Security Analysis on Privacy-Preserving Cloud Aided Biometric Identification Schemes

Shiran Pan[1,2,3], Shen Yan[1,2,3], and Wen-Tao Zhu[1,2(✉)]

[1] State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{panshiran,yanshen}@iie.ac.cn, wtzhu@ieee.org
[2] Data Assurance and Communication Security Research Center,
Chinese Academy of Sciences, Beijing, China
[3] University of Chinese Academy of Sciences, Beijing, China

Abstract. Biometric identification (BI) is the task of searching a pre-established biometric database to find a matching record for an enquiring biometric trait sampled from an unknown individual of interest. This has recently been aided with cloud computing, which brings a lot of convenience but simultaneously arouses new privacy concerns. Two cloud aided BI schemes pursuing privacy preserving have recently been proposed by Wang et al. in ESORICS '15. In this paper, we propose several elaborately designed attacks to reveal the security breaches in these two schemes. Theoretical analysis is given to validate our proposed attacks, which indicates that via such attacks the cloud server can accurately infer the outsourced database and the identification request.

Keywords: Biometric identification · Cloud computing · Security breaches · Privacy preserving

1 Introduction

Biometric identification (BI) is to identify an individual of interest by searching a pre-established biometric database to find a matching record for an enquiring user's biometric trait sampled from an unknown individual. Due to the universality, uniqueness, and permanence of the biometric data [1], BI has been wildly used in identifying an individual's identity (e.g., in forensic scenarios). There have been several kinds of BI systems in practical applications, such as fingerprint, voice pattern, and facial pattern recognition systems [2].

As cloud computing is now gaining much momentum, individuals, companies, and governments are motivated to outsource their data to the cloud to enjoy the benefits of high flexibility and cost-saving feature of the cloud computing [3]. As far as the BI system is concerned, the database owner may desire to outsource the biometric database to the cloud and enjoy the cloud aided identification service, which can relieve the database owner of the local storage burden and the high computation overhead introduced by searching over the large-scale database. However, the proliferation of cloud aided biometric identification (CABI)

J.K. Liu and R. Steinfeld (Eds.): ACISP 2016, Part II, LNCS 9723, pp. 446–453, 2016.
DOI: 10.1007/978-3-319-40367-0_29

also attracts increasing concerns on its security [4] and privacy [5], since the biometric data is highly sensitive and is impossible to be revoked and replaced once leaked. Therefore, appropriate protection mechanism should be carefully placed in CABI systems in order to combat unsolicited access and inadvertent information disclosure.

Several CABI schemes [6,7] have recently been proposed but these schemes are not appropriate for real-world cloud aided applications, since they will be cracked down when there exists collusion between the system participants. Focused on the collusion resistance, some other schemes have been proposed by Yuan et al. [8] and Wang et al. [9]. Yuan et al. [8] claimed that their scheme is secure under the known-plaintext attack (KPA) and even the chosen-plaintext attack (CPA). However, Wang et al. [9] observed that it is not the case and presented some attacks to show that the scheme proposed in [8] can be broken by KPA and CPA. As a following study, in ESORICS '15 Wang et al. [9] proposed two new CABI schemes considering the semi-honest participants. Wang et al. claimed their schemes achieve higher security since the proposed basic scheme is resilient to the known-sample attack (KSA), while the enhanced scheme can additionally defend against the collusion attack of the cloud server and some enquiring user. However, we observe that both schemes are vulnerable, even to exactly the adversaries designated in [9]. Specifically, we present several elaborately designed attacks that will completely break these schemes [9].

Our technical contributions can be summarized as follows:

- We propose several elaborately designed attacks to reveal the inherent security breaches in the two schemes proposed in [9].
- Theoretical analysis is given to validate our proposed attacks, which indicates that via such attacks the cloud server can accurately infer the outsourced database and the identification request.

The rest of the paper is organized as follows: In Sect. 2, we review Wang et al.'s schemes [9]. We propose several attacks on these schemes [9] in Sect. 3. The paper is concluded in Sect. 4.

2 Review of Wang et al.'s Two Schemes

Recently, Wang et al. [9] proposed two CABI schemes that focus on the fingerprint identification. Following [5,8], Wang et al. assumed that both the biometric records in the database and the biometric trait submitted by the enquiring user are represented by feature vectors. In this section, we will review these two schemes by describing their main bodies.

