SviaB: Secure and verifiable multi-instance iris remote authentication using blockchain

Mahesh Kumar Morampudi1,2 | Munaga V. N. K. Prasad3 | Surya Narayana Raju Undi1

1Department of Computer Science and Engineering, National Institute of Technology-Warangal, Telangana, India
2Department of Computer Science and Engineering, SRM University AP, Amaravati, Andhra Pradesh, India
3Institute for Development and Research in Banking Technology (IDRBT), Hyderabad, India

Correspondence
Mahesh Kumar Morampudi, Department of Computer Science and Engineering, SRM University AP, Amaravati, Andhra Pradesh, India.
Email: morampudimahesh@gmail.com

Abstract
Homomorphic encryption (HE) is the most widely explored research area in the construction of privacy-preserving biometric authentication systems because of its advantages over cancellable biometrics and biometric cryptosystems. However, most of the existing privacy-preserving biometric authentication systems using HE assume that the server performs computations honestly. In a malicious server setting, the server may return an arbitrary result to save computational resources, resulting in a false accept/reject. To address this, secure and verifiable multi-instance iris authentication using blockchain (SviaB) is proposed. Paillier HE provides confidentiality for the iris templates in SviaB. The blockchain offers the integrity of the encrypted reference iris templates as well as the trust of the comparator result. The challenges of using blockchain in biometrics are also addressed in SviaB. Extensive experimental results on benchmark iris databases demonstrate that SviaB provides privacy to the iris templates with no loss of accuracy and trust in the comparator result.

1 | INTRODUCTION

Unlike password or token authentication systems, a biometric authentication system (BAS) has more flexibility because users do not need to carry or remember anything. Fingerprint, iris, face, etc. are the commonly used biometric modalities [1, 2]. Properties such as stability and uniqueness make the iris the most widely used of the various biometric applications in comparison with other biometric modalities [3, 4]. On the other hand, such benefits are accompanied by challenges. The vast increase in the use of BAS in various applications has raised security and privacy concerns [5]. Because biometric data is unique to each person, it is irrevocable if it becomes compromised. Leakage or disclosure of biometric data to unauthorized persons/servers can cause the consequence of ‘Lose it once; it’s gone forever’. Several attacks such as hill climbing attack [6], replay attack, masquerade attack [7], and the stolen-token attack [8] occur if templates are stored in an unprotected manner that makes the system vulnerable. In 2018, hackers stole 1 billion Indian users’ biometrics from Aadhaar [9]. As a result, a biometric system capable of protecting sensitive biometric information needs to be designed. In general, the templates generated by biometric template protection schemes (BTPS) must satisfy properties such as diversity, irreversibility, and revocability. At the same time, the system’s template protection method must retain properties such as performance, storage requirements, and speed that are present on systems without a template protection method. BTPS can be broadly classified as cancellable biometrics, homomorphic encryption (HE), and biometric cryptosystem.

Cancellable biometrics suffer from performance degradation to the unprotected system [10]. Biometric cryptosystems use the auxiliary data (AD), and compromise of the AD leads to leakage of biometric information [11], which results in the entire system becoming vulnerable. HE was introduced as a BTPS to solve the limitations of cancellable biometrics and biometric cryptosystems [12]. HE offers operations such as multiplication and addition on the encrypted data itself; as a result, templates are secure not only during transit but also during computation. HE is categorized as partial HE (PHE), somewhat HE (SHE), and fully HE (FHE). FHE can perform unlimited multiplications and additions on the encrypted data. SHE can perform a limited number of multiplications and additions on the encrypted data. PHE can perform either
multiplication or addition, but not both, on the encrypted data. In the literature, several works are proposed to provide the privacy of biometric templates using HE [13–20].

Most state-of-the-art works have assumed that the server performs computations honestly. However, to save computational or storage resources, a server may return an arbitrary result. Confidentiality of the iris templates in state-of-the-art works is achieved because the computations are carried on the encrypted data, but they fail to provide trust to the user/client device that the computed result is correct. This scenario leads to false accept/reject. We utilize the concepts of smart contract and blockchain (BC) to address these issues. Blockchain (BC) with smart contract has been developed to allow decentralized consensus between two non-trusting agents. Mahesh et al. [21] integrated BC with ElGamal HE and proposed BC-based multi-instance iris authentication using additive ElGamal homomorphic encryption (BMIAE) to solve the override comparator attack. The cost and time to authenticate a person are greater in BMIAE. This work solves the limitation of BMIAE [21].

