POSTER: Towards Privacy-Preserving Biometric Identification in Cloud Computing

Changhee Hahn
Department of Computer Science and Engineering
Korea University
Seoul, Republic of Korea
hahn850514@korea.ac.kr

Junbeom Hur
Department of Computer Science and Engineering
Korea University
Seoul, Republic of Korea
jbhur@korea.ac.kr

ABSTRACT

Wang et al. recently proposed a privacy-preserving biometric identification scheme. However, the security assumption of the scheme does not capture practical aspects of real world attacks. In this paper, we consider a practical attack model which results in the leakage of biometric data in Wang et al.'s scheme. We first show the feasibility of our attack model and demonstrate how an attacker is able to recover the biometric data. Then, we propose a new biometric identification scheme that is secure against the attack model.

Keywords

Biometric identification; privacy; cloud

1. INTRODUCTION

Biometric identification systems are composed of enrollment and identification phase [1]. Wang et al.'s recently proposed a privacy-preserving biometric identification scheme which considers an attack model where adversaries attack the system only in the identification phase [2]. In real world applications, however, any user can enroll himself at any time. Thus, adversaries may attack the system not only in identification phase but also in enrollment phase by pretending to be legitimate users and registering false information into the database. For example, a real world attack was found on Oracle database, where an attacker registers a malicious instance which is used to bypass authentication [3]. This clearly demonstrates that biometric identification systems should consider possible attacks in enrollment phase. However, we found that the previous Wang et al.'s scheme [2] is not secure against the attack during the enrollment phase.

In this paper, we consider a comprehensive attack model which captures more pragmatic attack scenarios in both the enrollment and identification phases. We then show how the proposed attack model leads to a security breach in [2]. Next, we propose a privacy-preserving biometric identification scheme.

2. BIOMETRIC IDENTIFICATION

Prior to our attack model, we briefly explain Wang et al.'s biometric (specifically, fingerprint-based) identification scheme [2].

Preliminaries. In linear algebra, the trace of a square matrix M, namely $tr(M)$, is the sum of all diagonal entries in M. Given an invertible matrix M_1 and a lower triangular matrix Q with diagonal entries set to 1, the following equation holds: $tr(M) = tr(MQ) = tr(QM) = tr(M_1MM_1^{-1}) = tr(M_1MQM_1^{-1}) = tr(M_1QMM_1^{-1})$. Also, Given two square matrices A and B with the same size and a scalar c, $tr(A+B) = tr(A) + tr(B)$ and $tr(cA) = c \cdot tr(A)$ hold.

A fingerprint can be represented as an integer vector, called FingerCode. Two fingerCodes are considered to belong to a same user if a Euclidean distance between those FingerCodes is smaller than some threshold.

2.1 Enrollment Phase

A user encodes a fingerprint image into a vector, called FingerCode $b_i = [b_{i1}, ..., b_{in}]$ where $n = 640$, and b_{ij} is an 8-bit integer for $1 \leq j \leq n$. The user submits b_i to the database owner, who extends bi to an $(n+2)$-dimensional vector $B_i = [bi1, ..., b_{in}, b_{i(n+1)}, b_{i(n+2)}]$, where $b_{i(n+1)} = \frac{-1}{2}(b_{i1}^2 + ... + b_{in}^2)$ and $b_{i(n+2)} = 1$. B_i is further extended to an $(n+2) \times (n+2)$ matrix B_i' as

$$B_i' = \begin{pmatrix} b_{i1} & 0 & 0 & \cdots & 0 & 0 \\ 0 & b_{i2} & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & b_{in} & 0 & 0 \\ 0 & 0 & \cdots & 0 & b_{i(n+1)} & 0 \\ 0 & 0 & 0 & \cdots & 0 & b_{i(n+2)} \end{pmatrix}. \tag{1}$$

For each $B_i^{'}$, the database owner computes $C_i = M_1Q_iB_i^{'}M_2$, where Q_i is a randomly generated $(n+2)\times(n+2)$ lower triangular matrix with diagonal entries set to 1, and (M_1, M_2) is a pair of $(n+2)\times(n+2)$ invertible matrices with all entries chosen randomly. The database owner then sends C_i to the cloud and keeps (M_1, M_2) hidden. Thus, the fingerprint is encrypted and then outsourced to the cloud.