2.1 CloudBI-I: The Basic Scheme

We first describe Wang et al.'s basic scheme CloudBI-I. For the fingerprints collected from m individuals, the biometric database owner first generates the corresponding biometric records denoted as $\{\boldsymbol{b}_i\}_{i=1}^{m}$, which form the biometric

database $\mathcal{D}$. Each $\boldsymbol{b}_i$ is set to an n-dimensional vector, i.e., $\boldsymbol{b}_i = (b_{i1}, b_{i2}, \cdots, b_{in})$, with each entry b_{ij} lying in a pre-determined domain. To facilitate the identification, $\boldsymbol{b}_i$ will be extended to $\hat{\boldsymbol{b}}_i = (b_{i1}, b_{i2}, \cdots, b_{in}, b_{i,n+1}, 1)$, where $b_{i,n+1} = -(b_{i1}^2 + b_{i2}^2 + \cdots + b_{in}^2)/2$. The database owner then accordingly generates a diagonal matrix $\boldsymbol{B}_i$ with the diagonal entries set to $\{b_{i1}, b_{i2}, \cdots, b_{in}, b_{i,n+1}, 1\}$ (i.e., the entries of $\hat{\boldsymbol{b}}_i$).

Subsequently, the database owner randomly selects two $(n+2) \times (n+2)$ invertible matrices $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ as the encryption keys, and encrypts each $\boldsymbol{B}_i$ by computing

$$\boldsymbol{C}_i = \boldsymbol{M}_1 \boldsymbol{B}_i \boldsymbol{M}_2. \tag{1}$$

After encryption, the database owner outsources the encrypted database $\mathcal{C} = \{\boldsymbol{C}_i\}_{i=1}^m$ to the cloud server. When an enquiring user has a fingerprint to be identified, he first locally generates the corresponding biometric trait $\boldsymbol{b}_t = (b_{t1}, b_{t2}, \cdots, b_{tn})$ that is also an n-dimensional vector, and then submits it to the database owner who will select a random number r_t and extend $\boldsymbol{b}_t$ to $\hat{\boldsymbol{b}}_t = (b_{t1}, b_{t2}, \cdots, b_{tn}, 1, r_t)$. The database owner then generates a diagonal matrix $\boldsymbol{B}_t$ with the diagonal entries set to $\{b_{t1}, b_{t2}, \cdots, b_{tn}, 1, r_t\}$ (i.e., the entries of $\hat{\boldsymbol{b}}_t$), and subsequently encrypts $\boldsymbol{B}_t$ by computing

$$\boldsymbol{C}_T = \boldsymbol{M}_2^{-1} \boldsymbol{B}_t \boldsymbol{M}_1^{-1}.$$

Then $\boldsymbol{C}_T$ is submitted to the cloud server for identification. Upon receiving $\boldsymbol{C}_T$, the cloud server compares the Euclidean distance between each $\boldsymbol{b}_i$ and $\boldsymbol{b}_t$ by computing the trace (denoted as $\boldsymbol{tr}(\cdot)$) of the following matrix $\boldsymbol{P}_i$:

$$\boldsymbol{P}_i = \boldsymbol{C}_i \boldsymbol{C}_T = \boldsymbol{M}_1 \boldsymbol{B}_i \boldsymbol{M}_2 \boldsymbol{M}_2^{-1} \boldsymbol{B}_t \boldsymbol{M}_1^{-1} = \boldsymbol{M}_1 \boldsymbol{B}_i \boldsymbol{B}_t \boldsymbol{M}_1^{-1}.$$

Due to the property of matrix similarity transformation [10], $\boldsymbol{tr}(\boldsymbol{P}_i)$ is thus equal to $\boldsymbol{tr}(\boldsymbol{B}_i \boldsymbol{B}_t)$, i.e., $\boldsymbol{tr}(\boldsymbol{P}_i)$ equals $(\sum_{j=1}^n b_{ij} b_{tj} + b_{i,n+1} + r_t)$. The cloud server then sorts these values $\{\boldsymbol{tr}(\boldsymbol{P}_i)\}_{i=1}^m$ and accordingly returns the candidate results to the database owner. Here we omit other details of this scheme, since they are irrelevant to our proposed attacks.