1.1 Contribution

• We introduce a system, secure and verifiable multi-instance iris authentication (MIIA) using BC (SviaB). SviaB combines BC technology and Paillier HE [22]. Pailler HE provides confidentiality for iris templates. BC provides integrity for the encrypted reference iris templates as well as trust in the comparator result.
• SviaB addresses the challenges of using BC for biometrics, such as expensive storage cost and privacy [23].
• SviaB solves attacks such as modify templates, intercept channel, and override comparator in BAS, as shown in Figure 1.
• SviaB has experimented on publicly available benchmark iris databases and to check the effectiveness, performance measures like Equal Error Rate (EER), separability measures like Kolmogorov-Smirnov (KS)-test and d-prime are considered.

1.2 Organization

The remainder of the article is organized as follows. Section 2 explains the necessary preliminaries. In Section 3, the related work is discussed. Section 4 describes the proposed method. The implementation details and experimental and security analyses are illustrated in Section 5. The paper is concluded in Section 6.

2 PRELIMINARIES AND NOTATIONS

2.1 Autoencoder

An autoencoder is an unsupervised neural network method that optimizes a rebuilding of the input data in the output layer through a hidden layer of chosen dimensions. Similar to state-of-the-art dimensionality reduction techniques such as linear discriminant analysis (LDA), principal component analysis (PCA), isometric mapping (ISOMAP) etc., autoencoder can be used to reduce the high-dimensional feature vector [24]. Input, hidden, and output are the layers present in the autoencoder. The dimensions of the output and input layers are the same, whereas the hidden layer contains fewer dimensions. The autoencoder consists of two phases, (1) encoder and (2) decoder. An encoder converts the input data into a hidden code, and the decoder reconstructs the original input data from the hidden code. The input and output for an autoencoder are $I \in [0, 1]^d$ and $O \in [0, 1]^{d'}$, where $d$ is the number of dimensions. Firstly, the encoder maps the input into hidden (or) latent code, $b \in [0, 1]^b$, $d' < d$ using the transformation given in Equation (1):

$$b = S(W \times I + b)$$

where $S$ is a sigmoid function, $W$ is a weight matrix, and $b$ is the bias. The hidden code, $b$ is then converted back into $O$ with the same dimension as $I$ by using the decoder. The conversion occurs through the transformation given in Equation (2):

$$O = S(W' \times b + b')$$

where $S$ is a sigmoid function, $W'$ is a weight matrix of the reverse mapping, and $b$ is the bias. The average reconstruction error is maximized by optimizing the parameters $(W, b, b')$. The reconstruction error can be measured by either squared error, $L(I, O) = \|I - O\|^2$, or binary cross-entropy, $L(I, O) = -\sum_{k=1}^{d}[I_k \log O_k + (1 - I_k) \log(1 - O_k)]$. To use the autoencoder as a dimensionality reduction technique, use the data obtained in hidden layer and discard the decoder phase.

A brief introduction to BC and the need to use smart contract and BC in biometric authentication can be found in [21].

2.2 Notations

The notations used in SviaB are described in Table 1.
3 | RELATED WORK

3.1 | Homomorphic encryption applied to biometric authentication

Upmanyu et al. [13] proposed a secure protocol for biometric verification named ‘Blind Authentication’ by using Rivest–Shamir–Adleman and Paillier [22]. The ‘blind authentication’ protocol considers the enrolment server a trusted entity; as a result, it provides only privacy-preserving classification and fails to provide privacy-preserving enrolment. Osadchy et al. [25] introduced a secure face identification (‘Scifi’) system using the Paillier cryptosystem [22]. The ‘Scifi’ system yields results superior to those of existing works, even in illumination-invariant conditions. Rahulamathavan et al. [26] suggested a method to recognize the expression of a face by using the properties of Paillier and computed the required operations on encrypted data. Pastoriza et al. [14] proposed a secure face verification system in a non-interactive manner that can be applied to lightweight devices. The authors proposed a HE scheme to perform the operations on encrypted data. Sedenka et al. [27] designed a secure biometric authentication in an outsourced environment. Haghbighat et al. [28] suggested a biometric verification in a cloud environment. The method uses searching-based instead of distance-based matching.

