# Security and Privacy Enhancement for Outsourced Biometric Identification

Kai Zhou, Jian Ren and Tongtong Li
Department of ECE, Michigan State University, East Lansing, MI 48824-1226
Email: {zhoukai, renjian, tongli}@msu.edu

***Abstract*—A lot of research has been focused on secure outsourcing of biometric identification in the context of cloud computing. In such schemes, both the encrypted biometric database and the identification process are outsourced to the cloud. The ultimate goal is to protect the security and privacy of the biometric database and the query templates. Security analysis shows that previous schemes suffer from the enrolment attack and unnecessarily expose more information than needed. In this paper, we propose a new secure outsourcing scheme aims at enhancing the security from these two aspects. First, besides all the attacks discussed in previous schemes, our proposed scheme is also secure against the enrolment attack. Second, we model the identification process as a fixed radius similarity query problem instead of the kNN search problem. Such a modelling is able to reduce the exposed information thus enhancing the privacy of the biometric database. Our comprehensive security and complexity analysis show that our scheme is able to enhance the security and privacy of the biometric database and query templates while maintaining the same computational savings from outsourcing.**

***Index Terms*—Cloud computing, secure outsourcing, biometric identification, security and privacy**

## I. Introduction

Remote Storage and computation outsourcing are two integer services provided by cloud computing. Data owners such as individuals or organizational administrators are able to outsource to the cloud their private data for storage as well as some computational intensive tasks for computation. Various works [8]–[10] have been devoted to securing the outsource process, i.e., ensuring the security of the private data while still enjoying the convenience provided by cloud computing.

Among various applications, outsourced biometric identification is of special interest. This is because, on one hand, the biometric database itself is of huge size thus making cloud storage an appealing solution. On the other hand, the identification process is computationally expensive due to the large database size. As a result, several schemes have been proposed to outsource biometric identification to the cloud. The ultimate goal of these schemes is to protect the security and privacy of the biometric templates in the database as well as the query templates under different attacks.

Securing outsourcing of biometric identification has attracted much research effort. In [1], the authors proposed two different schemes where both single-server model and multiple-servers model are considered. However, the scheme for the single-server model still has prohibitive computational overhead for large databases thus making it less practical. In the multiple-servers model, it is required that the severs would not collude, otherwise the security of the biometric templates is compromised. Following works [2], [3] considered two non-colluding servers thus suffering from the same drawbacks. In [7], a privacy-preserving biometric identification scheme was proposed for the single-server model. However, such a scheme is not secure under an active attack where the cloud is able to collude with query users. Most recently, the authors in [6] proposed a secure outsourcing scheme that can defend against three different attacks as defined in [6], which include the attack where cloud and users are allowed to collude. The basic idea is to model the identification process as a Nearest Neighbor (NN)search problem. That is, given an encrypted query template, the nearest neighbor in the database is identified and returned to the data owner, who can later decide whether these two templates belong to the same individual.

Although the scheme in [6] is efficient and can defend against relatively severe attacks as to date, recent analysis [4] shows that it still suffers the enrollment attack. That is, if the cloud is allowed to inject selected templates into the database, then the cloud is able to recover the query templates based on the intermediate computation results. Also, we point out that modelling the identification process as kNN search problem has inherent limitations in terms of template privacy. To be more specific, in [6], given two encrypted templates $\mathbf{x}$ and $\mathbf{y}$ and an encrypted query template $\mathbf{z}$, the cloud can determine which template ($\mathbf{x}$ or $\mathbf{y}$) is closer to $\mathbf{z}$. Repeating this process, the cloud is able to identify the closet template to $\mathbf{z}$. As a result, the relative distance information among the templates in the database is inevitably exposed.

To deal with the above two security issues, we propose a new secure outsourcing schemes for biometric identification. Especially, our proposed scheme can defend against all the attacks defined in [6] as well as the enrollment attack. Moreover, we model the identification process as a fixed-radius similarity query problem rather than kNN search problem. That is, our scheme only enables the cloud to identify the templates within a fixed radius of the query template. No more information about the distance is revealed. In this way, less information is exposed compared to the scheme in [6], thus enhancing the privacy of the biometric database.

The rest of the paper is organized as follows. We introduce the system model and threat model in Section II. Then the secure outsourcing scheme is proposed in Section III. The security and complexity analysis is given in Section IV. In

978-1-5386-4727-1/18/$31.00 ©2018 IEEE

Section V, we present some numeric results showing the efficiency of our proposed scheme. At last, we conclude in Section VI.

