International Conference on Latest Advancements & Future Trends In Engineering, Science & Management
Sambhram Institute of Technology, Bangalore (India)
03rd - 04th May 2019 www.conferenceworld.in
SAMBHRAM INSTITUTE OF TECHNOLOGY
ISBN : 978-93-87793-86-6

An Effective and Secrecy Conserving Biometric Verification Scheme in Cloud with Blockchain

Sangeetha S L[1], Bhoomika B M[2], L Priyanka[3], Suma K C[4]

(Dept. of Information Science and Engineering, Sambhram Institute of Technology, India)

Abstract

Biometric verification is taking over traditional passwords and ID card based authentication worldwide. Evolution of cloud computing has made database owners to offshore the massive size of biometric data and authentication tasks to cloud which, however has IT security issues that is more profound and hence making data is vulnerable to thefts. In this paper we propose an effective secrecy conserving biometric verification offshoring scheme, where database owner receives the query data, encrypts and transmits data to cloud, here data in the cloud are stored in the form of Blockchain. Cloud server identifies the best match for the encrypted query and returns related index to the database owner. A complete security analysis specifies that the proposed scheme is secure, even if intruders try to forge authentication requests and collaborate with the cloud. Compared to previous protocols, experimental effects manifest that the proposed scheme attains better performance in both composition and authentication procedures.

Keywords- Biometric verification, Blockchain, cloud computing,data offshoring, privacy-preserving.

1.Introduction

Biometric Verification is a security process, by which a person can be uniquely identified by evaluating one or more distinguishing biological traits, which is a promising way to identify the valid users. Biometric Verification is considered to be more consistent and convenient [1] when compared with old-fashioned authentication techniques built on passwords and Identification cards. Moreover, biometric Verification has been broadly used in various fields by using biometric traits such as fingerprint[2], irises[3] and facial patterns[4], which may be collected from numerous sensors[5]-[9].

To get rid of the expensive storage and computation charges in a biometric verification system, the database owner such as FBI who is accountable to manage the national fingerprint data, might want to offshore huge biometric information to the cloud server (e.g., Amazon). But, the biometric data has to be encrypted prior to offshoring to maintain the privacy of biometric records. Every time, when a FBI's partner (e.g., the police station) needs to verify a person's identity, he goes to the FBI and creates a verification request by means of the individual's biometric traits (e.g., fingerprints, irises, voice patterns, facial pattern etc.). Later, the query is encrypted and submitted to cloud by the FBI to obtain the nearest match. Therefore, in what way to outline a

protocol that allow efficient and privacy-preserving biometric verification in the cloud computing is a challenging issue.

Various type of privacy-preserving biometric verification[10]-[17] solutions have been introduced. But, most of them primarily focus on privacy preservation and disregard the efficiency, like the schemes based on homomorphic encryption and unaware transfer in[10] and [11] for face image and fingerprint identification correspondingly. Drawback, due to the performance issue of confined devices, these schemes are not effective once the volume of the database is more than 10MB. Later, Evans et al. [12] proposed a biometric verification scheme by making use circuit design and ciphertext packing techniques to attain efficient verification for a massive database up to 1GB. In addition, Yuan and Yu[13] introduced an efficient privacy-preserving biometric verification scheme. Particularly, they built 3 units and outlined a concrete protocol to attain the safety of fingerprint trait. To increase the efficiency, in their scheme, the database owner offshores verification matching responsibilities to the cloud. But, Zhu et al.[18] discovered that Yuan and Yu's protocol can be cracked by a collusion attack thrown by a malicious user and cloud.

A blockchain is a extending list of records, called blocks, which are linked using cryptography. As shown in fig.1, each block contains a cryptographic hash of the previous block, a timestamp, data i.e., both encrypted index and the extracted features from fingerprint (generally represented as a Merkle tree root hash).

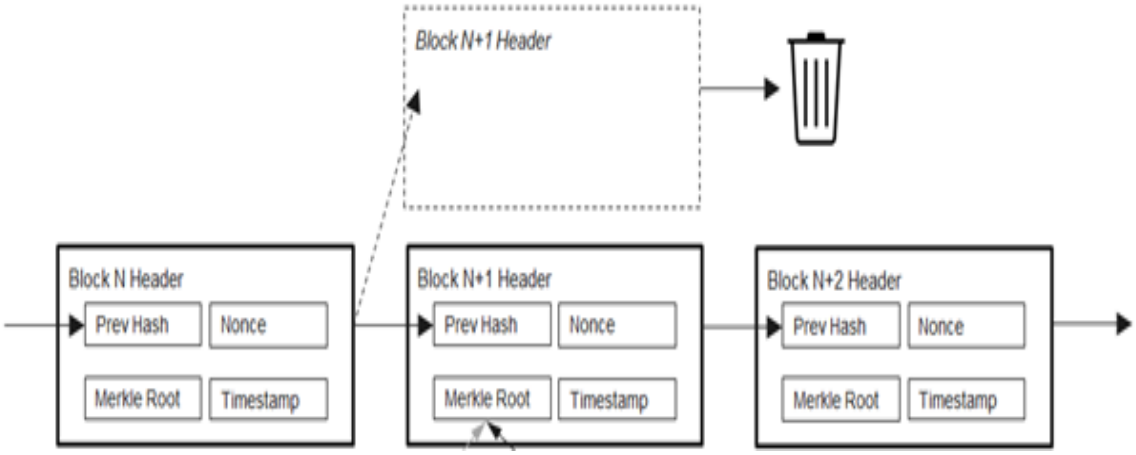


Figure 1.The structural view of Blockchain.