Xiang et al. [29] introduced secure face recognition with computation in a cloud server using public key encryption and a fully HE algorithm. The client is able to verify the correctness of the computed result by the cloud server. Hahn et al. [30] introduced an efficient and secure identification system using symmetric HE. The system performs better than those of existing works. Gomez et al. [18] proposed a template protection approach for multi-biometric recognition using Paillier. The final comparison is performed on the plaintext by the server; as a result, a breach is introduced into the security of the system. Santosh et al. [31] used the Paillier and elliptic curve encryption techniques to provide security of biometric templates stored in a cloud server. Zhu et al. [32] designed a method named efficient fingerprint authentication (‘e-Finga’) for secure online fingerprint authentication. ‘e-Finga’ uses lightweight multiparty polynomial aggregation and multiparty random masking techniques to provide security. A lightweight encryption scheme named ‘threshold predicate encryption’ (TPE) is proposed by Zhou et al. [19]. A privacy-preserving user-centric BAS (PassBio) is proposed using TPE. Barni et al. [33] designed a secure multimodal biometric authentication that combines iris and face templates. Guo et al. [34] used randomness techniques instead of HE to provide the privacy of the face templates result in a good performance.

3.2 | Blockchain for biometrics

The limitations and advantages of using biometrics in BC and vice versa are presented in Delgado-Mohatar et al. [23]. The authors extended their work in [35] to store the biometric templates in the BC using Merkle trees, direct hashing, and on-chain. The execution time and storage cost are less for Merkle tree-based storage when compared with direct hashing and on-chain. The limitations of BC for biometrics is not addressed. Delgado-Mohatar et al. [36] also analyzed cost and performance factors to store the protected and unprotected biometric templates and on-chain, off-chain biometric matching. Mohsin et al. [37] used BC to achieve availability and integrity in a finger vein verification system. Mahesh et al. [21] used the concepts of BC to provide trust in the computed result. The drawback of their method is the increased time and cost required to authenticate a person.

The literature reveals that privacy-preserving biometric systems based on HE assume that the server/cloud server performs computations honestly. However, because of financial or timing reasons, the server/cloud server assigned to a task may not honestly perform the computation. It may return an arbitrary result that leads to false accept or false reject. Therefore, SviaB is proposed, which provides not only confidentiality for the iris templates but also trust in the comparator result.

4 | SECURE AND VERIFIABLE MULTI-INSTANCE IRIS AUTHENTICATION USING BLOCKCHAIN

SviaB is the first MIIA system that combines BC technology and Paillier HE [22] to provide not only privacy to the iris templates as well as trust in the computed result. The confidentiality of iris templates is assured in SviaB by using Paillier HE. The BC provides the trust in the computational result by computing the distance in a smart contract. Figure 2 shows the block diagram of SviaB. The client device (CD), trusted authenticator (TA), cloud server (CS) and blockchain (BC) are the entities involved in SviaB. SviaB consists of two phases, namely enrolment and
authentication. Algorithms 1 and 2 depict the steps involved in the enrolment and authentication phases of SviaB.

4.1 Assumptions

SviaB assumes the following:

- The CD is a trusted entity and lacks computational and storage resources.
- The CS does not perform computations honestly.
- The TA is a trusted entity that produces public and secret keys differently for each user. The secret keys of users are stored securely and broadcast the public keys to the CD.
- The CS and CD need not store the entire ledger of the BC network, and the contract address of the smart contract is shared to the CS and CD before the enrolment phase.
- The underlying consensus algorithm of the BC is secure and robust against security attacks on the BC.

Algorithm 1 Enrolment phase of SviaB

**Input:** Reference images of both left and right eye. Identity label, UID

1: The TA generates the public key, \( P_k \) and secret key \( S_k \).
2: The CD generates the iris templates from the reference left and right iris images using University of Salzburg tool kit [38].
3: The CD generates the reference fused iris templates, \( F_i \) and send \( F_i \) to TA. //Refer Section 4.2.1
4: The TA reduces the dimensions of \( F_i \) and send the compressed template, \( X_i \) to the CD. //Refer section 4.2.2
5: The CD encrypts \( X_i \) and sends \((e(X_i), UID)\) to CS and BC.
6: The BC calculates the hash value of \( e(X_i) \), \( Hr \) and stores \((Hr, UID)\). (SviaB stores only \( Hr \) in BC and \( e(R) \) in CS to overcome the limitation of BC.)