## II. Problem Formulation

### A. System model

We consider a system consisting of three parties: a data owner, a set of end-users and a cloud service provider. The data owner has a biometric database composed of a collection of users' biometric templates, where each template can be represented by an $n$-dimensional vector $\mathbf{T} = (t_1, t_2, \cdots, t_n)$. Each template is registered by an end-user during an enrolment stage. During a preparation stage, the data owner will pre-process the templates and outsource them to the cloud. Later, in an identification stage, an end-user will submit her identification request composed of a query template $\mathbf{T}_j$ to the data owner. The data owner will generate a token for each specific query template $\mathbf{T}_j$ and submit the token to the cloud. Then, the cloud is responsible for identifying the template $\mathbf{T}_i$ in the database such that $\mathsf{dist}(\mathbf{T}_i, \mathbf{T}_j) < \theta$, where $\mathsf{dist}()$ is a distance measurement function and $\theta$ is a pre-defined threshold. In words, the cloud will identify and return the template(s) whose distance from the query template is within a threshold.

### B. Threat model

In this paper, we consider a *semi-malicious* cloud model where the cloud will follow the protocols but is allowed to collude with some malicious end-users. To be specific, the collusion happens when some malicious end-users will submit query templates to the data owner and choose to share the templates with the cloud. As a result, the cloud is able to learn pairs of template and its encrypted form.

In summary, depending on the different capabilities of the adversaries, we propose two attack models as follows.

1) Passive Attack: the cloud is able to know the encrypted templates $\mathsf{Enc}(\mathbf{T}_i)$, for $i = 1, 2, \cdots, m$, where $\mathsf{Enc}(\mathbf{T}_i)$ is the encrypted form of $\mathbf{T}_i$. Also, the cloud is able to observe a series of $w$ encrypted queries $\mathsf{Enc}(\mathbf{T}_j)$, $j = 1, 2, \ldots, w$. However, the service provider does not know the underlying templates $\mathbf{T}_j$ in plaintext.
2) Active Attack: besides the encrypted templates $\mathsf{Enc}(\mathbf{T}_i)$, the cloud is able to observe a series of $w$ encrypted queries $\mathsf{Enc}(\mathbf{T}_j)$ as well as the corresponding plaintext $\mathbf{T}_j$, $j = 1, 2, \ldots, w$. As mentioned earlier, such an attack can happen when the cloud collude with malicious end-users.

Besides the above two attacks, we also allow the enrollment attack as considered in [4].

3) Enrollment Attack: the cloud is able to inject templates in the enrollment stage. That is, the cloud is able to have a series of $v$ encrypted templates $\mathsf{Enc}(\mathbf{T}_i)$ as well as the corresponding plaintext $\mathbf{T}_i$, $i = 1, 2, \ldots, v$.

Informally, the security requirement of the outsourced biometric identification against the above three attacks is that the cloud is not able to learn more information about the templates than what is allowed through the identification process. That is, the cloud is only able to decide whether the distance between two templates is within a threshold or not. It is not feasible for the service provider to derive any key information about the enrolled templates and the query templates.

## III. Secure Outsourcing of Biometric Identification

### A. Basic Framework

Intuitively, our proposed secure outsourcing scheme is consist of four phases. In the first phase, the data owner will generate the system parameters and a transformation key. Then, for each biometric template in the database, it is transformed to an encrypted form using the transformation key. The transformed database is then outsourced to the cloud. In the query phase, for every submitted query template, the data owner will generate a token using the same transformation key. At last, in the identification phase, the cloud will identify the template whose distance from the query template is within a pre-defined threshold.

Our proposed scheme is composed of the following five algorithms.