2.2 CloudBI-II: The Enhanced Scheme

Wang et al. claimed that CloudBI-I can resist KSA but will be broken by the collusion between the cloud server and some enquiring user. Therefore, Wang et al. proposed an enhanced scheme CloudBI-II. The main idea is to introduce more randomness into the database encryption and the query encryption. Specifically, for each $\boldsymbol{B}_i$, the database owner additionally selects a random lower triangular matrix $\boldsymbol{Q}_i$ with the diagonal entries set to all 1's, and then encrypts $\boldsymbol{B}_i$ by computing

$$\boldsymbol{C}_i = \boldsymbol{M}_1 \boldsymbol{Q}_i \boldsymbol{B}_i \boldsymbol{M}_2.$$

Similarly, the database owner generates the encrypted identification query for the enquiring user as

$$\boldsymbol{C}_T = \boldsymbol{M}_2^{-1} \boldsymbol{B}_t \boldsymbol{Q}_t \boldsymbol{M}_1^{-1}, \tag{2}$$

where $\boldsymbol{Q}_t$ is also a random lower triangular matrix with the diagonal entries set to all 1's. The remaining operations are the same as the basic scheme CloudBI-I.

3 Proposed Attacks

In this section, we will propose several elaborately constructed attacks by exploiting the inherent structure of the biometric data and some important properties of matrix transformation.

3.1 Modified Signature Linking Attack on CloudBI-I

Wang et al. [9] claimed that their basic scheme CloudBI-I is resilient to KSA because the techniques they used for designing the scheme are not belong to distance-preserving transformation (DPT) [11], i.e., the Euclidean distances between any two plaintext biometric records will not be preserved after encryption. Therefore, according to the analysis presented in [12], the PCA attack [13] and the signature linking attack [12] will fail to attack CloudBI-I. However, we observe that the above reasoning is not rigorous and we will demonstrate a so-called modified signature linking attack (MSLA), which bypasses the computation on the Euclidean distances, to recover the outsourced database in CloudBI-I.

According to the definition of KSA, the adversary has some samples in the plaintext database $\mathcal{D}$. Without loss of generality, we assume that the knowledge of the adversary is $\mathcal{G} = \{\boldsymbol{b}_i\}_{i=1}^{k} \subseteq \mathcal{D}$ so that he can naturally generate $\{\boldsymbol{B}_i\}_{i=1}^{k}$ without knowing any of the corresponding encrypted values $\{\boldsymbol{C}_i\}_{i=1}^{k}$. As shown in Eq. 1, due to the property of matrix similarity transformation [10], we have

$$\boldsymbol{tr}\big(\boldsymbol{C}_i^{-1}\boldsymbol{C}_j\big) = \boldsymbol{tr}\big(\boldsymbol{M}_2^{-1}\boldsymbol{B}_i^{-1}\boldsymbol{B}_j\boldsymbol{M}_2\big) = \boldsymbol{tr}\big(\boldsymbol{B}_i^{-1}\boldsymbol{B}_j\big). \tag{3}$$

Although the Euclidean distances between the encrypted records are not preserved, we can define the signature of $\mathcal{G}$ as

$$sig(\mathcal{G}) = \{\boldsymbol{tr}\big(\boldsymbol{B}_1^{-1}\boldsymbol{B}_2\big), \cdots, \boldsymbol{tr}\big(\boldsymbol{B}_1^{-1}\boldsymbol{B}_k\big), \boldsymbol{tr}\big(\boldsymbol{B}_2^{-1}\boldsymbol{B}_3\big), \cdots, \boldsymbol{tr}\big(\boldsymbol{B}_{k-1}^{-1}\boldsymbol{B}_k\big)\}.$$

In MSLA, the adversary aims to find an ordered set of encrypted records $\mathcal{H} \subseteq \mathcal{C} = \{\boldsymbol{C}_i\}_{i=1}^{m}$, such that $\mathcal{H}$ has the same size and gives the same signature as $\mathcal{G}$. Let $\mathcal{H} = \{\boldsymbol{C}_{1'}, \boldsymbol{C}_{2'}, \cdots, \boldsymbol{C}_{k'}\}$ so that the signature of $\mathcal{H}$ is

$$sig(\mathcal{H}) = \{\boldsymbol{tr}\big(\boldsymbol{C}_{1'}^{-1}\boldsymbol{C}_{2'}\big), \cdots, \boldsymbol{tr}\big(\boldsymbol{C}_{1'}^{-1}\boldsymbol{C}_{k'}\big), \boldsymbol{tr}\big(\boldsymbol{C}_{2'}^{-1}\boldsymbol{C}_{3'}\big), \cdots, \boldsymbol{tr}\big(\boldsymbol{C}_{(k-1)'}^{-1}\boldsymbol{C}_{k'}\big)\}.$$

