A Further Note on the Security of CloudBI: Practical Privacy-Preserving Outsourcing of Biometric Identification in the Cloud

Shengshan Hu, Minxin Du, Minghui Li and Qian Wang

Dept. of CS, Wuhan University

1 Introduction

Recently, [2] presented a security attack on the privacy-preserving outsourcing scheme for biometric identification proposed in [1]. In [2], the author claims that the scheme CloudBI-II proposed in [1] can be broken under the collusion case. That is, when the cloud server acts as a user to submit a number of identification requests, CloudBI-II is no longer secure. In this technical report, we will explicitly show that the attack method proposed in [2] doesn't work in fact.

2 A Review of [2]'s Attack Method on CloudBI-II

The attack method presented in [2] is based on the equation $\mathbf{P}_i = \mathbf{M}_1\mathbf{Q}_i\mathbf{B}'_i\mathbf{B}'_c\mathbf{Q}_c\mathbf{M}_1^{-1}$, where $\mathbf{P}_i$ is a matrix received by the cloud server, $\mathbf{M}_1$ is an unknown random matrix and $\mathbf{M}_1^{-1}$ is the inverse of $\mathbf{M}_1$, $\mathbf{Q}_i$ and $\mathbf{Q}_c$ are two random lower triangular matrices that are unknown by the server, $\mathbf{B}'_i$ is an unknown diagonal matrix, and $\mathbf{B}'_c$ is a diagonal matrix that most of entries can be known when the collusion happens. Note that all the above matrices are n'-dimensional and for each different $\mathbf{B}'_c$ there comes a different $\mathbf{Q}_c$. [2] claims that this matrix equation can be solved because when there comes a number of $\mathbf{B}'_c$'s, the number of equations will exceed the number of unknowns such that it is able to solve all the unknowns. More specifically, it is argued in [2] that, $\mathbf{M}_1, \mathbf{Q}_i$ and $\mathbf{B}'_i$ are fixed for all identification requests, $\mathbf{B}'_c$ and $\mathbf{Q}_c$ are randomly generated for each identification request. Thus, for each new identification request, the server can construct n'^2 new equations while only introducing $\frac{n'^2-n'}{2}+1$ new unknowns, which makes it possible for the server to work out all the unknowns when there comes only 3 identifications.

However, [2] neglects the facts that i) the number of equations is larger than the number of unknowns is not a sufficient condition to solve a system of equations and determine the unique solution, and ii) the system of equations established in the analysis is a system of non-linear equations instead of linear equations. So, you cannot simply say unknowns are solved. In the following analysis of this report, we show that when there comes a new identification, the number of newly established equations is much less than n'^2 since many of them are *linearly dependent* and do not help in solving for the unknowns.

3 A Detailed Analysis to Show Why [2]'s Attack Method Cannot Work

To make the analysis more clear to the readers on the equation $\mathbf{P}_i = \mathbf{M}_1\mathbf{Q}_i\mathbf{B}'_i\mathbf{B}'_c\mathbf{Q}_c\mathbf{M}_1^{-1}$, we first make the following assumptions:
1. The randomly generated matrix $\mathbf{Q}_c$ are all the same for each new identification request $\mathbf{B}'_c$.
2. Since the last entry of $\mathbf{B}'_c$ is a random number r_c, we multiply it into the last entry of $\mathbf{B}'_i$ and set the last entry of $\mathbf{B}'_c$ to be a public number that can even be arbitrarily chosen by the server. In this way, $\mathbf{B}'_i$ will be an unknown matrix that consists of n' unknowns while the total number of unknowns in the equation of $\mathbf{P}_i$ stays the same and the value of $\mathbf{B}'_i\mathbf{B}'_c$ is unchanged.

It is obvious that the above assumptions will only make it easier for server to work out $\mathbf{B}'_i$ by exploiting $\mathbf{P}_i$. Then we will show that none of unknowns of $\mathbf{B}'_i$ can be solved even if under the above assumptions.