- $\mathsf{SetUp}() \rightarrow param$: the set up algorithm generate system parameters $param$.
- $\mathsf{KeyGen}() \rightarrow sk$: the key generation algorithm will generate transformation key $sk$.
- $\mathsf{Transform}(sk, \mathbf{x}) \rightarrow C_{\mathbf{x}}$: given a vector $\mathbf{x}$ and transformation key $sk$, the transformation algorithm will transform $\mathbf{x}$ into a disguised form $C_{\mathbf{x}}$.
- $\mathsf{TokenGen}(sk, \mathbf{y}) \rightarrow T_{\mathbf{y}}$: given a vector $\mathbf{y}$ and transformation key $sk$, the token generation algorithm will generate a token $T_{\mathbf{y}}$ for $\mathbf{y}$.
- $\mathsf{Evaluate}(C_{\mathbf{x}}, T_{\mathbf{y}}) \rightarrow \Lambda = \{0, 1\}$: given the transformed vector $C_{\mathbf{x}}$ and token $T_{\mathbf{y}}$, the evaluation algorithm will output a result $\Lambda$ satisfying

$$\Lambda = \begin{cases} 1, & \mathsf{dist}(\mathbf{x}, \mathbf{y}) \leq \theta \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathsf{dist}(\mathbf{x}, \mathbf{y})$ is the distance between $\mathbf{x}$ and $\mathbf{y}$ and $\theta$ is a pre-defined threshold.

### B. Secure Transformation and Evaluation

The essential part in our proposed scheme is the secure transformation and evaluation process. In a high level view, the enrolled templates are transformed through the $\mathsf{Transform}$ function and the $\mathsf{TokenGen}$ function will generate a token by transforming the query template. It is critical that given the transformed templates, it it computationally infeasible to recover the original vector. However, the $\mathsf{Evaluate}$ function is able to reveal some information of two templates. That is, whether the distance between the two templates is within a threshold or not.

Our transformation process is similar to the techniques utilized in [6]. However, the computational models as well as the security requirements are fundamentally different. We now give some intuition about our transformation and evaluation process. The detailed construction is presented in Protocol 1. Given a vector $\mathbf{x}$, we first extend it to $\mathbf{x}'$ by inserting the threshold $\theta$ and some random numbers. Then $\mathbf{x}'$ is transformed to a matrix from and is disguised by multiplying it with random matrix. Denote this disguised form as $C_x$. Given a query vector $\mathbf{y}$, the TokenGen function will transform $\mathbf{y}$ into a disguised form $C_y$. The Evaluate function takes in $C_x$ and $C_y$ as input and outputs $\alpha\beta(\mathbf{x} \circ \mathbf{y} - \theta)$, where $\alpha$ and $\beta$ are one-time random positive numbers associated with $\mathbf{y}$ and $\mathbf{x}$, respectively. By comparing $\alpha\beta(\mathbf{x} \circ \mathbf{y} - \theta)$ with 0, the cloud is able to determine whether the inner product of $\mathbf{x}$ and $\mathbf{y}$ is within the threshold $\theta$ or not. We note that since $\alpha$ and $\beta$ are one-time random numbers and are different for each template, the exact value of $(\mathbf{x} \circ \mathbf{y} - \theta)$ can be concealed from the final result. We note that the inner product $\mathbf{x} \circ \mathbf{y}$ is flexible to express different distance metrics between $\mathbf{x}$ and $\mathbf{y}$.

### C. The Proposed Scheme

In this section, we give the detailed implementation of our secure outsourcing scheme in Protocol 1. In the protocol, the result $I$ in function Evaluate is equal to $(\mathsf{dist}_E(\mathbf{x}, \mathbf{y}) - \theta)$, where $\mathsf{dist}_E(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$. We use the Euclidean distance as an example to measure the similarity between two templates. However, it is easy to design the vectors $\mathbf{x}$ and $\mathbf{y}$ such that the Evaluate function will give other distances such as the Hamming distance. The correctness of our proposed scheme is shown in Theorem 1.

*Theorem 1:* The proposed outsourcing scheme in Protocol 1 is correct. That is, $I = \alpha\beta(\mathsf{dist}_E(\mathbf{x}, \mathbf{y}) - \theta)$, where $\mathsf{dist}_E(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$.

*Proof:* For a square matrix $Y$, the trace $\mathsf{Tr}(Y)$ is defined as the sum of the diagonal entries of $Y$. Given an invertible matrix $M_1$ of the same size, the transformation $M_1 Y M_1^{-1}$ is called similarity transformation of $Y$. From linear algebra, we know the trace of a square matrix remains unchanged under similarity transformation. That is, $\mathsf{Tr}(Y) = \mathsf{Tr}(M_1 Y M_1^{-1})$. Then we have

$$I = \mathsf{Tr}(T_1) + \mathsf{Tr}(T_2) = \mathsf{Tr}(S_p P Y S_y) + \mathsf{Tr}(S_q Q Y S_y).$$