If there is only one set $\mathcal{H}$ with a matching signature, the adversary can conclude that $\boldsymbol{C}_{i'}$ is the encrypted value of $\boldsymbol{b}_i$. Then the adversary try to solve any plaintext biometric record $\boldsymbol{b}_j$ correspond to $\boldsymbol{C}_j$ by solving the following linear equations:

$$\boldsymbol{tr}(\boldsymbol{B}_i^{-1}\boldsymbol{B}_j) = \boldsymbol{tr}(\boldsymbol{C}_i^{-1}\boldsymbol{C}_j), \quad i = 1, \cdots, k.$$

Particularly, there are $(n+2)$ unknowns in each linear equation so that the adversary will successfully recover $\boldsymbol{B}_j$ if $k \geq (n+2)$ holds.

The main issue that the success of proposed MSLA rests on is whether there exists a signature collision, i.e., whether it is likely to find another set, which is not the encrypted values of $\mathcal{G}$ but happens to give the same signature as $\mathcal{G}$. As shown in the following theorem, the probability of the signature collision is very small and we can well control it by increasing the size k of $\mathcal{G}$.

Theorem 1. *Let α be the probability of an n-dimensional vector contained in $\mathcal{D}$. Assume the knowledge of the adversary is $\mathcal{G} = \{\boldsymbol{b}_i\}_{i=1}^{k} \subseteq \mathcal{D}$, $\forall \epsilon > 0$, if $k \geq n + 1 + \ln \epsilon / \ln \alpha$, then $Pr(\textit{signature collision}) < \epsilon$.*

Due to the space limit, here we omit the proof. Note that MSLA will also work for CloudBI-II [9] since Eq. 3 still holds. In conclusion, via MSLA the adversary can obtain the corresponding relationships between the plaintext and the encrypted biometric records, and further construct linear equations to get the plaintext database. Next, we will show other two attacks on CloudBI-II.

3.2 Two Attacks on CloudBI-II

In the design of CloudBI-II [9], several random lower triangular matrices $\{\boldsymbol{Q}_i\}_{i=1}^{m}$ and $\boldsymbol{Q}_c$ are introduced into the database encryption and the query encryption. Wang et al. claimed that such randomness makes it impossible for the adversary to figure out either the biometric records $\boldsymbol{b}_i$ in $\mathcal{D}$ or the biometric traits $\boldsymbol{b}_t$ submitted by non-colluding enquiring users, even the adversary can collude with some user and independently select the biometric traits submitted to the database owner. Therefore, Wang et al. asserted that CloudBI-II can defend against the collusion attack of the cloud server and some enquiring user.

Next we will demonstrate two attacks, which rely on the collusion ability of the adversary designated in [9], to break the scheme CloudBI-II. Via these two attacks, the adversary can obtain some certain information about the randomness that are added into the database encryption and the query encryption, and further recover the plaintext biometric records and the enquiring biometric traits.

We begin by describing an attack to recover the biometric records in the database. As defined in [9], the cloud server (i.e., the colluding adversary in the BI system) can independently select several vectors as the biometric traits to be identified. Without loss of generality, we assume that the cloud server selects $(n+2)$ vectors $\{\boldsymbol{b}_t^{(i)}\}_{i=1}^{n+2}$ to be submitted to the database owner, where $\boldsymbol{b}_t^{(i)} = (b_{t1}^{(i)}, b_{t2}^{(i)}, \cdots, b_{tn}^{(i)})$. Upon receiving these vectors, the database owner will encrypt them and send the encrypted values $\{\boldsymbol{C}_T^{(i)}\}_{i=1}^{n+2}$ to the cloud server, here $\boldsymbol{C}_T^{(i)} = \boldsymbol{M}_2^{-1}\boldsymbol{B}_t^{(i)}\boldsymbol{Q}_t^{(i)}\boldsymbol{M}_1^{-1}$ as shown in Eq. 2. Notice that, the cloud server knows all these $\{\boldsymbol{b}_t^{(i)}\}_{i=1}^{n+2}$ but does not know the randomness $\{r_t^{(i)}\}_{i=1}^{n+2}$ that are added into $\{\boldsymbol{C}_T^{(i)}\}_{i=1}^{n+2}$ by the database owner. However, the cloud server can obtain the proportional relationships between $\{r_t^{(i)}\}_{i=1}^{n+2}$ by computing

