TOWARDS BIOMETRIC-BASED AUTHENTICATION FOR CLOUD COMPUTING

Kok-Seng Wong and Myung-Ho Kim
School of Computer Science and Engineering, Soongsil University, Sangdo-Dong Dongjak-Gu, Seoul, South Korea

Keywords: Biometric-based Authentication, Cloud Authentication System, Privacy Preserving Squared Euclidean Distance, Data Protection.

Abstract: Cloud computing is an emerging technology that allows different service providers to offer services in an on-demand environment. Due to the advantages such as flexibility, mobility, and costs saving, the number of cloud user has increased tremendously. Consequently, a more secure and privacy preserving authentication system is becoming important to ensure that only the data owner or the authorized user can gain access and manipulate data stored in the cloud. In the current approach, the service provider authenticates its users based on the credential submitted such as password, token and digital certificate. Unfortunately, these credentials can often be stolen, accidentally revealed or hard to remember. In view of this, we propose a biometric-based authentication protocol, which can be used as the second factor for the cloud users to send their authentication requests. In our solution, the credential submitted by the users consists of the biometric feature vector and the verification code. For the user to successful authenticate, both the biometric feature vector and the verification code must be combined, transformed, and shuffled correctly. Our proposed solution not only provides the security mechanism for the authentication process, but also supports the privacy protection for all sensitive information of the user.

1 INTRODUCTION

Cloud computing is an emerging technology which allows multi-tenant to request for services and resources from their service providers in an on-demand environment. It is a complex yet resource saving infrastructure for today's modern business needs, providing the means through which services are delivered to the end users via Internet access. In the cloud environment, users can access services based on their needs without knowing how the services are delivered and where the service are hosted.

The US National Institute of Standards and Technology (NIST) has defined cloud computing as follows (Mell and Grance, 2009): Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Hardware devices, software, storage and network infrastructure are made available to user through Internet access. Rather than purchasing expensive but powerful resources, users lease these resources from the service providers. With cloud computing, user can access the services via Internet access regardless of time and location. They also get rid of software installation in their local machine and able to enjoy high availability of services. Furthermore, high efficiency and fast deployment benefits are also the attractions for company and individual who moves to cloud services.

Due to the advantages such as flexibility, mobility, and costs saving, the number of cloud user has increased tremendously. Industry analysts have made projections that entire computing industry will be transformed into Cloud environment (Buyya et al., 2009).

In this Cloud-driven era, security and privacy concerns are becoming growing problems for the user and the service provider. User authentication is often the key issue in the Cloud environment. It is an important operation for the service provider to verify who can access their services and to identify the group of each user.

Some commonly used authentication services

Wong K. and Kim M..
TOWARDS BIOMETRIC-BASED AUTHENTICATION FOR CLOUD COMPUTING.
DOI: 10.5220/0003909305010510
In *Proceedings of the 2nd International Conference on Cloud Computing and Services Science* (CLOSER-2012), pages 501-510
ISBN: 978-989-8565-05-1

include Kerberos (Neuman and Ts'o, 1994) and OpenID (Recordon and Reed, 2006). The service provider authenticates its users based on the credential submitted such as password, token and digital certificate. Unfortunately, these credentials can often be stolen, accidentally revealed or hard to remember. In view of this, we propose a biometric-based authentication protocol that can be used as the second factor for the cloud users to send their authentication requests. Biometric authentication can improve the quality of authentication (QoA) in cloud environment. Our solution ensures both security in the authentication and the privacy protection for all sensitive information.

1.1 Organization

The rest of this paper is organized as follows: The background for this research is in Section 2 and the technical preliminaries are described in Section 3. We present our proposed solution in Section 4 followed by the analysis in Section 5. Our conclusion is in Section 6.

2 BACKGROUND

2.1 Cloud Computing Models

Cloud services are delivered in three fundamental models (Lenk et al., 2009): Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). IaaS is the lowest level which is closest to the hardware devices whereas, SaaS is the highest level that provides services to the end-users. The Amazon web service is one type of IaaS which has been widely used since 2006 while the Salesforce.com CRM system is an example of SaaS.

PaaS level provides an application platform in the cloud. Windows Azure platform is one example of PaaS which enable the developers to build, host and scale their applications in the Microsoft data centers. Recently, a new concept called “Everything as a Service (XaaS)” has been adopted as the new trend in cloud computing. Several vendors such as Microsoft and Hewlett Packard (Fiveash, 2008) have been associated with it.