Since $S_y$, $S_p$ and $S_q$ are selected as lower triangular matrices, where all the diagonal entries are set to 1, the diagonal entries of $S_p P$, $S_q Q$ and $Y S_y$ are all the same as those of $P$, $Q$ and $Y$. Thus we have

$$I = \mathsf{Tr}(PY) + \mathsf{Tr}(QY).$$

Since $P$,$Q$ and $Y$ are diagonal matrices, we have $\mathsf{Tr}(PY) = \mathbf{p} \circ \mathbf{y}'$ and $\mathsf{Tr}(QY) = \mathbf{q} \circ \mathbf{y}'$. Thus

$$I = \mathbf{p} \circ \mathbf{y}' + \mathbf{q} \circ \mathbf{y}' = \mathbf{x}' \circ \mathbf{y}' = \alpha\beta(\mathsf{dist}_E(\mathbf{x}, \mathbf{y}) - \theta). \quad \blacksquare$$

## IV. Security and Complexity Analysis

### A. Security against Active Attack

We focus on the security of our proposed scheme under active attack since it implies the security under passive attack. We also utilize TokenGen function as the representative since

**Protocol 1** Secure Outsourcing of Biometric Identification

**Input:** $\mathbf{x} = \{x_1, \ldots, x_n\}, \mathbf{y} = \{y_1, \ldots, y_n\}, \theta$.
**Output:** $\Lambda = \{0, 1\}$.

$\mathsf{Setup}() \to param$:
1: Data owner sets $param = \{n, \theta\}$., where $n$ is the dimension of templates and $\theta$ is a pre-defined threshold.

$\mathsf{KeyGen}(\lambda) \to sk$:
1: Randomly generates two matrices $M_1$ and $M_2$ with dimension $(n+5) \times (n+5)$ and a permutation $\pi : \mathbb{R}^{n+5} \to \mathbb{R}^{n+5}$.
2: Set $sk = \{M_1, M_2, M_1^{-1}, M_2^{-1}, \pi\}$

$\mathsf{Transform}(sk, \mathbf{x}) \to C_{\mathbf{x}}$:
1: Generate random numbers $\beta$ and $r_x$.
2: (**Extend**) Extend $\mathbf{x}$ to an $(n+5)$-dimensional vector $\mathbf{x}' = (2\beta x_1, 2\beta x_2, \ldots, 2\beta x_n, -\beta \sum_{i=1}^{n} x_i^2, \beta, \beta\theta^2, r_x, 0)$.
3: (**Permute**) Permute $\mathbf{x}'$ to obtain $\mathbf{x}'' = \pi(\mathbf{x}')$.
4: Transform $\mathbf{x}''$ a diagonal matrices $X$ with $\mathbf{x}''$ being the diagonal.
5: Generate a random $(n+5) \times (n+5)$ lower triangular matrix $S_x$ with the diagonal entries fixed as 1. Compute $C_x = M_1 S_x X M_2$.

$\mathsf{TokenGen}(sk, \mathbf{y}) \to T_{\mathbf{y}}$:
1: On receiving a query template $\mathbf{y}$, data owner generates random numbers $r_y$ and $\alpha$.
2: (**Extend**) Extend $\mathbf{y}$ to an $(n+5)$-dimensional vector $\mathbf{y}' = (2\alpha y_1, 2\alpha y_2, \ldots, 2\alpha y_n, \alpha, -\alpha \sum_{i=1}^{n} y_i^2, \alpha, 0, r_y)$.
3: (**Permute**) Permute $\mathbf{y}'$ to obtain $\mathbf{y}'' = \pi(\mathbf{y}')$.
4: Transform $\mathbf{y}''$ to a diagonal matrix $Y$ with diagonal being $\mathsf{y}''$.
5: Generate a random $(n+5) \times (n+5)$ lower triangular matrix $S_y$ with the diagonal entries fixed as 1. Compute $C_y = M_2^{-1} Y S_y M_1^{-1}$
6: Send the token $C_y$ to the cloud.

$\mathsf{Evaluate}(C_{\mathbf{x}}, C_{\mathbf{y}}) \to \Lambda$:
1: For every transformed template $C_x$ in the database, the cloud computes $I = \mathsf{Tr}(C_x C_y)$, where $\mathsf{Tr}(\cdot)$ is the trace of a matrix.
2: Cloud sets $\Lambda = 1$ if $I \leq 0$, which means that the template $\mathbf{x}$ is identified; otherwise set $\Lambda = 0$.

the transformation process in Transform is similar. The basic idea is to show that an adversary cannot differentiate two transformed templates obtained from the TokenGen function. Thus, the adversary cannot learn key information from the disguised form of templates. We have the following theorem.