$$\begin{aligned}
&\boldsymbol{tr}\big((\boldsymbol{C}_T^{(i)})^{-1}\boldsymbol{C}_T^{(j)}\big) - (1/\boldsymbol{b}_t^{(i)})(\boldsymbol{b}_t^{(j)})^{\mathrm{T}} - 1 \\
&= \boldsymbol{tr}(\boldsymbol{M}_1(\boldsymbol{Q}_t^{(i)})^{-1}(\boldsymbol{B}_t^{(i)})^{-1}\boldsymbol{M}_2\boldsymbol{M}_2^{-1}\boldsymbol{B}_t^{(j)}\boldsymbol{Q}_t^{(j)}\boldsymbol{M}_1^{-1}) - (1/\boldsymbol{b}_t^{(i)})(\boldsymbol{b}_t^{(j)})^{\mathrm{T}} - 1 \\
&= \boldsymbol{tr}(\boldsymbol{M}_1(\boldsymbol{Q}_t^{(i)})^{-1}(\boldsymbol{B}_t^{(i)})^{-1}\boldsymbol{B}_t^{(j)}\boldsymbol{Q}_t^{(j)}\boldsymbol{M}_1^{-1}) - (1/\boldsymbol{b}_t^{(i)})(\boldsymbol{b}_t^{(j)})^{\mathrm{T}} - 1. \qquad (4)
\end{aligned}$$

Due to the property of matrix similarity transformation and the fact that the inverse matrix of a unit lower triangular matrix is also a unit lower triangular matrix [10], $\boldsymbol{tr}\big(\boldsymbol{M}_1(\boldsymbol{Q}_t^{(i)})^{-1}(\boldsymbol{B}_t^{(i)})^{-1}\boldsymbol{B}_t^{(j)}\boldsymbol{Q}_t^{(j)}\boldsymbol{M}_1^{-1}\big)$ is therefore equal to $\boldsymbol{tr}\big((\boldsymbol{B}_t^{(i)})^{-1}\boldsymbol{B}_t^{(j)}\big)$. Since the matrix $(\boldsymbol{B}_t^{(i)})^{-1}\boldsymbol{B}_t^{(j)}$ has the following structure:

$$(\boldsymbol{B}_t^{(i)})^{-1}\boldsymbol{B}_t^{(j)} = \begin{pmatrix}
b_{t1}^{(j)}/b_{t1}^{(i)} & 0 & \cdots & \cdots & 0 & 0 \\
0 & b_{t2}^{(j)}/b_{t2}^{(i)} & \cdots & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & \cdots & 0 & b_{tn}^{(j)}/b_{tn}^{(i)} & 0 & 0 \\
0 & \cdots & 0 & 0 & 1 & 0 \\
0 & \cdots & 0 & 0 & 0 & r_t^{(j)}/r_t^{(i)}
\end{pmatrix}$$

the result of Eq. 4 is thus equal to $r_t^{(j)}/r_t^{(i)}$ denoted as r_{ji}. By such computations, the cloud server can get a set of ratios $\{r_{j1}\}_{j=2}^{n+2}$, and further generate a novel matrix $\boldsymbol{D}$ for attacking as

$$\boldsymbol{D} = \begin{pmatrix}
b_{t1}^{(1)} & b_{t2}^{(1)} & \cdots & b_{tn}^{(1)} & 1 & 1 \\
b_{t1}^{(2)} & b_{t2}^{(2)} & \cdots & b_{tn}^{(2)} & 1 & r_{21} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
b_{t1}^{(n+2)} & b_{t2}^{(n+2)} & \cdots & b_{tn}^{(n+2)} & 1 & r_{n+2,1}
\end{pmatrix}.$$

With this matrix, the cloud server can figure out the biometric record $\boldsymbol{b}_i$ corresponding to $\boldsymbol{C}_i$ by solving the following linear equation:

$$\boldsymbol{Dy} = \big(\boldsymbol{tr}(\boldsymbol{C}_i\boldsymbol{C}_T^{(1)}), \boldsymbol{tr}(\boldsymbol{C}_i\boldsymbol{C}_T^{(2)}), \cdots, \boldsymbol{tr}(\boldsymbol{C}_i\boldsymbol{C}_T^{(n+2)})\big)^{\mathrm{T}}. \qquad (5)$$

The biometric record $\boldsymbol{b}_i$ corresponding to $\boldsymbol{C}_i$ actually consists of the first n entries of the solution $\boldsymbol{y}$ to Eq. 5. In this way, the cloud server can recover all the biometric records in the database. For the correctness, we present the following theorem.