Once the assumptions are given, we can see that for different requests $\mathbf{B}'_c$, the remaining unknown matrices in the equation $\mathbf{P}_i = \mathbf{M}_1\mathbf{Q}_i\mathbf{B}'_i\mathbf{B}'_c\mathbf{Q}_c\mathbf{M}_1^{-1}$ are all fixed. In $\mathbf{P}_i$, there are n'^2 unknowns in $\mathbf{M}_1$, $\frac{n'^2-n'}{2}$ unknowns in $\mathbf{Q}_i$, n' unknowns in $\mathbf{B}'_i$, and $\frac{n'^2-n'}{2}$ unknowns in $\mathbf{Q}_c$. Thus, there are $2n'^2$ unknowns in total. For ease of exposition, we set $\mathbf{N}_1 = \mathbf{M}_1\mathbf{Q}_i\mathbf{B}'_i$ and $\mathbf{N}_2 = \mathbf{Q}_c\mathbf{M}_1^{-1}$. In this equivalent substitution, there still have $2n'^2$ unknowns in total because none of them is dropped during the multiplication process. Furthermore, since there only have $2n'^2$ elements at most in $\mathbf{N}_1$ and $\mathbf{N}_2$ which are both n'-dimensional matrices, $\mathbf{N}_1$ and $\mathbf{N}_2$ each will consist of n'^2 unknowns, respectively.

Thus, in the equation $\mathbf{P}_i = \mathbf{N}_1\mathbf{B}'_c\mathbf{N}_2$, for different requests $\mathbf{B}'_c$'s, the matrices $\mathbf{N}_1$ and $\mathbf{N}_2$ which have $2n'^2$ unknowns in total are both fixed all the time. Next we will show that none of the unknowns can be figured out no matter how many requests $\mathbf{B}'_c$ are issued based on this equation.

Based on our assumptions as above, all the n' entries in $\mathbf{B}'_c$ can be arbitrarily chosen by the server. Note that the matrix $\mathbf{B}'_c$ is transformed from its corresponding vector $\mathbf{B}_c$ as illustrated in [1]. As has been claimed in [1], the server can only generate at most n' linearly independent $\mathbf{B}_c$, which are denoted as $\mathbf{B}_{c1}, \ldots, \mathbf{B}_{cn'}$ and called a *basis* of n'-dimensional vector space. That is,

$$\mathbf{B}_c = x_1\mathbf{B}_{c1} + x_2\mathbf{B}_{c2} + \ldots + x_{n'}\mathbf{B}_{cn'}.$$

After the vectors are transformed into their matrix forms denoted as $\mathbf{B}'_{c1}, \ldots, \mathbf{B}'_{cn'}$, we will have

$$\begin{aligned} \mathbf{P}_i &= \mathbf{N}_1\mathbf{B}'_c\mathbf{N}_2 \\ &= \mathbf{N}_1(x_1\mathbf{B}'_{c1} + x_2\mathbf{B}'_{c2} + \ldots + x'_n\mathbf{B}'_{cn'})\mathbf{N}_2 \\ &= x_1\mathbf{N}_1\mathbf{B}'_{c1}\mathbf{N}_1^{-1} + \ldots + x'_n\mathbf{N}_2^{-1}\mathbf{B}'_{cn'}\mathbf{N}_2. \end{aligned} \tag{1}$$

Thus, any $\mathbf{P}_i$ can be represented by n' "basic" matrices $\mathbf{P}_j = \mathbf{N}_2^{-1}\mathbf{P}'_{cj}\mathbf{N}_1^{-1}$ $(j = 1, \ldots, n')$. In other words, the server can get at most n' linearly independent pairs of $(\mathbf{B}'_c, \mathbf{P}_i)$, and the knowledge of other pairs will not help solve for

$\mathbf{N}_1$ and $\mathbf{N}_2$. Without loss of generality, we assume the server chooses a *basis* $\mathbf{B}_{c1} = [1, 0, 0, \ldots, 0]$, $\mathbf{B}_{c2} = [0, 1, 0, \ldots, 0], \ldots, \mathbf{B}_{cn'} = [0, 0, \ldots, 0, 1]$ in the n'-dimensional vector space. We denote all the unknowns in $\mathbf{N}_1$ and $\mathbf{N}_2$ by q'_{ij} and p'_{ij} for $(1 \leq i, j \leq n')$ respectively, then we have