*Theorem 2:* The proposed outsourcing scheme is secure against active attack, that is the cloud cannot derive key information from transformed templates.

*Proof:* Consider the transformation of vector $\mathbf{y}$, where $\mathbf{y} = (y_1, \ldots, y_n)$. The vector $\mathbf{y}$ is first extended to $\mathbf{y}' =$

$(2\alpha y_1, 2\alpha y_2, \ldots, 2\alpha y_n, \alpha, -\alpha \sum_{i=1}^{n} y_i^2, \alpha, r)$. The vector $\mathbf{y}'$ is then extended to a diagonal matrix $Y$. Then, it is transformed to $C_y = M_2^{-1} Y S_y M_1^{-1}$, where $S_y$ is a random lower triangular matrix. We note that the product of $Y$ and $S_y$ will produce a lower triangular matrix denoted as $G_y$. Now we focus on the product $C_y = M_2^{-1} G_y M_1^{-1}$.

Denote the entries in $M_2^{-1}$ and $M_1^{-1}$ as $a_{ij}$ and $b_{ij}$, respectively, where $i, j = 1, 2, \ldots, n+5$. For matrix $G_y$, denote its non-zero entries in the lower triangular part as $s_{ij}$, where $i > j$ and $i, j = 1, 2, \ldots, n+5$. Then, by law of matrix multiplication, each entry $c_{ij}$ in $C_y$ can be written in the form of

$$\begin{aligned} c_{ij} &= \sum f_{ij}^1(a_{ij}, b_{ij}) m_i + f_{ij}^2(a_{ij}, b_{ij})\alpha \\ &\quad + f_{ij}^3(a_{ij}, b_{ij}) r + f_{ij}^4(a_{ij}, b_{ij}, s_{ij}), \end{aligned} \tag{1}$$

where $f_{ij}^t$, $t = 1, 2, 3, 4$ are polynomials. Equation (1) is obtained by summing up each terms of $m_i$, $\alpha$ and $r$, respectively.

In the transformation process, $a_{ij}$ and $b_{ij}$ are fixed. $\alpha$,$r$ and $s_{ij}$ are one-time random numbers. $m_i$ are chosen and can be controlled by the adversary. However, since $\alpha$,$r$ and $s_{ij}$ are one-time random numbers, the polynomials $f_{ij}^2(a_{ij}, b_{ij})\alpha$, $f_{ij}^3(a_{ij}, b_{ij})r$ and $f_{ij}^4(a_{ij}, b_{ij}, s_{ij})$ all looks random to the adversary. As a result, the summation $c_{ij}$ is random. This means that, for any two templates chosen by the adversary and one transformed template, the adversary cannot distinguish which template is actually transformed. As a result, the adversary cannot derive key information from transformed templates. ■

The other important aspect of security is to what extent the Evaluate function can reveal information of the templates. It is clear that Evaluate will the distance information which is necessary for identification. However, we note that every vector $\mathbf{y}$ is associated with a one-time independent random number $\alpha$ and every vector $\mathbf{x}$ is associated with a one-time random number $\beta$. As a result, in the active attack, what an adversary can observe through Evaluate function is a series of results $I_i = \alpha\beta_i(\mathsf{dist}_E(\mathbf{x}, \mathbf{y}) - \theta)$. Since $\beta_i$ are selected independently, the final results $I_i$ only reveals whether $\alpha\beta_i(\mathsf{dist}_E(\mathbf{x}, \mathbf{y}) - \theta)$ is positive or not. No more key information can be derive from $I_i$.

### B. Security against Enrolment Attack

As mentioned earlier, an enrolment attack was proposed in [4] making the secure outsourcing scheme in [6] vulnerable. In an enrolment attack, an adversary (i.e., the cloud) is able to inject known templates into the database. During evaluation, the cloud is able to derive the following equation (i.e., Equation (3) in [4]):

$$b_{ci} = \frac{(\mathsf{Tr}(Y_i' B_c') - \mathsf{Tr}(X_i' B_c')) - (y_{i(n+1)} - x_{i(n+1)})}{y_{ii} - x_{ii}},$$

where $b_{ci}$ is the $i$-th entry in a submitted query template $\mathbf{b}_c$. Since $\mathsf{Tr}(Y_i' B_c')$ and $\mathsf{Tr}(X_i' B_c')$ are computable and $\mathbf{x}$ and $\mathbf{y}$ are selected by the cloud, the cloud is able to recover $b_{ci}$. Repeating such attack will finally recover the whole query template $\mathbf{b}_c$ as demonstrated in [4].