2.2 Identification Phase

Given a candidate fingerprint image for identification, a user generates the corresponding FingerCode $b_c = [b_{c1}, ..., b_{cn}]$

CCS'16 October 24-28, 2016, Vienna, Austria

ACM ISBN 978-1-4503-4139-4/16/10.

DOI: http://dx.doi.org/10.1145/2976749.2989048

and submits it to the database owner, who then extends it to an $(n+2)$-dimensional vector $B_c = [b_{c1}, b_{c2}, ..., b_{cn}, 1, r_c]$, where r_c is a random value. B_c is further extended to $B_c^{'}$ as in Eq. 1. The database owner randomly generates an $(n+2) \times (n+2)$ lower triangular matrix Q_c with diagonal entries set to 1, encrypts the plaintext query as $C_F = M_2^{-1} B_c^{'} Q_c M_1^{-1}$, and sends C_F to the cloud. Given C_i and C_F, the cloud computes

$$\begin{aligned} C_i C_F &= M_1 Q_i B_i^{'} M_2 M_2^{-1} B_c^{-1} Q_c M_1^{-1} \\ &= M_1 Q_i B_i^{'} B_c^{'} Q_c M_1^{-1}. \end{aligned}$$

Due to the property of *trace*, the following equation holds:

$$tr\left(C_i C_F\right) = tr\left(B_i^{'} B_c^{'}\right) = \sum_{j \in n} b_{ij} b_{cj} + b_{i(n+1)} + r_c.$$

Suppose b_i and b_z are enrolled FingerCodes. Let $dist_{ic}$ denote the Euclidean distance between two FingerCodes.

$$\begin{aligned} & 2\left(tr(B_z^{'} B_c^{'}) - tr(B_i^{'} B_c^{'})\right) \\ = \;& 2(\sum_{j \in n} b_{zj} b_{cj} - \frac{1}{2} \sum_{j \in n} b_{zj}^2) - 2(\sum_{j \in n} b_{ij} b_{cj} - \frac{1}{2} \sum_{j \in n} b_{ij}^2) \\ = \;& \sum_{j \in n} (b_{ij} - b_{cj})^2 - \sum_{j \in n} (b_{zj} - b_{cj})^2 \\ = \;& dist_{ic}^2 - dist_{zc}^2. \qquad (2) \end{aligned}$$

The cloud is then able to determine $dist_{ic} \geq dist_{ze}$ by checking if $2(tr(B_z^{'} B_c^{'}) - tr(B_i^{'} B_c^{'})) \geq 0$, otherwise $dist_{ic} \leq dist_{ze}$.

3. ON THE INSECURITY OF WANG ET AL.'S SCHEME IN OUR ATTACK MODEL

3.1 Attack Model

We assume that an attacker is allowed to submit arbitrary FingerCodes of interest to the database owner for enrollment. He then may collude with the cloud in order to recover any fingerprint submitted for identification. Compared to the attack model proposed in [2] where an attack does not appear in the enrollment phase, we allow the attacker to make enrollment queries to the database owner. That is, malicious users who are colluding with the cloud submit arbitrary FingerCodes to the database owner for enrollment. If those information are well crafted, i.e., FingerCodes look like genuine ones, then the database owner is not able to distinguish them from real FingerCodes. Thus, it is reasonable that the attacker's FingerCode is enrolled without being caught.

3.2 Leakage of Query FingerCode

In our attack, the attacker who colludes with the cloud submits multiple *decoyFingerCodes* to the database owner for enrollment, and then these are used to recover the victim's FingerCode during the identification phase.

Our attack. We exploit a set of decoy FIngerCode pairs $\{(x_i, y_i) | 1 \leq i \leq n\}$ to recover the victim's FingerCode b_c. Each pair (x_i, y_i) is used to reveal b_{ci}, the i-th entry in b_c.