Biometric Authentication as a Service (BioAaaS) has been defined as an approach for strong authentication in web environments based on the SaaS model (Senk and Dotzler, 2011).

2.2 User Authentication

When performing authentication over the Internet, credential will be submitted by the principal (the user, machine, or service requesting access) (Convery, 2007). If the credentials match, the user is allowed to access the services it subscribed from the service providers. In this paper, we only consider user as the principal who submits its credential for authentication over the cloud.

There are several types of credential the users can submit as proof of their identity. Shared-key is typically password used protocols such as Password Authentication Protocol (PAP) (Lloyd and Simpson, 1992) and Challenge Handshake Authentication Protocol (CHAP) (Simpson, 1996).

Digital certificate is second type of credential which can provide strong authentication in the cloud environment. It is an electronic document which uses a trusted Certificate Authority (CA) to blind the encryption key with an identity (Canetti, 2004). Decryption key is the only way to validate the signed certificate.

Another type of credential is the commonly used one-time-password (OTP) (Haller, 1994, Rubin, 1995). The end-user obtains the OTP from the token (hardware or software) during the login time. The token can generate a randomized password string based on a complex algorithm in real time. Since the password generated is unique and can only be used once, OTP is possible to be used in the Cloud environment. For example, Amazon Web Services (AMW) has already started to use its OTP token for use with individual AWS accounts (Brooks, 2009).

Recently, a German company BioID proposes the world’s first biometric authentication service for cloud computing (Krowneva, 2011). In their solution, biometric authentication as a service (BaaS) has been proposed to provide single sign-on for user authentication.

2.3 Biometric-based Authentication

Biometric characteristics such as iris patterns, face, fingerprints, palm prints and voice will be submitted by the user as the credential for authentication over the cloud. Biometric-based authentication systems provide a higher degree of security as compared with conventional authentication systems. Furthermore, it allows the system to keep track of the user’s activities because individual biometric characteristics cannot be shared with others.

Generally, biometric authentication systems consist of five modules, namely, the biometric sensor, feature extractor, template storage, matching module, and the decision module. Figure 1 illustrates the general design for the biometric-based authentication systems.

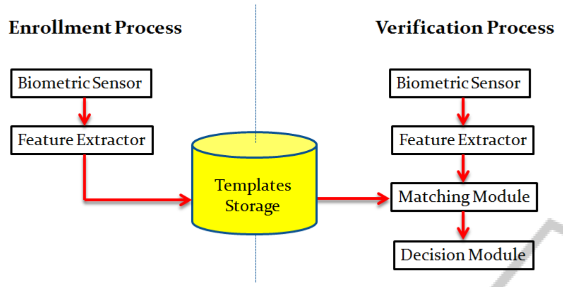

Figure 1: General design for biometric-based authentication systems.

During the enrolment process, the biometric sensor scans the biometric traits of the user while the feature extractor extracts the feature vector from the scanned biometric data. The feature vector is then stored in the template storage.

At the verification stage, the biometric sensor and the feature extractor perform the same tasks as in the enrolment process. However, the extracted feature vector (query feature vector) will not be stored in the storage. Instead, it will be used by the matching module to compare with the templates stored in the storage and output a similarity score.

The decision module is responsible in making the final decision (accept or reject), which depends on the similarity score or the threshold determined by the system administrator.

3 TECHNICAL PRELIMINARIES

In this section, we describe some technical preliminaries for our protocol design.

3.1 Security and Privacy Definitions

Security definition: In a generic sense, security is the prevention of unauthorized party from gaining access to confidential information and system resources. A secure authentication system needs to ensure that users are the persons they claim to be.

Our protocol is secure if no adversary party (the malicious client, the malicious service provider or the network intruder) gains access to the sensitive information such as the verification code and the template for each user, the shuffle protocol, all decryption keys, and the original feature vector of the user. During the authentication process, the protocol must prevent the malicious service provider from reconstructing the original feature vector based on the verification code and the template stored in the cloud. Also, the network intruder who watches the traffic on the network must not learn anything.

Privacy definition: Information or data privacy is referring to the ability of an individual or system to prevent the leakage of any sensitive information to any unauthorized party. A privacy-preserved system should ensure that unauthorized party does not improperly access confidential information.

In this paper, our solution bases on the user's biometric feature vector. Hence, we only consider the privacy issues on the biometric template and the verification code protections. No intermediate result should leak any sensitive information and the service provider cannot distinguish whether two authentication requests belong to the same user.