Theorem 2. *If the matrix rank of $\boldsymbol{D}$ (denoted as $\boldsymbol{rk}(\boldsymbol{D})$) equals $(n+2)$, then $\boldsymbol{b}_i$ consists of the first n entries of the solution $\boldsymbol{y}$ to Eq. 5.*

Proof. Eq. 5 can be rewritten as $\boldsymbol{Dy} = \boldsymbol{\gamma}$, where the augmented matrix can be denoted as $\tilde{\boldsymbol{D}} = (\boldsymbol{D}, \boldsymbol{\gamma})$. Since $\boldsymbol{rk}(\tilde{\boldsymbol{D}}) = \boldsymbol{rk}(\boldsymbol{D}) = n+2$, we can conclude that there exists a unique solution to Eq. 5. We assume the corresponding biometric record of $\boldsymbol{C}_i$ is $\boldsymbol{b}_i = (b_{i1}, b_{i2}, \cdots, b_{in})$, which will be extended to

$\hat{\boldsymbol{b}}_i = (b_{i1}, b_{i2}, \cdots, b_{in}, b_{i,n+1}, 1)$ for the encryption, where $b_{i,n+1} = -(b_{i1}^2 + b_{i2}^2 + \cdots + b_{in}^2)/2$. We now consider the vector $\boldsymbol{y}^* = (b_{i1}, b_{i2}, \cdots, b_{in}, b_{i,n+1}, r_t^{(1)})^{\mathrm{T}}$, here $r_t^{(1)}$ is the random number added into the generation of $\boldsymbol{C}_T^{(1)}$ by the database owner. As introduced in Sect. 2.1, we have

$$\boldsymbol{D}\boldsymbol{y}^* = \begin{pmatrix} b_{t1}^{(1)} & b_{t2}^{(1)} & \cdots & b_{tn}^{(1)} & 1 & 1 \\ b_{t1}^{(2)} & b_{t2}^{(2)} & \cdots & b_{tn}^{(2)} & 1 & r_{21} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{t1}^{(n+2)} & b_{t2}^{(n+2)} & \cdots & b_{tn}^{(n+2)} & 1 & r_{n+2,1} \end{pmatrix} \begin{pmatrix} b_{i1} \\ b_{i2} \\ \vdots \\ r_t^{(1)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{tr}(\boldsymbol{C}_i \boldsymbol{C}_T^{(1)}) \\ \boldsymbol{tr}(\boldsymbol{C}_i \boldsymbol{C}_T^{(2)}) \\ \vdots \\ \boldsymbol{tr}(\boldsymbol{C}_i \boldsymbol{C}_T^{(n+2)}) \end{pmatrix} = \gamma.$$

As shown above, we can conclude that $\boldsymbol{y}^*$ is actually the unique solution to Eq. 5. Therefore, $\boldsymbol{b}_i$ consists of the first n entries of the solution to Eq. 5.

Based on the ratios r_{ji} calculated by Eq. 4, we can design another attack on CloudBI-II so that the cloud server can recover all the biometric traits submitted by non-colluding enquiring users. Specifically, the adversary can construct another matrix $\boldsymbol{A}$ for attacking as

$$\boldsymbol{A} = \begin{pmatrix} 1/b_{t1}^{(1)} & 1/b_{t2}^{(1)} & \cdots & 1/b_{tn}^{(1)} & 1 \\ 1/b_{t1}^{(2)} & 1/b_{t2}^{(2)} & \cdots & 1/b_{tn}^{(2)} & 1/r_{21} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1/b_{t1}^{(n+1)} & 1/b_{t2}^{(n+1)} & \cdots & 1/b_{tn}^{(n+1)} & 1/r_{n+1,1} \end{pmatrix}.$$