In this paper, we present an effective and secrecy conserving biometric verification with blockchain scheme which can endure the collusion attack launched by the malicious users and the cloud. Particularly, our primary contributions can be briefed as follows:

- We study the biometric verification scheme [13] and show its insufficiencies and security weakness under the presented level-3 attack.
- We propose a novel effective and privacy-preserving biometric verification with blockchain scheme. The thorough security analysis manifests that the presented scheme could attain a required level of privacy protection. Particularly, the proposed scheme is secure under the biometric verification offshoring model and can also endure the attack presented by [18]

- Comparing with the existing biometric verification schemes, the performance analysis represents that the presented scheme provides a minimum computational cost in both preparation and verification procedures.

The remaining part of this paper is systemized as follows: section II manifests the units and design goals. In section III, we propose an effective and secrecy-conserving biometric verification with blockchain scheme. In section IV, Security analysis as represented. Performance analysis in section V, followed by related work in section VI and we present our conclusion and future work in section VII.

2. Models and Design Goals

This Section proposes the system unit, attack unit, designer goals and the notations employed in the upcoming sections.

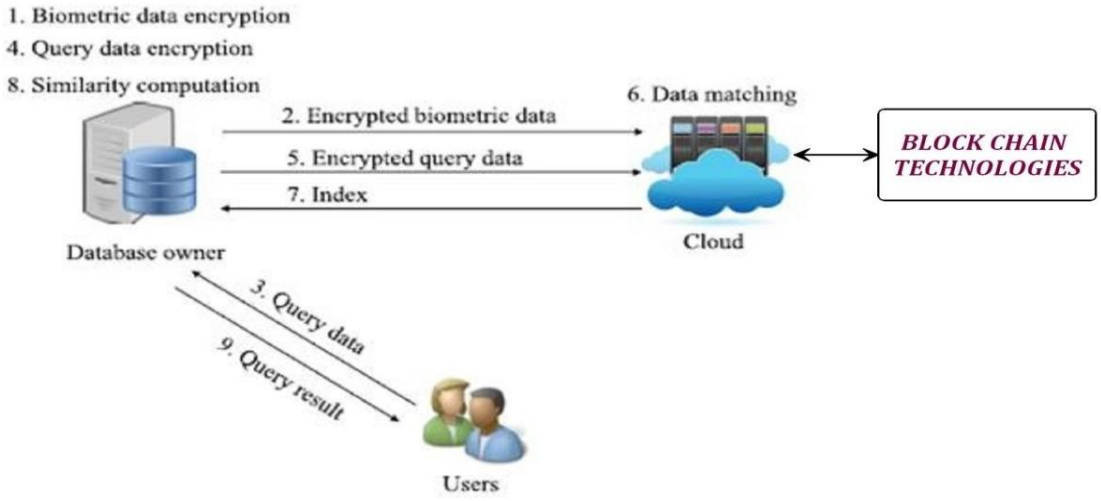


Figure 2. System model.

2.1 System Model

As manifested in Fig.2, three types of units are involved in the system which includes the database owner, users and the cloud. The massive size of biometric data (i.e., fingerprints, irises, voice and facial patterns etc.), is held by database owner, which in turn is encrypted and transferred to the cloud and is stored in the form of blockchain. When a user needs to verify himself/herself, a query request is sent to the database owner. The database owner receives the query request and creates the ciphertext for the biometric trait and then transmits the ciphertext to the cloud for verification. The cloud server identifies the foremost match for the encrypted query request and returns the corresponding index to the database owner. At last, the database owner determines the resemblance between the requested query data and the biometric data related with the index, and returns the query outcome to the user.

In our proposed scheme, we assume that the biometric data are processed so that its depiction can be utilized to execute the biometric match.Without losing generality, likely to [17] and [18], we earmark fingerprints and utilize FingerCodes [19] to depict the fingerprints. Particularly, a FingerCode comprise of n elements and every element is a l-bit integer (typically n=640 and l=8). x = [x1, x2, … , xn] and y = [y1, y2, … , yn] are the two

given FingerCodes. They are considered to be a good match, if their Euclidean distance is below a threshold Є, i.e., the two fingerprints are assessed from the same person.

2.2 Attack Model

As described in [13]-[15] and [17], the cloud server is considered to be "honest but peculiar". The cloud sternly follows the designed protocol, however reveals privacy from both the user and database owner. Assuming that an intruder can notice all the data stored in the cloud i.e., the encrypted queries, encrypted biometric database and matching results. Further, the intruder can impersonate as an honest user to build the subjective queries. Therefore, we classify the attack model into three levels as follows:

- Level 1: Intruders can only notice the encrypted records stored in cloud. This tracks the familiar ciphertext-only attack model [20].
- Level 2: Intruders are able to get a set of biometric traits in the database D even though the data are encrypted and stored in the cloud. However, the related cyphertexts in the database C are not known, similar to candidate attack model [20].
- Level 3: As well all the capabilities in level 2, intruders in level 3 can be authentic users. Therefore, intruders can forge as numerous verification queries as possible and determine the related cyphertexts. This attack manifests the plaintext attack model [20].