We now show that our proposed scheme is secure under the above enrolment attack. The underlying reason that the scheme in [6] cannot defend such attack is that the evaluate function will cancel all the randomness (i.e., the random lower-triangular matrix) introduced in the encryption process. In comparison, the evaluation function in our scheme will give $\alpha_z\beta_x(\mathsf{dist}^2(\mathbf{x}, \mathbf{z}) - \theta)$, where $\alpha_z$ and $\beta_x$ are one-time random numbers associated with the templates $\mathbf{z}$ and $\mathbf{x}$ respectively. As a result, Equation (3) in [4] is modified to

$$b_{ci} = \frac{(\mathsf{Tr}(PB_c') - \mathsf{Tr}(QB_c')) - (p_{n+1} - q_{n+1})}{\alpha_c\beta_x(p_n - q_n)}.$$

Note that $\alpha_c$ is a one-time random number associated with a query $b_c$ and $\beta_x$ is a one-time random number associated with $\mathbf{x}$. Thus, although the adversary is able to insert known templates into the database, it cannot derive $b_{ci}$ due to the one-time randomness. In other words, our proposed outsourcing scheme is able to defend against the enrolment attack.

### C. The Effect of Randomness on Security

It is important to understand the effect of different randomness on security. We briefly categorize the one-time randomness utilized by our scheme into three types.

- Type I: *result-disguising randomness*. In the **Extend** step in both Transform and TokenGen, we use random $\beta$ and $\alpha$ respectively to multiple with each entry of $\mathbf{x}$ and $\mathbf{y}$. Since $\alpha$ and $\beta$ will remain in the decryption result, we name it as result-disguising randomness.
- Type II: *vector-extension randomness*. In TokenGen, we extend the vector $\mathbf{y}$ and pad it with a random $r$.
- Type III: *matrix-multiplication randomness*. In both Transform and TokenGen, we multiple the extended matrices ($P$, $Q$ and $Y$) with random matrices ($S_p$, $S_q$ and $S_y$).

The Evaluate function will calculate the trace of the matrix (e.g., $C_pC_y$). We note that the function $\mathsf{Tr}(\cdot)$ will cancel Type II and Type III randomness. However, Type I randomness will remain in the evaluation result. This is important since it will only reveal partial information of the plaintext, which is just sufficient for the purpose of biometric authentication. Also, the underlying reason that the scheme in [6] is vulnerable to enrollment attack is that it lacks Type I randomness.

### D. Complexity Analysis

We focus on the complexity analysis of TokenGen and Evaluate since they are executed repeatedly in the identification process while SetUp, KeyGen and Transform are one-time processes. As shown in Protocol 1, it is obvious that the computational bottleneck of TokenGen and Evaluate lies in matrix multiplication. Without loss of generality, we assume that the matrices involved in the computation all have the same dimension $n \times n$.

The function TokenGen will take 3 matrix multiplications. Since matrix multiplication generally has a complexity of $\mathcal{O}(n^3)$ without optimization, the complexity of TokenGen is also $\mathcal{O}(n^3)$. In the function Evaluate, the trace of two matrices $C_pC_y$ and $C_qC_y$ need to be computed. We note that there is no need to calculate the matrix multiplication first. What needs to be computed are the main diagonals of the two matrices. Thus, Evaluate has a complexity of $\mathcal{O}(n^2)$.

In terms of communication overhead, we assume that each entry in the matrix or vector has the same size $l$. For each template in the database, the data owner needs to outsource the encrypted template $C_x = \{C_p, C_q\}$ to the cloud. Thus the communication overhead is $2n^2l$. Similarly, the communication overhead for each query template is $n^2l$.

## V. Numeric Results

The most important parameter that affects the performance of the identification process is the length of the vectors denoted as $n$. It will determine the execution time for both Transform and Evaluate, which are executed frequently in the querying process. Another parameter is the size of the database $N$. However, since our identification algorithm is basically a linear scan of the database, it can be predicted that the time for identification is also linear to $N$. Thus, it is of more interest to measure the performance of Transform and Evaluate in terms of the dimension $n$.