Algorithm 2 Authentication phase of SviaB

**Input:** Probe images of both left and right eye. Identity label, UID

**Output:** Reject or Accept

1: The CD generates the iris templates from the probe iris images using University of Salzburg tool kit [38].
2: The CD generates the probe fused iris templates, \( G \) and send \( G \) to the TA. //ReferSection 4.2.1
3: The TA reduces the dimensions of \( G \) and send the compressed template, \( Y \) to the CD. //

Refer section 4.2.2

4: The CD encrypts \( Y \) and sends \((e(Y), UID)\) to the BC.
5: The BC retrieves \( e(X_i) \) with the same identity label from the CS.
6: The hash value of retrieved \( e(X_i) \), \( Hp \) is computed by the BC and compares \( Hr \) and \( Hp \). The BC computes the distance \( d \) if the hash values are same and send \( e(d) \) to the TA. Otherwise, it will send an Integrity Failed message to the TA.
7: The TA decrypts \( e(d) \) using \( S_k \) and obtains \( R \). The TA compares \( R \) with a threshold, \( r \) and returns accept/reject.

In BC, the storage cost is expensive when compared with computation [23]. To overcome this limitation, SviaB stores only the hash value of the encrypted reference templates in the BC and stores the encrypted reference templates in the CS itself.

4.2 Fusion and reducing the dimensions of iris template

Fusion and reducing the dimensions of iris template are the two phases involved in this section. In the fusion phase, the iris templates from the right and left irises are fused. The size of the fused iris template is reduced using a non-linear dimensionality reduction technique autoencoder in the reduction phase.

4.2.1 Fusion of iris templates

The CD performs the fusion of iris templates in this phase. The extracted iris codes \( E_L, E_R \) are of size \( 1 \times 10,240 \). The fused iris template is created by concatenating \( E_L, E_R \) as shown in Equation (3):

\[
Z = \begin{pmatrix} E_L \\ E_R \end{pmatrix}
\]  

(3)

4.2.2 Reducing dimensions of iris template

The performance of the system depends on the size of the iris template. The fused phase produces an iris template of size \( 1 \times 20,480 \). The computational performance can be improved by reducing the size of the iris template. SviaB uses the autoencoder as a technique to reduce the dimensions of the iris code. Autoencoder is a neural network-based reduction technique and performs better than other existing linear dimensionality reduction techniques such as PCA and non-linear dimensionality reduction techniques such as LDA and
ISOMAP [39]. Firstly, the TA trains the autoencoder using both encoder and decoder phases, but after training, the data obtained after the encoder phase—that is, in the hidden layer—is considered as the reduced feature vector and discards the decoder phase. As the iris code, that is, the input data to autoencoder, contains 1s and 0s, SviaB uses the cross-entropy as an error function. The 20,480-bit binary vector is given as an input to the autoencoder compressed into 64, 128, 256, and 512 bits.

We considered 64, 128, 256, and 512 nodes in the hidden layer and computed the EER. Table 2 represents the EER values for various sizes of iris template. From Table 2, we can infer that the performance of the latent codes for different sizes of iris template is directly proportional to the number of epochs. An epoch indicates the number of passes of the entire training dataset the autoencoder has completed. We considered the batch size as 16 for all the experiments. The same EER on the uncompressed iris template is obtained for compressed iris template of size 128-bit with 520 epochs, 256-bit with 830 epochs, and 512-bit with 940 epochs, respectively. It is also observed that the time required for the process is proportional to the number of epochs. The latent code of 128-bit size gives the same EER as the uncompressed iris template and considers very little time compared with the other latent codes. Thus, SviaB reduces the dimensions to 128 bits using the autoencoder and uses the 128-bit iris template for further operations.

4.3 | Assuring confidentiality of iris templates using Paillier HE

The confidentiality of the iris templates is assured in SviaB by using Paillier HE [22]. The security of Paillier HE depends on the decisional composite residuosity assumption (DCRA). To compute the distance, HE must satisfy the additive property. Paillier HE [22] is an additive homomorphic cryptosystem. Key generation, decryption, evaluation, and encryption (EEnc) are the four functions involved in the Paillier HE scheme.