3.2 Homomorphic Encryption Scheme

We will use the additive property of the homomorphic encryption scheme proposed by Paillier (Paillier, 1999) in our protocol.

Let $E_a(m_1)$ denote the encryption of message m_1 with encryption key, E_a. The scheme supports the following operations in an encrypted form:

- *Addition*: Given two ciphertexts $E_a(m_1)$ and $E_a(m_2)$, there exists an efficient algorithm $+_h$ to compute $E_a(m_1 + m_2)$.
- *Scalar multiplication*: Given a constant c and a ciphertext $E_a(m_1)$, there exists an efficient algorithm $\cdot_h$ to compute $E_a(c \cdot m_1)$.

Note that when a scheme supports the additive operation, it also supports scalar multiplication because $E_a(c \cdot m_1)$ can be achieved by summing $E_a(m_1)$ successively c times.

By using the homomorphic encryption scheme, we can compute the additive operation directly on the encrypted data without the decryption. This is a useful feature because the biometric template stored in the server does not require decryption during the matching operation. In our solution, the encryption keys used by both parties are symmetric-key.

3.3 Notations Used

In Table 1, we summarize all the notations used hereafter in this paper.

Table 1: Common notations used.

Notation	Definition
X	original feature vector extracted from the user during the enrolment process
Y	original feature vector extracted from the user during the verification process
X'	transformed vector during the enrolment process
Y'	transformed vector during the verification process
X''	shuffled vector during the enrolment process
Y''	shuffled vector during the verification process
π_u	shuffle protocol for the user U
$x_i^{'}$	i -th element of X'
$y_i^{'}$	i -th element of Y'
s	squared Euclidean distance
n	length of the original feature vector
m	length of the verification code
k	length of the transformed vector where, $k = n + m + 4$
TID	template identification number
VID	verification code identification number
E_u	encryption key from the user U
D_u	decryption key from the user U
E_p	encryption key from the service provider
D_p	decryption key from the service provider
$E_{pk}(\cdot)$	encryption operation by using the E_{pk}
$D_{pk}(\cdot)$	decryption operation by using the D_{pk}
ω	random non-zero number

Note that all datasets (both the original and the transformed feature vectors) are ordered set.

4 PROPOSED SOLUTION

In our solution, the credential submitted by the user consists of two parts: user's biometric feature vector and the verification code. Both parts must be combined, transformed, and shuffled correctly in order for the user to successful authenticate.

Like most existing biometric-based authentication systems, our solution consists of both the enrolment and the verification processes. In the following sections, we will describe in details the components and the authentication workflows of our solution.

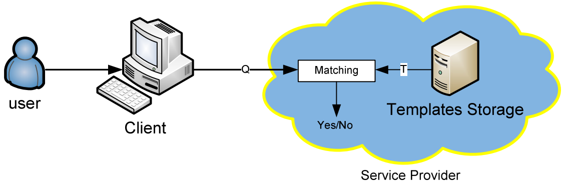

Figure 2: Basic components for the proposed solution.

4.1 Components

We now formally describe the components in our proposed solution as follow: (as illustrated in Figure. 2):

- *User*: individual who sends the authentication request.
- *Client*: computer or workstation with Internet access.
- *Service provider*: company or organization who provides cloud services (SaaS, PaaS or IaaS) to the user.
- *Template (T)*: transformed feature vector stored in the cloud storage.
- *Query feature vector (Q)*: transformed feature vector used to compare with the template.

Hereafter in this section, we refer sensitive information as our protected information includes the biometric template, the verification code, and the shuffle protocol.

Both the client and the service provider manage separate components in order to verify the user. The client has the following components:

- *Biometric sensor*: scans the biometric traits of the user.
- *Feature extractor*: extracts the feature vector from the scanned biometric data.
- *Verification code generator*: generates unique verification code for the user.
- *Transformation module*: transforms the original feature vector and shuffles the transformed feature vector.
- *Encryption module*: encrypts the transformed and shuffled feature vector with the correct encryption key (i.e., encrypts with the user's key during the enrolment process).
- *Decryption module*: decrypts the computation output.

The service provider requires the following components:

- *Verification code retrieval*: retrieves the verification code for the user.
- *Templates storage*: stores the template for each user.