Upon receiving a new encrypted query $\boldsymbol{C}_T^*$ submitted by the database owner, the cloud server can figure out the corresponding enquiring biometric trait $\boldsymbol{b}_t^*$ by solving the following linear equation:

$$\boldsymbol{A}\boldsymbol{x} = \left(\boldsymbol{tr}((\boldsymbol{C}_T^{(1)})^{-1}\boldsymbol{C}_T^*) - 1, \boldsymbol{tr}((\boldsymbol{C}_T^{(2)})^{-1}\boldsymbol{C}_T^*) - 1, \cdots, \boldsymbol{tr}((\boldsymbol{C}_T^{(n+1)})^{-1}\boldsymbol{C}_T^*) - 1\right)^{\mathrm{T}}. \tag{6}$$

The vector that consists of the first n entries of the solution $\boldsymbol{x}$ to Eq. 6 is exactly the $\boldsymbol{b}_t^*$ corresponding to $\boldsymbol{C}_T^*$. Similarly, we have the following theorem.

Theorem 3. *If* $\boldsymbol{rk}(\boldsymbol{A})$ *equals* $(n+1)$*, then* $\boldsymbol{b}_t^*$ *consists of the first* n *entries of the solution* $\boldsymbol{x}$ *to Eq. 6.*

The proof of Theorem 3 is similar to that of Theorem 2. Here we omit the proof due to the space limit.

4 Conclusion

In this paper, we have proposed several elaborately designed attacks to reveal the inherent security breaches in the two CABI schemes proposed by Wang et al. [9]. Additionally, theoretical analysis has been given to validate our proposed attacks. As our future work, we will address the privacy-preserving CABI problem by constructing new encryption algorithms for the biometric data.

Acknowledgment. The authors would like to thank the anonymous reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China under Grant 61272479, the National 973 Program of China under Grant 2013CB338001, and the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA06010702

References

1. Bolle, R., Pankanti, S.: Biometrics: Personal Identification in Networked Society. Kluwer Academic Publishers, Norwell (1998)
2. Jain, A.K., Hong, L., Pankanti, S.: Biometric identification. Commun. ACM **43**, 90–98 (2000)
3. Marstona, S., Li, Z., Bandyopadhyay, S., Zhang, J., Ghalsasi, A.: Cloud computing - The business perspective. Decis. Support Syst. **51**, 176–189 (2011)
4. Al-Assam, H., Jassim, S.: Security evaluation of biometric keys. Comput. Secur. **31**, 151–163 (2012)
5. Huang, Y., Malka, L., Evans, D., Katz, J.: Efficient privacy-preserving biometric identification. In: 18th Annual Network & Distributed System Security Symposium NDSS 2011, February 2011
6. Blanton, M., Aliasgari, M.: Secure outsourced computation of iris matching. J. Comput. Secur. **20**, 259–305 (2012)
7. Chun, H., Elmehdwi, Y., Li, F., Bhattacharya, P., Jiang, W.: Outsourceable two-party privacy-preserving biometric authentication. In: 9th Symposium on Information, Computer and Communications Security ASIACCS 2014, pp. 401–412. ACM (2014)
8. Yuan, J., Yu, S.: Efficient privacy-preserving biometric identification in cloud computing. In: 32nd IEEE International Conference on Computer Communications INFOCOM 2013, pp. 2652–2660. IEEE (2013)
9. Wang, N., Hu, S., Ren, K., He, M., Du, M., Wang, Z.: CloudBI: practical privacy-preserving outsourcing of biometric identification in the cloud. In: Pernul, G., Ryan, P.Y.A., Weippl, E. (eds.) ESORICS 2015. LNCS. Springer, Heidelberg (2015)
10. Strang, G.: Introduction to Linear Algebra. Wellesley, Cambridge (2009)
11. Oliveira, S.R.M., Zaiane, O.R.: Privacy preserving clustering by data transformation. J. Inf. Data Manag. **1**, 53–56 (2010)
12. Wong, W.K., Cheung, D.W., Kao, B., Mamoulis, N.: Secure kNN computation on encrypted databases. In: 28th ACM International Conference on Management of Data, SIGMOD 2009, pp. 139–152. ACM (2009)
13. Liu, K., Giannella, C.M., Kargupta, H.: An attacker's view of distance preserving maps for privacy preserving data mining. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 297–308. Springer, Heidelberg (2006)