A biometric verification scheme is reliable and safe, if it can end your level-α (α Є {1,2,3}) attack. Noticing that if the presented scheme can endure level-2 and level-3 attacks, but cannot be both the valid users and notice some plaintexts of the biometric database concurrently. This intricate attack is robust and no efficient methods are designed to fortify against this type of attack [14]. In this paper, we concentrate on the collusion attack between venomous user and cloud servers. The association between the cyphertext and plaintexts of the biometric database, are not known to the intruder, which is likely to attack model introduced in [14].

2.3Design Goals

In our proposed scheme, both security and efficiency are considered an order to attain practicality. Specifically, design goals of the presented scheme are defined as follows:

- Security: The privacy of biometric data should be protected during the verification process. Intruders and semi-truthful cloud should study nothing about the delicate information.
- Efficiency: It should be lowest computational costs at both user side and database owner side. To obtain higher efficiency, most of the biometric operations should be executed in the cloud.

2.4 Notations

Here, we list the primary notations used in the remaining section as follows:

- b_i– the i-th sample FingerCode, denoted as an n-dimensional vector $b_i = [b_{i1}, b_{i2}, \cdots b_{in}]$.
- B_i– the extended sample FingerCode of b_i, denoted as an(n+1)-dimensionalvector $B_i = [b_{i1}, b_{i2}, \cdots b_{i(n+1)}]$, where $b_{i(n+1)} = -0.5(b^2_{i1} + b^2_{i2} + \cdots + b^2_{in})$.
- b_c– the query FingerCode, denoted as an n- dimensional vector $b_c = [b_{c1}, b_{c2}, \cdots b_{cn}]$.

- B_c – the extended query FingerCode of b_c, denoted as an (n + 1)-dimensional vector $B_c = [b_{c1}, b_{c2}, \cdots b_{c(n+1)}]$, where $b_{c(n+1)}=1$.
- W – the secret keys collection, denoted asW=(M_1, M_2, M_3, H, R), where M_1, $M_2$and M_3 are (n+1) × (n + 1) invertible matrices, and H, R are (n+1)- dimensional row vectors.
- I_i – the searchable index associated with thei-th sample FingerCodeb_i.
- Γ - the query FingerCode collection constructed by the attacker, denoted as Γ = ($\tilde{b}_1, \tilde{b}_2, \cdots \tilde{b}_{t+1}$).
- $\tilde{B}_i$ – the i-th extended query FingerCode constructed by the attacker, denoted as $\tilde{B}_i$= [$\tilde{b}_{i1}$, $\tilde{b}_{i2}$, ... $\tilde{b}_{i(n+1)}$], where $\tilde{b}_{i(n+1)} = 1$.

3. A new biometric verification scheme

In this section, we describe the details of the proposed biometric verification scheme.

3.1 Overview

We build a novel biometric verification scheme to address the insufficiency of other existing scheme [13]. To obtain a higher level of privacy protection, a new retrieval way is constructed to endure the level-3 attack. Further, we also reconstruct the ciphertext to reduce the amount of uploaded data and improve the efficiency both in the preparation and verification procedures.

In the remaining part of this section, we present the preparation and the verification process.

3.2 Preparation process

In the preparation process, b_i is thei-th sample feature vector derived from the fingerprint image using a feature extraction algorithm [19]. To be more specific,

b_i is an n-dimensional vector with *l* bits of each element where n = 640 and l = 8.

For ease of verification, bi is extended by adding an (n + 1)-th element as B_i. Then, the database owner encrypts $B_i$with the secret key M_1 as follows:

$$C_i = B_i \times M_1. \quad (1)$$

The database owner further performs the following operation:

$$C_h = M_2^{-1} \times H^T \quad (2)$$

Each FingerCode B_i is associated with an index I_i. After execute the encryption operations, the database owner uploads (C_i, C_h, I_i) to the cloud.

3.3 Verification process

The verification process includes the following steps:

Step 1: When a user has query fingerprint to be identified, he/she first gets the query FingerCode b_c derived from the query fingerprint image. The FingerCode b_c is also an n-dimensional vector. Then, the user sends b_c to the database owner.

Step 2: After receiving b_c, the database owner extends b_c to B_c by adding an (n + 1)-th element equals to 1. Then the database owner randomly generates an (n+1)×(n+1) matrix E. The i-th row vector $E_i = [E_{i1}, E_{i2}, \cdots E_{i(n+1)}]$ is

set as a random vector, where the (n+1)-th element is $(1-\sum_{j=1}^{n} E_{ij} * H_j) / H_{n+1}$, $1 \leq i \leq (n+1)$. After that, the database owner performs the following computation to hide B_c:

$$F_c = [E^T_1 * b_{c1}, E^T_2 * b_{c2}, \ldots E^T_{(n+1)} * b_{c(n+1)}]^T . \quad (3)$$

To securely send F_c to the cloud, the database owner needs to encrypt F_c with the secret keys and a random integer r(r > 0). The computation is performed as follows:

$$C_f = M_1^{-1} \times r \times F_c \times M_2. \quad (4)$$

Then, the database owner sends C_f to the cloud for verification.