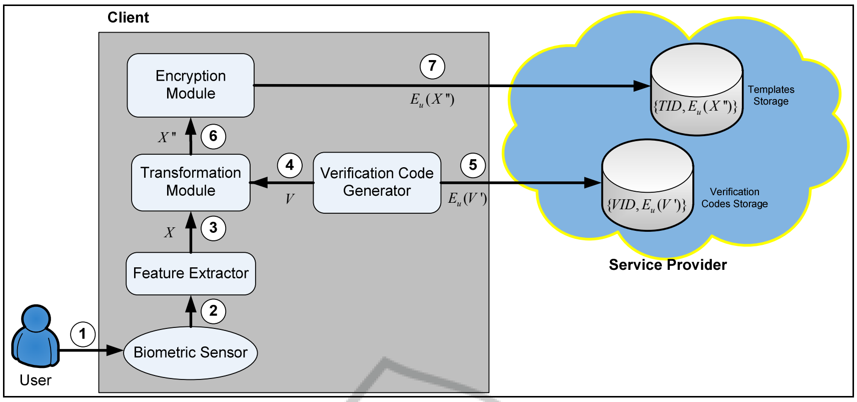

Figure 3: The overview of the enrolment process.

- *Verification codes storage*: stores the verification code for each user.
- *Computation module*: performs the squared Euclidean distance (s) computation between the query feature vector and the template.
- *Decision module*: making the final decision by comparing the s with the given threshold τ .

4.2 Enrolment

The objective of the enrolment process is to process the scanned biometric data and extract a set of feature vector to be stored as the template for the user. The enrolment process is required for the new user who wants to join the cloud. A successful enrolment process enables the user to receive the TID and the VID .

4.2.1 Transformation

Let $X=\{x_1,x_2,...,x_n\}$, $n>0$ and $V=\{v_1,v_2,...,v_m\}$, $m>0$ be the feature vector of the user and the verification code generated, respectively. We transform X into $X'=\{x_i' \mid i=1,2,...,n+m+4\}$ such that $x_i'=x_i$ for $1\le i\le n$, $x_{n+j}'=v_j$ for $1\le j\le m$, $x_{n+m+1}'=x_{n+m+2}'=1$, $x_{n+m+3}'=\sum_{i=1}^{n}x_i^2$ and $x_{n+m+4}'=\sum_{j=1}^{m}v_j^2$.

4.2.2 Shuffle Protocol

We require a shuffle protocol (π_u) to permute the order of elements in the transformed vector X' . We use the same shuffle protocol during the verification process for the same user.

4.2.3 Overview of the Enrolment Process

We illustrate the overview of the enrolment process in Figure. 3 and the workflow as follow:

1. The biometric sensor scans the biometric trait of the user.
2. The feature extractor processes the scanned biometric data to extract the feature vector of the user, $X=\{x_1,x_2,...,x_n\}$.
3. The feature extractor sends X to the transformation module.
4. The verification code generator generates a unique verification code $V=\{v_1,v_2,...,v_m\}$ for the user and sends it to the transformation module.
5. We transform the verification code into $V'=-2V$ and encrypt it using the encryption key from the user. Then, it will be stored at the verification codes storage.
6. The transformation module transforms X into X' . Next, it shuffles the transformed vector X' i.e. $X''=\pi_u(X')$ before sending it to the encryption module.
7. The encryption module encrypts X'' with the user's encryption key. Finally, the $E_u(X'')$ is sent to the service provider and stored as the user's template in the templates storage.

4.3 Verification

When the user wants to access data stored in the cloud storages or uses the cloud services, the user must be authenticated first. The verification process is responsible to verify the users who they claim to be.

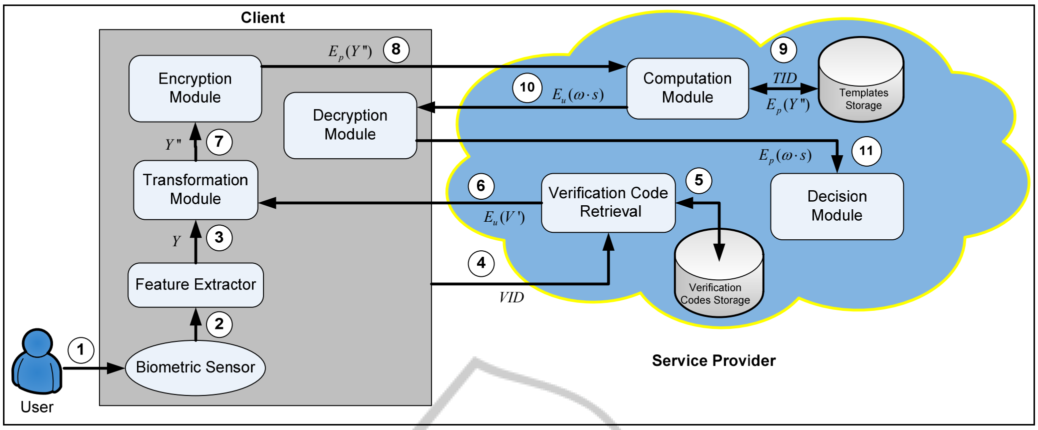

Figure 4: The overview of the verification process.