The steps needed to perform key generation, encryption, evaluation, and decryption are shown in Figure 3. The random number present in the encryption function provides the randomness to the encryption result of Paillier. Therefore, Paillier prevents chosen plaintext attacks. The template obtained in Section 4.2.2 is encrypted using the encryption function of the Paillier HE scheme [22].

Properties of Paillier HE:

Property 1: Given two encrypted values $e(m_1) = EEnc(P_k, m_1)$ and $e(m_2) = EEnc(P_k, m_2)$ for messages $m_1$ and $m_2$, the addition of two unencrypted values can be obtained by
Decryption of multiplication of the two encrypted values and is given in the Equation (4):

$$D_S(\epsilon(m_1) \cdot \epsilon(m_2) \mod p^2) = m_1 + m_2 \mod p \quad (4)$$

**Property 2:** Given an encrypted value $\epsilon(m) = EEnc(P_k, m)$ for a message $m$ and a constant $k$, decryption of encrypted value raised to a constant results in the multiplication of the original message and the constant and is given in the Equation (5):

$$D_S(\epsilon(m)^k \mod p^2) = m \cdot k \mod p \quad (5)$$

### 4.4 Distance computation on encrypted data and ensuring the trust in the computed result

This section describes the computation of Manhattan distance on the encrypted data and a smart contract code to ensure the trust in the computed distance.

#### 4.4.1 Encrypted distance computation

SviaB considers Manhattan distance to compare the reference and probe iris templates. The reasons behind the selection of Manhattan distance than other distance measures such as Euclidean, hamming distance etc. are as follows. (1) The HE used in SviaB to ensure that the confidentiality of iris templates is Pailler HE, which satisfies only the additive property. If we consider the Euclidean distance as a distance measure, it involves the computation of the square root on the encrypted data that Pailler HE fails to achieve. (2) Manhattan distance is usually preferred over Euclidean distance when there is high dimensionality in the data [40]. The distance $S_{man} = d_{man}(x, y)$ can be precisely calculated on the unencrypted values by using Equation (6):

$$S_{man} = |x - y| \quad (6)$$

The distance can be calculated on the encrypted values by using Equations (4) and (5):

---

**Table 2** Comparison of equal error rate (EER) for various sizes of iris template.

| $d/m$ | EER (in %) | DB1 | DB2 | DB3 | No. of epochs | Time (in secs) |
|-------|------------|-----|-----|-----|---------------|----------------|
| 64-bit | 1.38 | 1.62 | 0.28 | 480 | 24.64         |
| 128-bit | 0.34 | 0.88 | 0.14 |     |               |
| 256-bit | 0.45 | 1.15 | 0.19 |     |               |
| 512-bit | 0.49 | 1.20 | 0.19 |     |               |
| 64-bit | 1.38 | 1.62 | 0.28 | 520 | 26.9          |
| 128-bit | 0.31 | 0.86 | 0.13 |     |               |
| 256-bit | 0.43 | 1.12 | 0.17 |     |               |
| 512-bit | 0.47 | 1.16 | 0.19 |     |               |
| 64-bit | 1.38 | 1.62 | 0.28 | 680 | 35.2          |
| 128-bit | 0.31 | 0.86 | 0.13 |     |               |
| 256-bit | 0.37 | 1.00 | 0.15 |     |               |
| 512-bit | 0.40 | 1.05 | 0.17 |     |               |
| 64-bit | 1.38 | 1.62 | 0.28 | 830 | 43.86         |
| 128-bit | 0.31 | 0.86 | 0.13 |     |               |
| 256-bit | 0.31 | 0.86 | 0.13 |     |               |
| 512-bit | 0.35 | 0.94 | 0.15 |     |               |
| 64-bit | 1.38 | 1.62 | 0.28 | 940 | 48.72         |
| 128-bit | 0.31 | 0.86 | 0.13 |     |               |
| 256-bit | 0.31 | 0.86 | 0.13 |     |               |
| 512-bit | 0.31 | 0.86 | 0.13 |     |               |
| 64-bit | 1.38 | 1.62 | 0.28 | 940 | 48.72         |
| 128-bit | 0.31 | 0.86 | 0.13 |     |               |
| 256-bit | 0.31 | 0.86 | 0.13 |     |               |
| 512-bit | 0.31 | 0.86 | 0.13 |     |               |
| 20480-bit | 0.31 | 0.86 | 0.13 | -   | -             |

Abbreviations: EER, equal error rate.
\( (P_k, S_k) \leftarrow \text{KeyGen}(m, n) \)