Step 3: After receiving C_f from the database owner, the cloud begins to search the FingerCode which has the minimum Euclidean distance with the query FingerCode B_c. P_i represents the relative distance between B_i and B_c as follows:

$$\begin{aligned} P_i &= C_i \times C_f \times C_h \\ &= B_i \times M_1 \times M_1^{-1} \times r \times F_c \times M_2 \times M_2^{-1} \times H^T \\ &= B_i \times r \times F_c \times H^T \\ &= \sum_{j=1}^{n+1} r * b_{ij} * b_{cj}. \end{aligned} \quad (5)$$

In equation 5, the computation result is an integer, which can be used to compare two FingerCodes. For example, to compare the query b_c with two FingerCodes, say b_i and b_z, the cloud computes P_i, P_z and performs the following operation, where $1 \leq i, z \leq t, i \neq z$:

$$P_i - P_z = \sum_{j=1}^{n+1} r * b_{ij} * b_{cj} - \sum_{j=1}^{n+1} r * b_{zj} * b_{cj}$$

$$= \left(\sum_{j=1}^{n} r * b_{ij} * b_{cj} - 0.5 \sum_{j=1}^{n} r * b_{ij}^2\right) - \left(\sum_{j=1}^{n} r * b_{zj} * b_{cj} - 0.5 \sum_{j=1}^{n} r * b_{zj}^2\right)$$

$$= 0.5r(dist_{zc}^2 - dist_{ic}^2). \quad (6)$$

As manifested in equation 6, if $P_i - P_z > 0$, the cloud learns that bi matches the query FingerCode much better than b_z. After repeating the operations for the encrypted FingerCodedatabase C in the cloud, the ciphertext C_i which has the minimum Euclidean distance with b_c can be found. The cloud further gets the corresponding index I_i according to the tuple (C_i, C_h, I_i) and sends it back to the database owner.

Step 4: After receiving the index I_i, the database owner gets the corresponding sample FingerCode b_i in the database D and calculates the accurate Euclidean distance between b_i and b_c by using the formula $dist_{ic} = \sqrt{\sum_{j=1}^{n} (b_{ij} - b_{cj})^2}$. Then, the database owner compares the Euclidean distance with the standard threshold. If the distance is less than the threshold value, the query is identified else, the verification fails.

Step 5: At last, the database owner returns the verification result to the user.

4. Security analysis

In this section, we first prove that our scheme is secure under level-2 and level-3 attacks, and then we will show the proposed scheme can endure the attack proposed by Zhu et al. [18].

4.1Security analysis under level-2 attack

According to the attack scenario 2, an attacker can obtain some plaintexts of the biometric database, but does not know the corresponding ciphertexts.

We consider C_i which is obtained by multiplying B_i and M_1. Since the mapping relationship between B_i and C_i is not known, it is impossible for the attacker to compute B_i and M_1.

Schemes	Level 1 attack	Level 2 attack	Level 3 attack
Existing scheme	Yes	Yes	No
Our scheme	Yes	Yes	Yes

TABLE 1. Security comparison with Existing and proposed schemes.

4.2Security analysis under level-3 attack

In the level-3 attack, besides the knowledge of encrypted data in the cloud, the attacker can forge a large number of query FingerCodes Γ as inputs. In the following, we will show the proposed scheme is secure by proving that the secret keys cannot be recovered.When colluding with the cloud, the attacker gets C_f and C_h, and then performs the following operation:

$C_f \times C_h = M_1^{-1} \times r \times F_c \times M_2 \times M_2^{-1} \times H^T$

$= M_1^{-1} \times r \times F_c \times H^T$

$= M_1^{-1} \times r \times B_c^T.$ (7)

In equation 7, since r is a positive random integer in identification process, the attacker cannot compute the secret key M_1^{-1}directly.

Playacting a valid user, the intruder can build t query FingerCodes $\Gamma=[\tilde{b}_1, \tilde{b}_2, \cdots \tilde{b}_t]$ extended as $[\tilde{B}_1, \tilde{B}_2, \ldots \tilde{B}_t]$ for verification, which introduces a set of positive random values r_j and C_{fj}, $1 \le j \le t$. Let $\tilde{P}_j$ be the value of $C_{fj} \times C_h$. The intruder computes $\tilde{P}_j$ as follows:

$\tilde{P}_j = M_1^{-1} \times r_j \times \tilde{B}_j^T.$ (8) After constructing t equations, we have $\tilde{P} = M_1^{-1} \times [\tilde{B}_1^T, \tilde{B}_2^T, \cdots \tilde{B}_t^T] \times \begin{bmatrix} r_1 & 0 & \cdots & 0 \\ 0 & r_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_t \end{bmatrix}$

$= M_1^{-1} \times \tilde{B} \times R.$ (9)

Here $[\tilde{B}_1^T, \tilde{B}_2^T, \cdots \tilde{B}_t^T]$ is denoted as $\tilde{B}$, $\begin{bmatrix} r_1 & 0 & \cdots & 0 \\ 0 & r_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_t \end{bmatrix}$ is denoted as R. In this equation, $\tilde{P}$ is an (n+1)×t matrix known to the attacker, $\tilde{B}$is an (n + 1) × t matrix constructed by the attacker, R is an t × t matrix,since r_j is arandom positive integer, it is unknown to the attacker.