4.3.1 Transformation

Let $Y = \{y_1, y_2, ..., y_n\}$, $n > 0$ and $V = \{v_1, v_2, ..., v_m\}$, $m > 0$ be the feature vector extracted from the user and the verification code, respectively. The verification code used must be the same in both enrolment and verification processes. We transform Y into $Y' = \{y_i' \mid i = 1, 2, ..., n+m+4\}$ such that $y_i' = -2y_i$ for $1 \le i \le n$, $y_{n+j}' = -2v_j$ for $1 \le j \le m$, $y_{n+m+1}' = \sum_{i=1}^{n} y_i^2$, $y_{n+m+2}' = \sum_{j=1}^{m} v_j^2$, $y_{n+m+3}' = y_{n+m+4}' = 1$. The length for Y' must be same as X' which is $k = n+m+4$.

4.3.2 Shuffle Protocol

We require the same shuffle protocol used in the enrolment process during the verification process. The transformed feature vector Y' needs to be shuffled in the same order as X' .

4.3.3 Overview of the Verification Process

The workflow for the verification process is as follow (as illustrated in Figure. 4):

1. The biometric sensor scans the biometric trait of the user.
2. The feature extractor processes the scanned biometric data to extract the feature vector of the user, $Y = \{y_1, y_2, ..., y_n\}$.
3. The feature extractor sends Y to the transformation module.
4. Next, the client retrieves the verification code by using the VID .
5. The verification code retrieval retrieves the verification code for the user from the storage which is associated with the VID .
6. The transformation module receives the verification code $E_u(V')$.
7. The transformation module computes $D_u\left(E_u(V')\right)$ and transforms Y into vector Y' . Next, it shuffles Y' i.e. $Y" = \pi_u(Y')$ and sends $Y"$ to the encryption module.
8. The encryption module encrypts $Y"$ with the service provider's encryption key E_p . Next, the $E_p(Y")$ is sent together with the TID to the computation module.
9. The computation module retrieves $E_u(X")$ from the templates storage which is associated with the TID .
10. If both $E_u(X")$ and $E_p(Y")$ have the same size, the computation module computes:
    i. Decryption: $D_p\left(E_p(Y")\right) = Y"$
    ii. Scalar multiplication: $Y" \cdot E_u(X") = E_u(X" \cdot Y")$
    iii. Homomorphic additive operation: $E_u(s) = E_u\left(\sum_{i=1}^{n+m+4}\left(x_i" \cdot y_i"\right)\right)$
    iv. Add noise: $\omega \cdot E_u(s) = E_u(\omega \cdot s)$, where ω is a random non-zero number.

    The computation module sends $E_u(\omega \cdot s)$ to the client.
11. The decryption module of the client decrypts $E_u(\omega \cdot s)$ and then encrypts $\omega \cdot s$ with E_p . Then, the decryption module sends $E_p(\omega \cdot s)$ to the decision module of the service provider for

making the decision. The decision module decrypts $E_p(\omega \cdot s)$ and makes the decision as follows (τ is the threshold determined by the service provider):

$$decision = \begin{cases} Accept, if\ s < \tau \\ Reject, if\ \ s > \tau \end{cases}$$

Note that for different authentication requests, we may require different security levels. Hence, our system can assign different threshold values for different users.

5 ANALYSIS

In this section, we present the correctness, security, privacy and efficiency analysis for our proposed solution.

5.1 Correctness Analysis

Theorem 1. Our protocol correctly computes the squared Euclidean distance between the query feature vector and the template stored in the storage if both the client and the service provider follow the protocol faithfully.