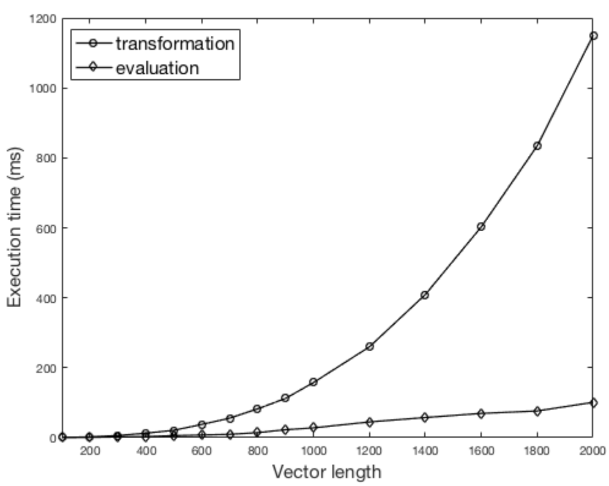


Fig. 1. Template transformation and evaluation time for each template

The simulation is conducted in a personal computer with 1.6 GHz Intel Core i5 CPU, 4 GB RAM and macOS Version 10.12.6. The algorithm is implemented using the Armadillo C++ linear algebra library. In the simulation, we let the length of the vector vary from 100 to 2000, which is able to cover the length of some typical biometric templates such as FingerCodes [5]. The execution time of Transform and Evaluate for each template is presented in Fig.1. We can see that both TokenGen and Evaluate are quite efficient. For example, it takes around 1 second to generate a token for a template with length 2000, which is quite long in real applications. The numeric results also correspond with the complexity analysis that Transform has $\mathcal{O}(n^3)$ complexity while Evaluate has $\mathcal{O}(n^2)$ complexity.

## VI. Conclusion

In this paper, we proposed a new secure outsourcing scheme for biometric identification aiming at enhancing the security and privacy for the outsourced biometric database and the query templates. Specifically, our scheme is able to defend against the enrollment attack that makes previous schemes vulnerable. By modelling identification as fixed radius similarity search problem, our scheme exposes less information than previous schemes that based on kNN search problem. In summary, our comprehensive security and complexity analysis show that our scheme is able to enhance the security and privacy of the biometric database and query templates while maintaining the same computational savings from outsourcing.

## References


[1] Marina Blanton and Mehrdad Aliasgari. Secure outsourced computation of iris matching. *Journal of Computer Security*, 20(2-3):259–305, 2012.

[2] Hu Chun, Yousef Elmehdwi, Feng Li, Prabir Bhattacharya, and Wei Jiang. Outsourceable two-party privacy-preserving biometric authentication. In *Proceedings of the 9th ACM symposium on Information, computer and communications security*, pages 401–412. ACM, 2014.

[3] Yousef Elmehdwi, Bharath K Samanthula, and Wei Jiang. Secure k-nearest neighbor query over encrypted data in outsourced environments. In *Data Engineering (ICDE), 2014 IEEE 30th International Conference on*, pages 664–675. IEEE, 2014.

[4] Changhee Hahn and Junbeom Hur. Poster: Towards privacy-preserving biometric identification in cloud computing. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1826–1828. ACM, 2016.

[5] Anil K Jain, Salil Prabhakar, Lin Hong, and Sharath Pankanti. Filterbank-based fingerprint matching. *IEEE transactions on Image Processing*, 9(5):846–859, 2000.

[6] 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 *European Symposium on Research in Computer Security*, pages 186–205. Springer, 2015.

[7] Wai Kit Wong, David Wai-lok Cheung, Ben Kao, and Nikos Mamoulis. Secure knn computation on encrypted databases. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 139–152. ACM, 2009.

[8] Kai Zhou, MH Afifi, and Jian Ren. Expsos: Secure and verifiable outsourcing of exponentiation operations for mobile cloud computing. *IEEE Transactions on Information Forensics and Security*, 12(11):2518–2531, 2017.

[9] Kai Zhou and Jian Ren. Linsos: Secure outsourcing of linear computations based on affine mapping. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–5. IEEE, 2016.

[10] Kai Zhou and Jian Ren. Secure fine-grained access control of mobile user data through untrusted cloud. In *Computer Communication and Networks (ICCCN), 2016 25th International Conference on*, pages 1–9. IEEE, 2016.