We then demonstrate that the attacker cannot recover M_1 according to **Theorem 1**.

Theorem 1: Assume after t equations are constructed, M_1 cannot be computed in $\tilde{P} = M_1^{-1} \times \tilde{B} \times R$. When (t + 1)equations are constructed, the following equation holds, and M_1 cannot be recovered.

$$[\tilde{P}|\tilde{P}_{t+1}] = M_1^{-1} \times [\tilde{B}|\tilde{B}_{t+1}^T] \times \begin{bmatrix} R & 0 \\ 0 & r_{t+1} \end{bmatrix}. \quad (10)$$

Proof: This theorem is proven with the inductive method. When t = 1, M_1 cannot be computed in equation 16. Assume the equation 17 holds, where (t > 1). When (t + 1) query FingerCodes are constructed, we get:

$$[\tilde{P},\tilde{P}_{t+1}] = [M_1^{-1} \times \tilde{B}, M_1^{-1} \times \tilde{B}_{t+1}^T] \times \begin{bmatrix} R & 0 \\ 0 & r_{t+1} \end{bmatrix}. \quad (11)$$

For (t + 1)-th query FingerCode $\tilde{B}_{t+1}$, we have

$$\tilde{P}_{t+1} = M_1^{-1} \times \tilde{B}_{t+1}^T \times r_{t+1}. \quad (12)$$

$$\tilde{B}_{t+1} \times (M_1^{-1})^T = (r_{t+1}^{-1})^T \times \tilde{P}_{t+1}^T. \quad (13)$$

Let $(d_1, d_2, \cdots d_{n+1})$ be the vector $(r_{t+1}^{-1})^T \times \tilde{P}_{t+1}^T$ where $d_j = (r_{t+1}^{-1})^T \times \tilde{P}_{(t+1)j}^T$. $(M_1^{-1})^T = (m_1^T, m_2^T, \cdots, m_{n+1}^T)^T$, where m_j denotes a row vector in M_1^{-1},where j ranges from 1 to (n+1). The following equations hold:

$$\tilde{B}_{t+1} \times (m_1^T, m_2^T, \cdots, m_{n+1}^T)^T = (d_1, d_2, \cdots d_{n+1}) \quad (14)$$

$$\tilde{B}_{t+1} \times m_j^T = d_j. \quad (15)$$

Equation 15 is a typical non-linear homogeneous equation. Since the rank of $\tilde{B}_{t+1}$ is $r(\tilde{B}_{t+1})$, we assume the result is $\alpha_1\beta_1 + \alpha_2\beta_2 + \cdots + \alpha(n-r(\beta_{t+1}))\beta(n-r(\beta_{t+1}))$. We further state the special solution of equation 15 is β^*. which satisfies the formula $\tilde{B}_{t+1} \times m_j^T = d_j$. Because $d_j = (r_{t+1}^{-1})^T \times \tilde{P}_j^T$, $(r_{t+1}^{-1})^T$ is included in the special solution β^*. For m_j^T in matrix $(M_1^{-1})^T$, the particular solution of m_j^T is $\alpha_1\beta_1 + \alpha_2\beta_2 + \cdots + \alpha_{((n-r(\beta_{t+1})))}\beta_{((n-r(\beta_{t+1})))} + \beta^*$. Since r is a random integer, the special solution β^* is uncertain as well, which means the attacker cannot derive the exact particular solution for m_j^T in $(M_1^{-1})^T$.

Therefore, when (t+1) query FingerCodes are constructed, the secret key M_1 cannot be computed by the attacker as well. As discussed above, the attacker cannot recover the secret key even if he is a malicious user. Therefore, the attacker cannot recover the biometric data as well.

Moreover, we compare our scheme with the schemes proposed in [13] and [14]. According to Table 1, other schemes have some weaknesses, while our scheme is secure under all the three level attacks.

4.3 Security analysis under the attack proposed by Zhu et al.

Zhu et al. [18] showed an attack for Yuan and Yu's scheme. In their attack, the attacker observes the cloud and gets the values of relative distance. According to the equation 1, 2, 3, 4, the relative distance in Yuan and Yu's scheme can be computed as follows:

$$\begin{aligned} P_i &= C_h \times C_i \times C_f \times C_r \\ &= H \times M_1^{-1} \times M_1 \times D_i \times M_2 \times M_2^{-1} \times F_c \\ &\quad \times M_3 \times M_3^{-1} \times R^T \\ &= H \times D_i \times F_c \times R^T \end{aligned}$$

$$= \sum_{j=1}^{n+1} b_{ij} * b_{cj}$$

$$= B_i \times B_c^T . \quad (16)$$

As shown in equation 16, P_i is an integer which the attacker can get in the cloud, B_c is the extended query FingerCode which can be constructed by the attacker pretending to be a user. B_i is the extended sampleFingerCode which is sensitive and should not be leaked. To recover B_i , the attacker can construct t query FingerCodes $\Gamma=[\tilde{b}_1, \tilde{b}_2, \cdots \tilde{b}_t]$ extended as $[\tilde{B}_1, \tilde{B}_2, \ldots \tilde{B}_t]$ for verification. $\tilde{P}_{ij}$ denotes the relative distance between the sample FingerCode B_i and the query FingerCode $\tilde{B}_j$ where $1 \le j \le t$. Then, the intruder has:

$$[\tilde{P}_{i1}, \tilde{P}_{i2}, \cdots \tilde{P}_{it}] = [b_{i1}, b_{i2}, \cdots b_{i(n+1)}] \times [\tilde{B}_1^T, \tilde{B}_2^T, \cdots \tilde{B}_t^T] \quad (17)$$

In this equation, $\tilde{P}_{ij}$ and $\tilde{B}_j$ are known to the . For each element in B_i , it can be recovered if t equations are built, where $t > (n + 1)$.

