum of less elements. Moreover, we set $d > d_{attack}$ so as to decrease R_{attack}. As a result, we can say $d = 128$ is the appropriate choice.

VI. Conclusion

We have proposed a secure practical packing-based privacy-preserving biometric authentication scheme by adding two schemes to the conventional scheme. We solve the security issue of the conventional scheme by introducing the second verification phase called shift-then-inner product.. We have evaluated the attack success rate, the true positive rate and the false positive rate with an iris dataset and have shown that our proposed scheme achieves the attacker's success rate $10^{-5.31}$, $10^{-6.05}$ and $10^{-7.35}$ against different attackers and both satisfy the required rate $10^{-4.69}$.

Appendix
Computing Hamming Distance in a Ciphered State

we show how to compute $d_H(\boldsymbol{A}, \boldsymbol{B})$ without decrypting $\boldsymbol{A}$ and $\boldsymbol{B}$. The computation server $\mathcal{S}$ can calculate the inner product of $\boldsymbol{A}$ and $\boldsymbol{B}$.

$$\begin{aligned}
\mathsf{ct}_0^{pack} &= \mathsf{ct}_1^{pack}(\boldsymbol{A}) \dot{\times} \mathsf{ct}_2^{pack}(\boldsymbol{B}) \\
&= \mathsf{Enc}_{\mathsf{pk}}(\mathsf{pm}_1(\boldsymbol{A}) \times \mathsf{pm}_2(\boldsymbol{B})) \\
&= \mathsf{Enc}_{\mathsf{pk}}(-(a_0 b_0 + \cdots + a_{n-1} b_{n-1}) x^n + \ldots) \\
&= \mathsf{Enc}_{\mathsf{pk}}(-(\boldsymbol{A} \cdot \boldsymbol{B}) x^n + \ldots). \qquad (7)
\end{aligned}$$

The decryption server $\mathcal{D}$ can obtain the inner product $\boldsymbol{A} \cdot \boldsymbol{B}$ by decrypting the encrypted multiplication ct_0^{pack}.

$$\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}_0^{pack}) = \boldsymbol{A} \cdot \boldsymbol{B} + \ldots, \qquad (8)$$

where $x^n = -1$ in R_t.

In the above authentication phase, from the inner product calculation method, the computation server $\mathcal{S}$ calculates $d_H(\boldsymbol{A}, \boldsymbol{B})$ by using a feature vector $\boldsymbol{C} = (1, 1, \ldots, 1)$, whose elements are all 1.

$$\begin{aligned}
\mathsf{ct}^{pack} &= \mathsf{ct}_1^{pack}(\boldsymbol{A}) \dot{\times} \mathsf{ct}_2^{pack}(\boldsymbol{C}) \dot{+} \mathsf{ct}_1^{pack}(\boldsymbol{C}) \dot{\times} \mathsf{ct}_2^{pack}(\boldsymbol{B}) \\
&\quad \dot{+} (-2\mathsf{ct}_1^{pack}(\boldsymbol{A}) \dot{\times} \mathsf{ct}_2^{pack}(\boldsymbol{B})) \\
&= \mathsf{Enc}_{\mathsf{pk}}(-(\boldsymbol{A} \cdot \boldsymbol{C} + \boldsymbol{C} \cdot \boldsymbol{B} - 2(\boldsymbol{A} \cdot \boldsymbol{B})) x^n + \ldots) \\
&= \mathsf{Enc}_{\mathsf{pk}}\left(-\left(\sum_{i=0}^{n-1} a_i + \sum_{i=0}^{n-1} b_i - 2\sum_{i=0}^{n-1} a_i b_i\right) x^n + \ldots\right) \\
&= \mathsf{Enc}_{\mathsf{pk}}(-(d_H(\boldsymbol{A}, \boldsymbol{B})) x^n + \ldots). \qquad (9)
\end{aligned}$$

The decryption server $\mathcal{D}$ can obtain the Hamming distance between $\boldsymbol{A}$ and $\boldsymbol{B}$ by decrypting the encrypted multiplication ct^{pack}

$$\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}^{pack}) = d_H(\boldsymbol{A}, \boldsymbol{B}) + \ldots, \qquad (10)$$

where $x^n = -1$ in R_t.

Acknowledgment

This work is partly supported by the Grant in Aid for Scientific Research (No.26420369) from Ministry of Education, Sport, Science and Technology, Japan.

References

[1] J. Bringer and H. Chabanne, "An authentication protocol with encrypted biometric data," in *Progress in Cryptology AFRICACRYPT 2008*. Springer Berlin Heidelberg, 2008, pp. 109–124.

[2] S. Rane, W. Sun, and A. Vetro, "Secure distortion computation among untrusting parties using homomorphic encryption," in *16th IEEE International Conference on Image Processing*, Nov 2009, pp. 1485–1488.

[3] M. Hattori, N. Matsuda, T. Ito, Y. Shibata, K. Takashima, and T. Yoneda, "Provably-secure cancelable biometrics using 2-DNF evaluation," *Journal of Information Processing*, vol. 20, no. 2, pp. 496–507, 2012.

[4] J. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148–1161, Nov 1993.

[5] A. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 846–859, May 2000.

[6] A. Kong and D. Zhang, "Competitive coding scheme for palmprint verification," in *Proceedings of the 17th International Conference on Pattarn Recognition*, Aug 2004, pp. 520–523.

[7] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, "Practical packing method in somewhat homomorphic encryption," in *Data Privacy Management and Autonomous Spontaneous Security*. Springer Berlin Heidelberg, 2014, pp. 34–50.

[8] A. Mandal, A. Roy, and M. Yasuda, "Comprehensive and improved secure biometric system using homomorphic encryption," in *Data Privacy Management, and Security Assurance*. Springer International Publishing, 2016, pp. 183–198.

[9] T. Izu, Y. Sakemi, M. Takenaka, and N. Torii, "A spoofing sttack against a cancelable biometric authentication scheme," in *IEEE 28th International Conference on Advanced Information Networking and Applications*, May 2014, pp. 234–239.

[10] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop*, 2011, pp. 113–124.

[11] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *Advances in Cryptology CRYPTO 2011*. Springer Berlin Heidelberg, 2011, pp. 505–524.

[12] IITD Iris Database version 1.0. [Online]. Available: http://www4.comp.polyu.edu.hk/wcsajaykr/IITD/Database_Iris.htm

[13] USIT - University of Salzburg Iris Toolkit v1.0. [Online]. Available: http://www.wavelab.at/sources/

[14] Iris Segmentation Ground Truth Database – Elliptical/Polynomial boundaries (IRISSEG-EP). [Online]. Available: http://www.wavelab.at/sources/