Recovering FingerCodes. We choose a decoy FingerCode $x_i = [x_{i1}, \cdots, x_{in}]$, where x_{ij} are 8-bit integers chosen randomly. We then choose another decoy FingerCode $y_i = [y_{i1}, \cdots, y_{in}]$, where y_{ij} are random values if $i = j$, otherwise $y_{ij} = x_{ij}$. For example, if $i = 1$, then $x_{11} \neq y_{11}$, and $x_{1j} = y_{1j}$ for all $j \in \{j | 2 \leq j \leq n\}$. These two decoy FingerCodes are then enrolled as follows. x_i and y_i are extended to $X_i = [x_{i1}, \cdots, x_{in}, x_{i(n+1)}, 1]$ and $Y_i = [y_{i1}, \cdots, y_{in}, y_{i(n+1)}, 1]$ respectively, where $x_{i(n+1)} = -0.5 \sum_{j=1}^{n} x_{ij}^2$ and $y_{i(n+1)} = -0.5 \sum_{j=1}^{n} y_{ij}^2$. (X_i, Y_i) is then extended to $(X_i^{'}, Y_i^{'})$ as in Eq. 1. The database owner computes $C_{x_i} = M_1 Q_{x_i} X_i^{'} M_2$ and $C_{y_i} = M_1 Q_{y_i} Y_i^{'} M_2$, where Q_{x_i} and Q_{y_i} are chosen randomly. C_{x_i} and C_{y_i} are then sent to the cloud.

Given the identification query $C_F = M_2^{-1} B_c^{'} Q_c M_1^{-1}$, the following condition holds due to trace property:

$$2(tr(C_{x_i} C_F) - tr(C_{y_i} C_F)) = 2(tr(Y_i^{'} B_c^{'}) - tr(X_i^{'} B_c^{'})),$$

where

$$\begin{aligned} tr(Y_i^{'} B_c^{'}) &= y_{ii} b_{ci} + \sum_{j \neq i} y_{ij} b_{cj} + y_{i(n+1)} + r_c, \\ tr(X_i^{'} B_c^{'}) &= x_{ii} b_{ci} + \sum_{j \neq i} x_{ij} b_{cj} + x_{i(n+1)} + r_c. \end{aligned}$$

Note that the following $(\sum_{j \neq i} y_{ij} b_{cj} = \sum_{j \neq i} x_{ij} b_{cj})$ holds because $y_{ij} = x_{ij}$ for all $j \in \{j | j \neq 1, 1 \leq j \leq n\}$. We then have

$$b_{ci} = \frac{(tr(Y_i^{'} B_c^{'}) - tr(X_i^{'} B_c^{'})) - (y_{i(n+1)} - x_{i(n+1)})}{y_{ii} - x_{ii}}. \qquad (3)$$

Since $tr(Y_i^{'} B_c^{'})$, $tr(X_i^{'} B_c^{'})$, y_{ii}, $y_{i(n+1)}$, xii, and $x_{i(n+1)}$ are all known to the cloud, the right-hand side of Eq. 3 is computable. Thus, the cloud can recover b_{ci}. By repeating our attack for all $1 \leq i \leq n$, the victim's FingerCode b_c is leaked.

4. OUR CONSTRUCTIONS

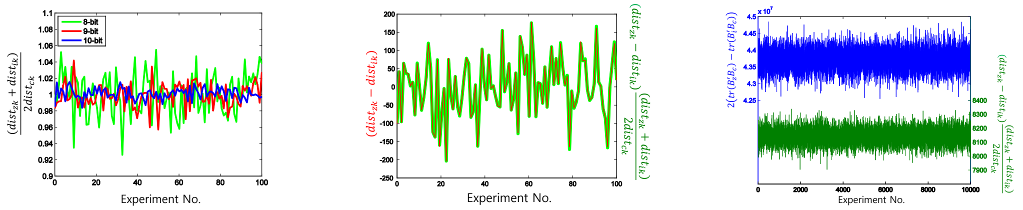
We adopt the same enrollment and identification processes depicted in Section 2, except for a one-time random mask k_c at identification phase. Accordingly, we assume i-th and z-th users have enrolled their fingerprints: $C_i = M_1 Q_i B_i^{'} M_2$ and $C_z = M_1 Q_i B_z^{'} M_2$ are in the cloud.