Then, we demonstrate the proposed scheme is secure under the attack proposed by Zhu et al. In the proposed scheme, $\tilde{P}_{ij}$ is set as the relative distance between B_i and $\tilde{B}_j$.

$$\tilde{P}_{ij} = C_i \times C_{fj} \times C_h$$

$$= r_j \times B_i \times \tilde{B}_j^T . \quad (18)$$

r_j is the j-th positive random integer in t verification processes. The attacker constructs t query FingerCodes and gets the equation as follows:

$$[\tilde{P}_{i1}, \tilde{P}_{i2}, \cdots \tilde{P}_{it}] = [b_{i1}, b_{i2}, \cdots b_{i(n+1)}] \times [r_1 \tilde{B}_1^T, r_2 \tilde{B}_2^T, \cdots r_t \tilde{B}_t^T]$$

$$= [b_{i1}, b_{i2}, \cdots b_{i(n+1)}] \times [\tilde{B}_1^T, \tilde{B}_2^T, \cdots \tilde{B}_t^T] \times \begin{bmatrix} r_1 & 0 & \cdots & 0 \\ 0 & r_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_t \end{bmatrix} . \quad (19)$$

In this equation, r_j is a positive random integer which is unknown to the attacker. For every element in B_i , after t computations, the attacker can only get the value of $r_j * b_{iq}$ where $t > (n + 1)$, $1 \le q \le (n + 1)$. For the reason that r_j is a random integer, $r_j * b_{iq}$ is also unexpected which means the attacker cannot acquire B_i . Thus, the proposed scheme can resist the attack proposed by Zhu et al.

5. Performance analysis

We implement a cloud with blockchain-based privacy preserving fingerprint verification system. In order to estimate the performance of presented scheme. We utilize 32 GB memory and 2 nodes with 6 core 2.10 GHz Intel Xeous CPU. We use laptop with Intel Core 2.40 GHz CPU and 8G. Likewise to [13] and [14], the query FingerCodes are arbitrarily choosed from the database which is built with 640-entry vectors arbitrarily.

5.1 Complexityanalysis

Table 2 briefs out the communication and computation costs on the database owner side. In this, the cost of each matrix multiplication is $O(n^3)$, where n is the dimension of a FingerCode, and the cost of sorting fuzzy Euclidean

distances has the time complexity of O(m log m). As demonstrated in Table 2, our proposed scheme has lesser complexities in the preparation phase. Thus, much of bandwidth and computation costs can be hoard for the database owner. In the verification phase, the computation complexity of our proposed scheme is lesser than that in [14]. This is because, our proposed scheme performs vector-matrix multiplication to identify the best match, however [14] requires to execute matrix-multiplication operations.Even though the complexity of our proposed scheme issimilar as that in [13], we intensify that [13] offers the sturdy security to attain that fast computation of P_i.Further,our proposed scheme executes only a few multiplication operations, hence attain better performance.

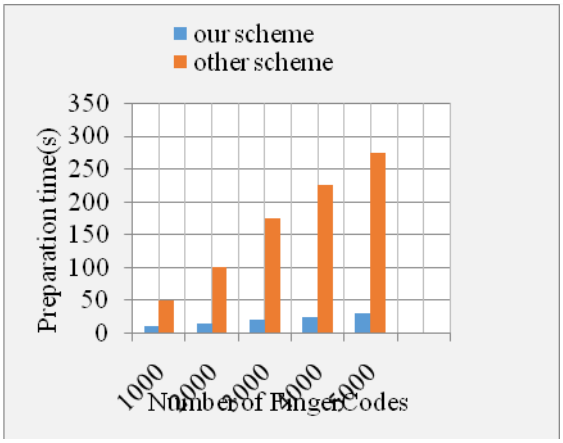
		Phases	Existing scheme	Our Scheme
Computation	Database owner	Preparation	$O(mn^3)$	$O(mn^2)$
		Verification	$O(n^3)$	$O(n^3)$
		Retrieval	$O(n)$	$O(n)$
	Cloud owner	Verification	$O(mn^3 + m\log n)$	$O(mn^2 + m\log n)$
	User	Verification	/	/
Communication	Database owner	Preparation	$O(mn^2)$	$O(mn)$
		Verification	$O(n^2)$	$O(n^2)$
		Retrieval	O(1)	O(1)
	Cloud owner	Verification	/	/
		Retrieval	O(1)	O(1)
	User	Verification	O(1)	O(1)

TABLE 2. A summary of complexity costs. In the table, m represents the number of FingerCodes in the biometric database; n<<m.