**Input:** Two large prime numbers, \( m \) & \( n \) randomly and independently such that \( \gcd(mn, (m-1)(n-1)) = 1 \)

**Output:** Secret key \( (S_k) \) & Public key \( (P_k) \)

- Compute \( p = m \cdot n \) and \( \lambda = \text{lcm}(m, n) \)
- Choose a random integer \( g, g \in \mathbb{Z}_p^{*} \) such that \( p \) divides the order of \( g \).
- Compute \( k = I(g^\lambda \mod p^2) \), where function \( I \) is defined as \( I(u) = \frac{(n-1)}{p} \cdot \phi = k^{-1} \mod p \)

Public key \( (P_k) : [p, g] \) Secret key \( (S_k) : [\lambda, \phi] \).

\[ \varepsilon(m) \leftarrow \text{EEnc}(P_k, m) \]

**Input:** \( P_k \), message \( m \)

**Output:** \( \varepsilon(m) \)

Choose a random element \( r \) such that \( \gcd(r, p) = 1, r \in (0, p) \) and \( r \in \mathbb{Z}_p^{*} \).

Compute \( \varepsilon(m) = g^m \cdot r^p \mod p^2 \).

\[ \varepsilon(R) \leftarrow \text{EEval}(\varepsilon(a), \varepsilon(b), k) \]

**Input:** \( m \), encrypted values \( \varepsilon(a), \varepsilon(b) \), constant \( k \)

**Output:** Computed result \( \varepsilon(R) \)

The addition of two values can be computed as

\[ \varepsilon(R) = \text{add}(a, b) = \varepsilon(a, r_1) \cdot \varepsilon(b, r_2) \mod p^2 \]

where \( r_1, r_2 \) are the random values considered in encryption function.

The multiplication of the original value with a constant can be computed by raising the encrypted value to the constant.

\[ \varepsilon(R) = k \times m_1 \mod p = \varepsilon(a, r_1)^k \mod p^2 \]

\[ m \leftarrow \text{EDec}(S_k, \varepsilon(m)) \]

**Input:** \( \varepsilon(m), S_k \)

**Output:** \( m = I(\varepsilon(m)^{\lambda} \mod p^2) \cdot \phi \mod p \)

**4.5 | Ensuring the trust in computed distance and integrity of encrypted reference templates using smart contract**

The smart contract running on a BC network helps the SviaB address the override comparator attack of BAS (refer to Figure 1). Therefore, the user or CD can ensure the computed distance is correct without including any third party or centralized server. The integrity of the encrypted reference iris templates is also ensured in SviaB by comparing the hash values in BC. The formal smart contract to ensure the trust in the computed distance and integrity of encrypted reference iris template is given in Figure 4. The CD sends \( (\varepsilon(X_i), UID) \) to CS and smart contract during the enrolment phase and invokes the Enrollment function of a smart contract. The CS stores \( (\varepsilon(X_i), UID) \). The hash value of \( \varepsilon(X_i), Hr = H(\varepsilon(X_i)) \) is computed and stores \( Hr \) in refer\[UID\] by the smart contract. The limitations of using BC in biometrics such as expensive storage cost and privacy are described in [23]. To solve expensive storage cost limitations, SviaB stores only the hash value of the encrypted reference iris template instead of \( \varepsilon(X_i) \). To overcome the privacy limitation, SviaB encrypts the fused templates using Paillier HE before sending the template to BC.
The CD sends \( \varepsilon(\gamma), \text{UID} \) to a smart contract and invokes the Verification function during the authentication phase. The smart contract asks the server for \( \varepsilon(X) \) with the same identity label UID. If the server sends \( \varepsilon(X) \) within a stipulated time, \( \delta \), then the Computation function of a smart contract is invoked; otherwise, the Timer function of smart contract gets executed. It sends ‘Session Expired’ message to the TA. When the computation function is invoked, the smart contract computes hash value \( H_p = H(\varepsilon(X)) \). The smart contract computes the distance between \( \varepsilon(X) \) and \( \varepsilon(\gamma) \), if the values of \( H_r \) and \( H_p \) are the same; otherwise, it indicates that \( \varepsilon(X) \) is modified by the intruder. Therefore, a smart contract helps to check the integrity of the encrypted reference template. The differences in the methodology of BMIAE [21] and SviaB are listed in Table 3.