$$\begin{aligned}
\mathbf{P}_1 =& \mathbf{N}_1\mathbf{B}'_{c1}\mathbf{N}_2 = \begin{pmatrix} q'_{11} & \cdots & q'_{1n'} \\ q'_{21} & \cdots & q'_{2n'} \\ \vdots & \cdots & \vdots \\ q'_{n'1} & \cdots & q'_{n'n'} \end{pmatrix} \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & 0 & 0 \end{pmatrix} \begin{pmatrix} p'_{11} & \cdots & p'_{1n'} \\ p'_{21} & \cdots & p'_{2n'} \\ \cdots & \cdots & \cdots \\ p'_{n'1} & \cdots & p'_{n'n'} \end{pmatrix} \\
&= \begin{pmatrix} q'_{11}p'_{11} & \cdots & q'_{11}p'_{1n'} \\ \cdots & \cdots & \cdots \\ q'_{n'1}p'_{11} & \cdots & q'_{n'1}p'_{1n'} \end{pmatrix} \\
&\vdots \\
\mathbf{P}_{n'} =& \mathbf{N}_1\mathbf{B}'_{cn'}\mathbf{N}_2 \\
&= \begin{pmatrix} q'_{1n'}p'_{n'1} & \cdots & q'_{1n'}p'_{n'n'} \\ \cdots & \cdots & \cdots \\ q'_{n'n'}p'_{n'1} & \cdots & q'_{n'n'}p'_{n'n'} \end{pmatrix}.
\end{aligned}$$

It is easy to find that all the above equations contain different unknowns, namely each unknown only occurs in one of the above equations. Thus, we only need to consider one matrix equation of the above if the server wants to figure out any elements of $\mathbf{N}_1$ and $\mathbf{N}_2$. Taking $\mathbf{P}_1$ for example, we assume all the constants in $\mathbf{P}_1$ are denoted as A_{ij} for $(1 \leq i, j \leq n')$, that is

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n'} \\ A_{21} & A_{22} & \cdots & A_{2n'} \\ \vdots & \vdots & & \vdots \\ A_{n'1} & A_{n'2} & \cdots & A_{n'n'} \end{pmatrix} = \begin{pmatrix} q'_{11} & 0 & \cdots & 0 \\ 0 & q'_{21} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & q'_{n'1} \end{pmatrix} \begin{pmatrix} p'_{11} & p'_{12} & \cdots & p'_{1n'} \\ p'_{11} & p'_{12} & \cdots & p'_{1n'} \\ \vdots & \vdots & & \vdots \\ p'_{11} & p'_{12} & \cdots & p'_{1n'} \end{pmatrix}. \tag{2}$$

By leveraging linear transformations, we can get the following equivalent matrix equation

$$\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n'} \\ A_{21} - \frac{q'_{21}}{q'_{11}}A_{11} & A_{22} - \frac{q'_{21}}{q'_{11}}A_{12} & \cdots & A_{2n'} - \frac{q'_{21}}{q'_{11}}A_{1n'} \\ \vdots & \vdots & & \vdots \\ A_{n'1} - \frac{q'_{n'1}}{q'_{11}}A_{11} & A_{n'2} - \frac{q'_{n'1}}{q'_{11}}A_{12} & \cdots & A_{n'n'} - \frac{q'_{n'1}}{q'_{11}}A_{1n'} \end{pmatrix} = \begin{pmatrix} q'_{11}p'_{11} & q'_{11}p'_{12} & \cdots & q'_{11}p'_{1n'} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \tag{3}$$

Note that, although it *seems* that n'^2 equations with $2n'$ unknowns can be obtained from the first sight of the matrix equation (2). However, after the linear transformation, it can be seen that only $2n' - 1$ equations indeed can be established with strictly $2n'$ unknowns shown in equation (3), the remaining equations are all *linearly dependent*. In other words, there exists an unlimited number of solutions for $\mathbf{N}_1$ and $\mathbf{N}_2$. Hence, we have shown that the attack method proposed in [2] by exploiting the equations $\mathbf{P}_i$ doesn't work on CloudBI-II at all.

References

1. Qian Wang, Shengshan Hu, Kui Ren, Meiqi He, Minxin Du, and Zhibo Wang. Cloudbi: Practical privacy-preserving outsourcing of biometric identification in the cloud. In *ESORICS 2015*, pages 186–205. Springer, 2015.
2. Jiawei Yuan. Security attack on cloudbi: Practical privacy-preserving outsourcing of biometric identification in the cloud. Cryptology ePrint Archive, Report 2015/1257, 2015. http://eprint.iacr.org/2015/1257.pdf/.