Proof. Let $X = \{x_1, x_2, ..., x_n\}$ be the extracted feature vector of user A during the enrolment process. We transform the feature vector X into X':

$$X' = \begin{Bmatrix} x_1, ..., x_n, v_1, ..., v_m, 1, 1, \\ \left(\sum_{i=1}^{n} x_i^2\right), \left(\sum_{j=1}^{m} v_j^2\right) \end{Bmatrix} \tag{1}$$

Then, we randomly shuffle the order of elements in X'. Let $X'' = \pi_A(X')$ be the shuffled vector by using the shuffle protocol π_A. Next, we encrypt X'' by using the encryption key E_A and store the following result as the template in the templates storage:

$$E_A(X'') = \begin{Bmatrix} E_A(x_1), ..., E_A(x_n), \\ E_A(v_1), ..., E_A(v_m), \\ E_A(1), E_A(1), \\ E_A\left(\sum_{i=1}^{n} x_i^2\right), E_A\left(\sum_{i=1}^{m} v_j^2\right) \end{Bmatrix} \tag{2}$$

Note that for ease of explanation, we do not change the order of elements in Eq. (2).

Assume that $Y = \{y_1, y_2, ..., y_n\}$ is the query feature vector during the verification process. The client retrieves the verification code from the service provider and transforms Y into Y' as follows:

$$Y' = \begin{Bmatrix} -2y_1, ..., -2y_n, -2v_1, ..., -2v_m, \\ \left(\sum_{i=1}^{n} y_i^2\right), \left(\sum_{j=1}^{m} v_j^2\right), 1, 1 \end{Bmatrix} \tag{3}$$

By using the same shuffle protocol π_A (if the user is A), the client computes $Y'' = \pi_A(Y')$ and encrypts Y'' with the encryption key E_P to produce:

$$E_P(Y'') = \begin{Bmatrix} E_P(-2y_1), ..., E_P(-2y_n), \\ E_P(-2v_1), ..., E_P(-2v_m), \\ E_P\left(\sum_{i=1}^{n} x_i^2\right), E_P\left(\sum_{i=1}^{m} v_j^2\right), \\ E_P(1), E_P(1) \end{Bmatrix} \tag{4}$$

For ease of explanation, we do not change the order of elements in Eq. (4).

The squared Euclidean distance is computed as follow: The service provider first decrypts $E_P(Y'')$ to obtain Y'' and computes the scalar multiplication for each i-th element in Y'' and $E_A(X'')$ according to their index position:

$$\begin{aligned} & Y'' \cdot E_A(X'') \\ & = E_A(X'' \cdot Y'') \\ & = \begin{Bmatrix} \left(-2y_1 \cdot E_A(x_1)\right), ..., \left(-2y_n \cdot E_A(x_n)\right), \\ \left(-2v_1 \cdot E_A(v_1)\right), ..., \left(-2v_m \cdot E_A(v_m)\right), \\ \left(\sum_{i=1}^{n} y_i^2 \cdot E_A(1)\right), \left(\sum_{j=1}^{m} v_j^2 \cdot E_A(1)\right), \\ \left(1 \cdot E_A\left(\sum_{i=1}^{n} x_i^2\right)\right), \left(1 \cdot E_A\left(\sum_{i=1}^{m} v_j^2\right)\right) \end{Bmatrix} \\ & = \begin{Bmatrix} E_A(-2x_1y_1), ..., E_A(-2x_ny_n), \\ E_A(-2v_1^2), ..., E_A(-2v_m^2), \\ E_A\left(\sum_{i=1}^{n} y_i^2\right), E_A\left(\sum_{j=1}^{m} v_j^2\right), \\ E_A\left(\sum_{i=1}^{n} x_i^2\right), E_A\left(\sum_{i=1}^{m} v_j^2\right) \end{Bmatrix} \end{aligned} \tag{5}$$

Next, the service provider computes homomorphic additive operation for each $\left(x_i^{"} \cdot y_i^{"}\right) \in \left(X'' \cdot Y''\right)$ in Eq. (5):

$$\begin{aligned} E_A(s) = & E_A\left(\sum_{i=1}^{n} -2x_i y_i\right) +_h E_A\left(\sum_{j=1}^{m} -2v_j^2\right) \\ & +_h E_A\left(\sum_{i=1}^{n} y_i^2\right) +_h E_A\left(\sum_{j=1}^{m} v_j^2\right) \\ & +_h E_A\left(\sum_{i=1}^{n} x_i^2\right) +_h E_A\left(\sum_{i=1}^{m} v_j^2\right) \\ = & E_A\left(\sum_{i=1}^{n} x_i^2\right) +_h E_A\left(\sum_{i=1}^{n} -2x_i y_i\right) \\ & +_h E_A\left(\sum_{i=1}^{n} y_i^2\right) \\ = & E_A\left(\sum_{i=1}^{n}\left(x_i^2 - 2x_i y_i + y_i^2\right)\right) \\ = & E_A\left(\sum_{i=1}^{n}\left(x_i - y_i\right)^2\right) \end{aligned} \tag{6}$$