Secure Identification. Given a FingerCode b_c, the user generates a random mask $k_c = [k_{c1}, \cdots, k_{cn}]$ which is extended to $(n+2) \times (n+2)$ matrix K_c as

$$K_c = \begin{pmatrix} k_{i1} & 0 & 0 & \cdots & 0 & 0 \\ 0 & k_{i2} & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & k_{in} & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}.$$

He then computes $B_c^{'} = (2dist_{ck} \cdot B_c) + K_c$, and sends $B_c^{'}$ to the database owner for identification, where $dist_{ck}$ is the distance between b_c and k_c. By the identification process in Section 2.2, the cloud obtains $C_F = M_2^{'} B_c^{'} Q_c M_1^{-1}$, and computes $C_i C_F$ such that

$$\begin{aligned} tr(M_1 Q_i B_i^{'} B_c^{'} Q_c M_1^{-1}) &= tr(B_i^{'} B_c^{'}) \\ &= tr(B_i^{'} (2dist_{ck} \cdot B_c + K_c)) \\ &= tr(B_i^{'} \cdot 2dist_{ck} \cdot B_c + B_i^{'} K_c) \\ &= 2dist_{ck} \cdot tr(B_i^{'} B_c) + tr(B_i^{'} K_c). \end{aligned}$$



(a) The size of C at different masks (b) Comparison of B and B×C (c) Comparison of A and B×C

Figure 1: Experimental result of our scheme

By Eq. 2, we have

$$
\begin{aligned}
&2\left(tr(B_z^{'}B_c^{'})-tr(B_i^{'}B_c^{'})\right)\\
=&2(\sum_{j\in n}b_{zj}(2dist_{ck}\cdot b_{cj}+k_{cj})-\frac{2dist_{ck}}{2}\sum_{j\in n}b_{zj}^2+r_c)\\
&-2(\sum_{j\in n}b_{ij}(2dist_{ck}\cdot b_{cj}+k_{cj})-\frac{2dist_{ck}}{2}\sum_{j\in n}b_{ij}^2+r_c)\\
=&2dist_{ck}\cdot 2(tr(B_z^{'}B_c)-tr(B_i^{'}B_c))+2(\sum_{j\in n}b_{zj}k_{cj}-\sum_{j\in n}b_{ij}k_{cj}),
\end{aligned}
\tag{4}
$$

where $2(\sum_{j\in n}b_{zj}k_{cj}-\sum_{j\in n}b_{ij}k_{cj})$ is *noise* derived from K_c.

Discussion. By carrying out the attack described in Section 3.2, the cloud obtains $b_{ci}+k_{ci}$, where k_{ci} is one-time mask chosen randomly. Since both b_{ci} and k_{ci} are unknown, the cloud cannot deduce any of it from $b_{ci}+k_{ci}$. However, the noise is undesirable because it may affect the outcome of Eq. 4. That is, the noise may influence the correctness of the identification result.

As of writing the paper, there is no clear way to eliminate noise from Eq. 4. Still, the noise can be *fairly* negligible at the cost of $(\sum_{j=1}^n b_{zj}^2, \sum_{j=1}^n b_{ij}^2, dist_{ck})$ as side information. We demonstrate how it is possible in the remainder of the paper.