5.2 Experimental evaluation

5.2.1Preparation phase

Fig. 3 and Fig. 4 represents the computation and communication costs in the phase of preparation with numerous FingerCodes differing from 1000 to 5000. As manifested in Fig.2, in our proposed scheme, registering 5000 FingerCodes requires 29.37s, which can hoard almost 88.85% and 90.58% time cost comparing to [13] and [14]. This is because while encrypting a sample FingerCode, in the proposed scheme, only one matrix is required which leads to a very few matrix multiplication operations. Fig. 3 describes the bandwidth costs of existing and proposed scheme. Since the data offshored to the cloud as blockchain, the data is in the form of vectors in comparison with matrices in the other schemes, the cost of communication in one proposed scheme is lesser than [13], [14].



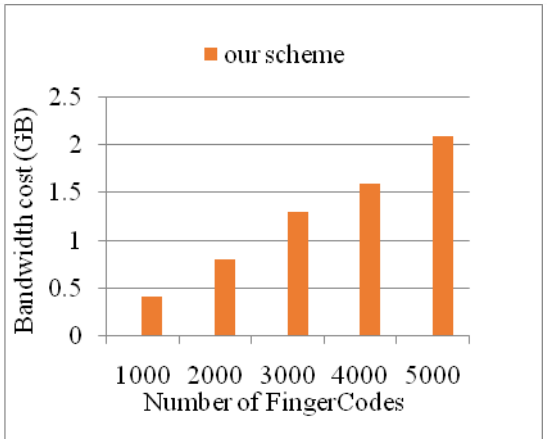
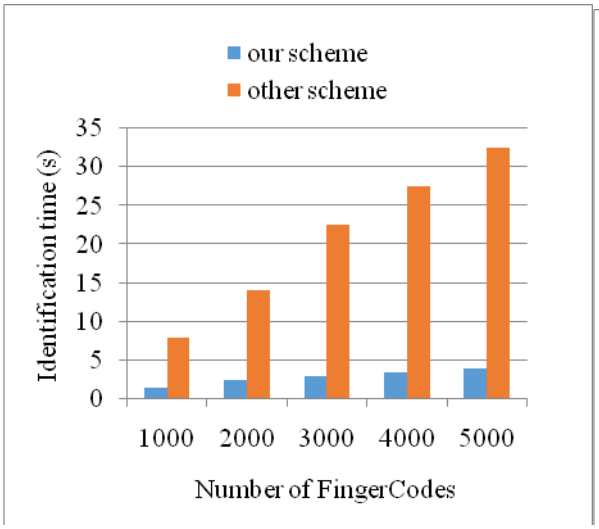


Figure 3. Time costs in the preparation phase.Figure 4. Bandwidth costs in the preparation phase.

5.2.2 Verification phase

Fig. 5 and Fig. 6 represents the computation and communication costs in the verification phase with numerous FingerCodes varying from 1000 to 5000. Fig.4 describes the existing and the proposed schemes extend linearly as the database size increases. Our proposed scheme can save about 56% time cost. Since it uses a few matrix multiplication operations than [13]. Comparing to [14], the verification time can be saved up to 84.75%, the reason is it executes the vector-matrix multiplication operation. As shown Fig.5, the bandwidth costs of the existing and the proposed scheme are similar, since all schemes required to transfer a matrix in the verification phase.



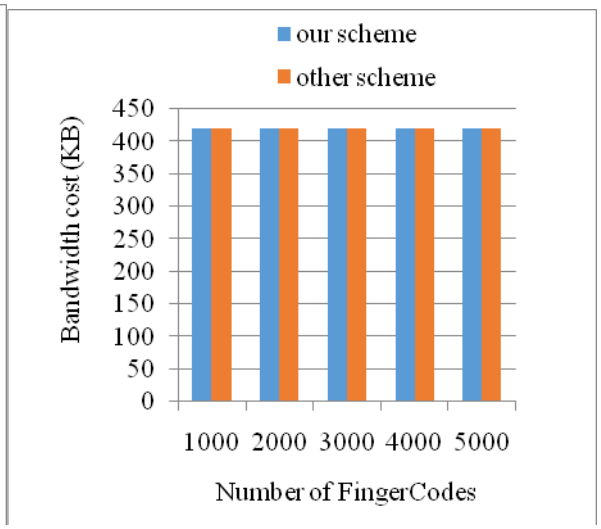


Figure 5. Time costs in the verification phase. Figure 6. Bandwidth costs in the verification phase.

6. Relatedwork

Associated works on privacy-preserving biometric verification are described in this section. Later, an efficient biometric verification schemes have been introduced. Wang and Hatzinakos introduced a privacy-preserving

face recognition scheme [22].Particularly, a face recognition method is built by measuring the similarity between sorted index numbers vectors. Wong and Kim presented aprivacy-preserving biometric matching protocol for iris codes identification. In their protocol, it is computationally not possible for a venomous user to act as a valid user. Barni et al. [10] proposed a FingerCode verification protocol based on Homomorphic Encryption technique. But,alldistances are evaluated between the sample FingerCodes in the database and the query, in which result in burden as the fingerprints size increases. To advance its efficiency, Evans et al. [12] introduced a protocol which decreases the time for verification. They utilized an advanced Homomorphic Encryption algorithm to evaluate the Euclidean distance and constructed novel garbled circuits to identify the minimumdistance. But, their scheme has proved to be insecure in [15] and [14].