After we decipher the result in Eq. (6), we can obtain the squared Euclidean distance $s = \sum_{i=1}^{n}(x_i - y_i)^2$. Note that in Eq. (6), we eliminate the verification code and all additional features. Hence, if the service provider retrieves the correct verification code and the client computes Y'' correctly, our protocol outputs the correct squared Euclidean distance for X and Y .

Theorem 2. If one of the parties (either the client or the service provider) is not following the protocol, the final output will not reflect the squared Euclidean distance for the two vectors (X and Y). Subsequently, the verification process will fail and the user cannot access the system.

Proof. The client or the service provider who is not following the protocol is considering as the malicious party in our protocol. The proof of this theorem is same as the proof in Theorem 3 and Theorem 4 under the security analysis.

5.2 Security Analysis

In this section, we will analyze two possible attacks: internal and external attack. Internal attack involves malicious party such as employee at client who attempts to gain access into the cloud. External attack involves external parties (intruders or network attackers) who watch the traffic on the network. They are interested in learning some knowledge from the computation protocol or intercept the data in the network. Note that internal attack is more serious as compared to the external attack because attackers are having more knowledge about the protocol.

Theorem 3. Our protocol is secure against malicious user who tries to gain access to the cloud. Without the knowledge of sensitive information and the decryption key, the authentication is not possible for attacker at the client side.

Proof. During the enrolment process, the system generates the biometric template for each user. Only the user who enrolled into the cloud has its template and the verification code stored in the cloud storages. In the absence of the template, the system cannot authenticate the user.

In our protocol, any malicious user who wants to pose as an enrolled user must gain access to three sensitive information: (1) the verification code, (2) the original feature vector and (3) the shuffle protocol. Since the verification code is stored at the cloud storage and is encrypted using the encryption key of the respective user, the attacker is not able to view it because he has no knowledge about the decryption key. If the attacker gains access to the original feature vector of the user, he is not able to use it directly for the verification process because the verification code and the shuffle protocol are not accessible. In the worst scenario, if the attacker obtains the decryption key of any user, the security for the user is still can be guaranteed. Hence, our protocol is secure against attacker who tries to gain access to the cloud system.

Theorem 4. Our protocol is secure against malicious service provider who tries to gain access to the verification codes and templates stored in the cloud storages. The malicious service provider is not able to reconstruct the original feature vector of any user.

Proof. A malicious service provider wants to learn the original feature vector of the user. Although the verification code and the template are reside in the service provider's side, but the malicious service provider is not able to reconstruct the original feature vector for any user. This is because the verification code and the template are encrypted using the encryption key from each respective user. The service provider has no knowledge about the decryption key of any user. Gaining access to these encrypted vector is as difficult as attacking the encryption algorithm. Brute-force attack is also impossible since all the verification codes and the templates are different (after the encryption operation). Hence, our protocol is able to prevent the malicious service provider from reconstruct the original feature vector of the user.

Theorem 5. Network attacker who listens to the traffic is not able to learn any sensitive information.

Proof. In our protocol, all the data transmit over the network (between the client and the service provider) are encrypt either with the user's encryption key or with the service provider's key. When the network attacker watches the network, he cannot learn any information because he has no knowledge about the decryption key. During the verification process, network attacker is not possible to be authenticated by the cloud because he has no knowledge about any sensitive information. Hence, our protocol is secure against the network attacker.

5.3 Privacy Analysis

The main privacy concern in our protocol is the amount of information revealed to the service provider during the authentication process. Our protocol should ensure the confidentiality of all sensitive information. All the intermediate results and the final output will not compromise the privacy of the user.

Theorem 6. The feature vector extracted from the user is never stored in their original form. Although the template and the verification code are stored with the service provider, but the service provider learns nothing from its storage.

Proof. In our protocol, we encrypt both the verification code and the template using the encryption key from the user and then store the encrypted data at the service provider's side. The service provider is not able to learn anything because it has no knowledge about the decryption key from the user. In the worst scenario, if the decryption key of the user has been compromised, the service provider also not able to identify the original feature vector of the user because the template has been transformed and shuffled during the enrolment process. Furthermore, we include some additional features and the verification code into the template.