### 5 | IMPLEMENTATION DETAILS AND SECURITY ANALYSIS

The following measures are used to evaluate the efficiency of a biometric system according to biometric information protection [41]:

1. Performance evaluation in terms of EER, \( d' \), and KS tests.
2. Irreversibility and unlinkability analysis.
3. Computational cost in terms of time taken to perform operations.

### Performance Evaluation

**Test environment:** Python 3.5.2 is compiled on a Microsoft Windows 10 Pro operating system to execute the HE code. We implemented the smart contract in Solidity 0.5.0 using a truffle framework. We deployed the contract in a private Ethereum BC network. To validate SviaB, we ran the experiments on a 2.50 GHz Intel Core i5 CPU and 16 GB RAM system.

**Experimental design:** The University of Salzburg tool kit [38] is used in SviaB to extract the iris template from the human eye. Subjects consisting of minimum five left and right iris samples are required to develop a MIIR system. So 106, 208, and 115 subjects from SDUMLA-HMT [42], IITD [43], and CASIA-V3-Interval [44] iris databases are considered to conduct our experiments, as they contain both right and left irises with a minimum of five samples each.

**Accuracy:** The EER obtained for only right iris (ORI), only left iris (OLI), fused iris (FT) and fused compressed iris template (FCT) in unprotected and protected systems for SviaB are shown in Tables 4 and 5. We observe no loss of accuracy in the protected system from Tables 4 and 5.

The baseline comparison of EER, storage cost, and time for SviaB is shown in Table 6. The unprotected and uncompressed template (UUT) indicates the template without compression and encryption, compressed and unprotected template (CUT) indicates the template with compression and without encryption, and compressed and protected template (CUT) indicates the template with compression and encryption.
(CPT) indicates the template with compression and encryption. We can infer from Table 6 that there is no degradation of accuracy with our template protection method. The detection error trade-off curves of SviaB for different databases are shown in Figure 5. The clear separations between genuine and imposter scores of SviaB for different databases are shown in Figure 6. The distributions of false acceptance rate and false rejection rate for considered databases are represented in Figure 7. The separability measures (d’ and KS test values) and EER on encrypted data of SviaB for different databases are shown in Figure 8.

5.2 | Security analysis

The template protection method must satisfy the requirements of irreversibility, revocability, and unlinkability to ensure the privacy of the iris templates. In SviaB, the attacks may happen in the CS, CD, BC network, TA, the communication channel between the server, and the CD.

The security has to be assured for the CD and TA because the CD extracts the features from the iris image and TA produces the \( P_k \) and \( S_k \). SviaB assume that the CD and TA are trusted entities. Therefore, the information about iris and keys in CD and TA are secure. The templates in the CS can be attacked by the intruder. Because CS only stores the encrypted templates in SviaB, and the security of encrypted templates depends on DCRA the templates stored in CS are secure. In SviaB, the hash value of the encrypted template is stored in the smart contract. So, there will not be any information leakage. Because it is infeasible to decrypt the encrypted templates without a secret key, the templates are secure even if the intruder attacks the communication channel.

Irreversibility Analysis: In SviaB, the CD transmits the encrypted reference templates to CS and smart contract during the enrolment phase. The encrypted reference template is stored in the CS, whereas the hash value of the encrypted reference template is stored in the smart contract. The CD transmits the encrypted probe template to the BC during the authentication phase. The smart contract retrieves the encrypted reference template from the CS and calculates the Manhattan distance between encrypted reference and encrypted probe iris templates. The smart contract sends the \( \epsilon(d) \) to the TA. The TA only can decrypt the result with \( S_k \). As the SviaB uses the Paillier HE scheme [22] to protect the templates, and the security of Paillier scheme relies on solving the DCRA, it is difficult to decrypt the templates by the CS or an imposter without secret key (\( S_k \)). Therefore, SviaB satisfies the irreversibility property.

Revocability: Revocability ensures that a new protected template should be generated by the protection method if the old template is compromised or stolen. In SviaB, revocability can be achieved by re-encrypting the samples in the database with a new key pair (\( P_k', S_k' \)) instead of acquiring the new samples from the users.

Unlinkability: Unlinkability ensures that there will not be any correlation between the protected templates used in different applications. Paillier scheme used in SviaB are based on probabilistic encryption. Because of the randomness involved in Paillier scheme, different ciphertexts can be generated even if the same message is encrypted multiple times with the same key, and there will not exist any similarity between the generated ciphertexts.