Scale-down of Noise. We reduce the influence of noise by adding $\sum_{j=1}^n b_{zj}^2 - \sum_{j=1}^n b_{ij}^2$ to Eq. 4. At the right-side of Eq. 4, the following equation holds.

$$
\begin{aligned}
&2(\sum_{j\in n}b_{zj}k_{cj}-\sum_{j\in n}b_{ij}k_{cj})+\sum_{j=1}^n b_{zj}^2-\sum_{j=1}^n b_{ij}^2\\
=&2(\sum_{j\in n}b_{zj}k_{cj}+\frac{1}{2}\sum_{j=1}^n b_{zj}^2-\frac{1}{2}\sum_{j=1}^n k_{cj}^2)\\
&-2(\sum_{j\in n}b_{ij}k_{cj}+\frac{1}{2}\sum_{j=1}^n b_{ij}^2-\frac{1}{2}\sum_{j=1}^n k_{cj}^2)\\
=&dist_{zk}^2-dist_{ik}^2.
\end{aligned}
$$

Thus, we have

$$2dist_{ck}\cdot 2(tr(B_z^{'}B_c)-tr(B_i^{'}B_c))+(dist_{zk}^2-dist_{ik}^2).$$

Note that from the above expressions, we re-construct the noise in the form of *noisy distance*, e.g., $dist_{zk}^2$ is the distance between b_z and k_c. We then divide it by $2dist_{ck}$, so we have

$$2(tr(B_z^{'}B_c)-tr(B_i^{'}B_c))+(dist_{zk}-dist_{ik})\frac{(dist_{zk}+dist_{ik})}{2dist_{ck}}.$$

As a result, the noise has the form of $(dist_{zk}-dist_{ik})\frac{(dist_{zk}+dist_{ik})}{2dist_{ck}}$.

Simulation. It is evident that, as long as $|2(tr(B_z^{'}B_c)-tr(B_i^{'}B_c))|>|(dist_{zk}-dist_{ik})\frac{(dist_{zk}+dist_{ik})}{2dist_{ck}}|$ holds, the noise does not affect the identification result. To prove it, we generate $30,000$ arbitrary FingerCodes, where we set $n=640$. The first $10,000$ FingerCodes are used as queries, next $10,000$ are f_z, and remaining $10,000$ are f_i. Then, we compute $2(tr(B_z^{'}B_c)-tr(B_i^{'}B_c))$, $(dist_{zk}-dist_{ik})$, and $\frac{(dist_{zk}+dist_{ik})}{2dist_{ck}}$ (for brevity, we denote each of them as A, B, and C respectively.).

Fig. 1 shows the simulation result of our scheme. In Fig. 1a, we measure C at different size of masks, e.g., we draw the green line when the size of k_{ci} is 8-bit. As the size of masks grows, C converges on 1. It means that C remains almost invariant, irrespective of FingerCodes. Fig. 1b shows the comparison result of B and B×C at 8-bit mask. Since C remains close to 1, B is highly similar to B×C. Therefore, two figures indicate that C has little influence on the noise. Next, Fig. 1c shows the comparison result of A and B×C, where A is always *much* bigger than B×C, i.e., the smallest value of A is bigger than 4.25×10^7, while the biggest value of B×C is smaller than $8,400$. Accordingly, Fig. 1c demonstrates that B×C has little effect on A. Thus, the noise does not influence the identification result.

5. CONCLUDING REMARKS

In this paper, we analyze the security of Wang et al.'s biometric identification scheme. We show that any FingerCode submitted for identification is leaked under our attack model even if it is encrypted. We propose a secure identification scheme with some noise during identification, and we prove that the noise has little effect on the identification result.

6. ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. 2016R1A2A2A05005402).

7. REFERENCES

[1] M. Barni, T. Bianchi, D. catalano, M.D. Raimondo, R.D. Labati, and P. Faillia. Privacy-preserving fingercode authentication. *In MM&Sec'*, pages 231–240. 2010.

[2] Q. Wang, S Hu, K Ren, M. He, M. Du, and Z. Wang. CloudBI: Practical privacy-preserving outsourcing of biometric identification in the cloud. *In Computer Security-ESORICS*, pages 186–205. 2015.

[3] Oracle Database TNS Listener Poison Attack. http://www.oracle.com/technetwork/topics/security/alert-cve-2012-1675-1608180.html.