During the verification process, the service provider decrypts the $E_P(Y'')$ before performing the scalar multiplication operation. After the decryption, the service provider is not able to identify the original feature vector and verification code used in the query feature vector. Hence, our protocol protects both the verification code and the template stored in the cloud storages.

Theorem 7. The service provider is not able to distinguish whether two authentication requests belong to the same user.

Proof. In our protocol, the verification code and the template are stored separately in the service provider’s side. This design prevents the malicious party from knowing which verification code is associated with which template in the case when both storages are compromised. The decision module makes the verification decision based on the similarity score (squared Euclidean distance) and the threshold determined by the system. If the similarity score is lower than the threshold, it can reject the user. Otherwise, the system verifies the user and the authentication process is successful. With only the similarity score, the decision module is not able to distinguish whether two authentication requests belong to the same user.

5.4 Efficiency Analysis

The total communication cost depends on the amount of data transferred in the protocol. During the verification process, the client sends one query feature vector with $k = n + m + 4$ encrypted data to the service provider. The service provider responds with one encrypted result. The client then replies with one encrypted value. The communication complexity incurred by our protocol is $O(k)$.

In terms of complexity, our protocol requires $O(k)$ encryptions, $O(k)$ scalar multiplications and $O(k)$ homomorphic additive operations.

6 CONCLUSIONS

In this paper, we proposed a biometric-based authentication protocol for cloud computing. Our target is to achieve secure authentication while protecting the sensitive information of users. We incorporate the homomorphic encryption scheme in our Squared Euclidean distance computation that allows us to compare both the query feature vector and the template in an encrypted form. In order for the user to successful authenticate the biometric feature vector and the verification code must be correctly transformed and shuffled. Our solution preserves the privacy for sensitive information of users and securely performs the authentication process.

REFERENCES

Brooks, C. 2009. Amazon adds onetime password token to entice the wary. *SearchCloudComputing.*

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J. & Brandic, I. 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Gener. Comput. Syst.,* 25**,** 599-616.

Canetti, R. 2004. Universally Composable Signature, Certification, and Authentication. *Proceedings of the 17th IEEE workshop on Computer Security Foundations.* IEEE Computer Society.

Convery, S. 2007. Network Authentication, Authorization, and Accounting Part One: Concepts, Elements, and Approaches. *The Internet Protocol Journal,* 10**,** 2-11.

Fiveash, K. 2008. HP sells cloud vision amidst economic downpour. *Will customers get soaked on transformation journeys?* : King's College London.

Haller, N. 1994. The S/KEY One-Time Password System. *Internet Society Symposium on Network and Distributed Systems.*

Krowneva. 2011. *BioID Announces World's First Biometric Authentication as a Service (BaaS)* [Online]. Available: http://silicontrust.wordpress.com/

2011/03/04/bioid-announces-worlds-first-biometric-authentication-as-a-service-baas/ [Accessed].

Lenk, A., Klems, M., Nimis, J., Tai, S. & Sandholm, T. 2009. What's inside the Cloud? An architectural map of the Cloud landscape. *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing.* IEEE Computer Society.

Lloyd, B. & Simpson, W. 1992. PPP Authentication Protocols. *RFC Editor*.

Mell, P. & Grance, T. 2009. The NIST Definition of Cloud Computing. Available: http://www.csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc.

Neuman, B. C. & Ts'o, T. 1994. Kerberos: An Authentication Service for Open Network Systems. *IEEE Communications,* 32**,** 33-38.

Paillier, P. 1999. Public-key cryptosystems based on composite degree residuosity classes. *Proceedings of the 17th international conference on Theory and application of cryptographic techniques.* Prague, Czech Republic: Springer-Verlag.

Recordon, D. & Reed, D. 2006. OpenID 2.0: a platform for user-centric identity management. *Proceedings of the second ACM workshop on Digital identity management*. Alexandria, Virginia, USA: ACM.

Rubin, A. D. 1995. Independent one-time passwords. *Proceedings of the 5th conference on USENIX UNIX Security Symposium - Volume 5.* Salt Lake City, Utah: USENIX Association.

Senk, C. & Dotzler, F. 2011. Biometric Authentication as a Service for Enterprise Identity Management Deployment: A Data Protection Perspective. *Sixth International Conference on Availability, Reliability and Security.* Vienna Austria.

Simpson, W. 1996. PPP Challenge Handshake Authentication Protocol (CHAP). *RFC Editor.*