7. Conclusion and Future works

We propose a privacy-preserving biometric verification in cloud computing with Blockchain scheme. To understand the secure and efficiency necessity, we present a cloud verification certification. The study of proposed scheme manifests that it can endure the potential attack while storing the encrypted index and feature of fingerprint data as blocks in cloud using Blockchain technology. In the performance evaluation section, we showed that the proposed scheme is efficiency as well.

As a future work we plan to improve the performance of a blockchain network that cope with the massive amount of data without reducing the speed of transaction.

REFERENCES

[1] A. Jain, L. Hong, and S. Pankanti, ‘‘Biometric identification,’’ Commun. ACM, vol. 43, no. 2, pp. 90–98, 2000.

[2] R. Allen, P. Sankar, and S. Prabhakar, ‘‘Fingerprint identification technology,’’ in Biometric Systems. London, U.K.: Springer, 2005, pp. 22–61.

[3] J. de Mira, Jr., H. V. Neto, E. B. Neves, and F. K. Schneider, ‘‘Biometric- oriented iris identification based on mathematical morphology,’’ J. Signal Process. Syst., vol. 80, no. 2, pp. 181–195, 2015.

[4] S. Romdhani, V. Blanz, and T. Vetter, ‘‘Face identification by fitting a 3D morphable model using linear shape and texture error functions,’’ in Proc. Eur. Conf. Comput. Vis., 2002, pp. 3–19.

[5] Y. Xiao et al., ‘‘A survey of key management schemes in wireless sensor networks,’’ Compute Commun., vol. 30, nos. 11–12, pp. 2314–2341, Sep. 2007.

[6] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, ‘‘An effective key manage- ment scheme for heterogeneous sensor networks,’’ Ad Hoc Netw., vol. 5, no. 1, pp. 24–34, 2007.

[7] X. Du and H. H. Chen, ‘‘Security in wireless sensor networks,’’ IEEE Wireless Commun. Mag., vol. 15, no. 4, pp. 60–66, Aug. 2008.

[8] X. Hei and X. Du, ‘‘Biometric-based two-level secure access control for implantable medical devices during emergencies,’’ in Proc. IEEE INFO- COM, Apr. 2011, pp. 346–350.

[9] X. Hei, X. Du, J. Wu, and F. Hu, ''Defending resource depletion attacks on implantable medical devices,'' in Proc. IEEE GLOBECOM, Dec. 2010, pp. 1–5.

[10] M. Barni et al., ''Privacy-preserving fingercode authentication,'' in Proc. 12th ACM Workshop Multimedia Secure, 2010, pp. 231–240.

[11] M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich, ''SCiFI—A system for secure face identification,'' in Proc. IEEE Symp. Secure Privacy (SP), May 2010, pp. 239–254.

[12] D. Evans et al., ''Efficient privacy-preserving biometric identification,'' in Proc. 17th Conf. Netw. Distrib. Syst. Secur. Symp. (NDSS), 2011, pp. 1–40.

[13] J. Yuan and S. Yu, ''Efficient privacy-preserving biometric identification in cloud computing,'' in Proc. IEEE INFOCOM, Apr. 2013, pp. 2652–2660.

[14] 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 Proc. Eur. Symp. Res. Comput. Secur., 2015, pp. 186–205.

[15] Y. Zhu, Z. Wang, and J. Wang, ''Collusion-resisting secure nearest neigh- bor query over encrypted data in cloud,'' in Proc. IEEE/ACM 24th Int. Symp. Quality Ser. (IWQoS), Jun. 2016, pp. 1–6.

[16] S. Pan, S. Yan, and W. Zhu, ''Security analysis on privacy-preserving cloud aided biometric identification schemes,'' in Proc. Australasian Conf. Inf. Secur. Privacy, 2016, pp. 446–453.

[17] C. Zhang, L. Zhu, and C. Xu, ''PTBI: An efficient privacy-preserving biometric identification based on perturbed term in the cloud,'' Inf. Sci., vols. 409–410, pp. 56–67, Oct. 2017.

[18] Y. Zhu, T. Takagi, and R. Hu, ''Security analysis of collusion-resistant nearest neighbor query scheme on encrypted cloud data,'' IEICE Trans. Inf. Syst., vol. E97.D, no. 2, pp. 326–330, 2014.

[19] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, ''Filterbank- based fingerprint matching,'' IEEE Trans. Image Process., vol. 9, no. 5, pp. 846–859, May 2000.

[20] H. Delfs, H. Knebl, and H. Knebl, Introduction to Cryptography. Berlin, Germany: Springer, 2002.

[21] K. Liu, C. Giannella, and H. Kargupta, ''An attacker's view of dis- tance preserving maps for privacy preserving data mining,'' in Knowledge Discovery in Databases: PKDD (Lecture Notes in Computer Science). Heidelberg, Germany: Springer, 2006, pp. 297–308.

[22] Y. Wang and D. Hatzinakos, ''Face recognition with enhanced privacy protection,'' in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., Apr. 2009, pp. 885–888.