Hill Climbing (or) Inverse Biometrics Attack: The TA produces an output as either accept or reject by decrypting the \( \epsilon(R) \) using \( S_k \). Hence, attacks such as hill climbing that are due to the score evolution for different probe signatures described in [45] or the inverse biometrics approaches

| Database         | Template Type                        | Template Size | Storage Cost | EER (%) | Time  |
|------------------|--------------------------------------|---------------|--------------|---------|-------|
| CASIA-V3-Interval| UUT and distance computation in server| 20,480        | 228          | 0.31    | 0.035 |
|                  | CUT and distance computation in server| 128           | 12           | 0.31    | 0.0094|
|                  | CPT and distance computation in server| 128           | 58           | 0.31    | 0.33  |
|                  | CPT and distance computation in blockchain (SviaB) | 128 | - | 0.31 | 1.33 |
| IITD             | UUT and distance computation in server| 20,480        | 413          | 0.86    | 0.054 |
|                  | CUT and distance computation in server| 128           | 88           | 0.88    | 0.094 |
|                  | CPT and distance computation in server| 128           | 88           | 0.86    | 0.33  |
|                  | CPT and distance computation in blockchain (SviaB) | 128 | - | 0.86 | 1.33 |
| SDUMLA-HMT       | UUT and distance computation in server| 20,480        | 207.5        | 0.13    | 0.048 |
|                  | CUT and distance computation in server| 128           | 10.5         | 0.13    | 0.0094|
|                  | CPT and distance computation in server| 128           | 40           | 0.13    | 0.33  |
|                  | CPT and distance computation in blockchain (SviaB) | 128 | - | 0.13 | 1.33 |

Abbreviations: EER, equal error rate; CPT, compressed and protected template; CUT, compressed and unprotected template; UUT, unprotected and uncompressed template.
in [46] are hindered (averted), that is, they lack the vital feedback to recreate a relevant template or biometric sample.

5.3 | Computational analysis of SviaB

The storage cost in terms of space and computational cost in terms of time, cost, and number of operations are discussed in the following sections.

5.3.1 | Computational cost in terms of time and cost

The computation cost and time required to execute operations in a smart contract in units of gas, dollars, and seconds are shown in Table 7. The time required to perform the encryption/decryption of SviaB on different databases is shown in Table 8. The reduced iris code size is the same for all considered databases in SviaB, whereas the iris code size varies for each database in BMIAE [21] to obtain optimal accuracy.
Therefore, in Tables 7 and 8, the computation cost and time, encryption/decryption time is the same for all databases in SviaB. The comparison of time to compute the distance in the BC and the server is illustrated in Table 6. The increase in the computation provides an enhanced functionality (i.e., trust in the computed distance without any third party) to SviaB. We observe from Tables 7 and 8 that SviaB considers less time and cost compared with BMIAE [21].

**FIGURE 6** Genuine and imposter distributions for (a) CASIA-V3-Interval, (b) IITD, and (c) SDUMLA-HMT databases

5.3.2 | Computational cost in terms of number of operations

The privacy of the fused reference and probe iris templates in SviaB is ensured by performing the encryption using $P_k$ before sending to the server/BC. The CD/TA needs to perform only one encryption/decryption in SviaB. The number of exponentiations, multiplications, and encryptions/
decryptions required in SviaB for different databases are shown in Table 9.

- SviaB needs to compute the distance between the probe and corresponding reference template associated with UID only.

5.4 Comparison analysis

The EER comparison of SviaB with state-of-the-art works is shown in Table 10. We can infer that SviaB shows a better EER value when compared with other existing works. The d’ comparison of SviaB with the existing
Different approaches is shown in Table 11. We can infer from Table 11 that the genuine and imposter scores are well separated when compared with other works. SviaB satisfies the properties of template protection schemes and provides trust to the user that the cloud server/BC computes the distance honestly.

\section{Conclusion}

An MIAA system, namely SviaB, is proposed to provide privacy to the iris templates and trust in the comparator result. Paillier HE is used to provide the privacy of the iris templates. In SviaB, a smart contract is used to check the similarity between encrypted reference and probe iris templates. The privacy and the expensive storage limitations of BC for biometrics are addressed in SviaB. Experimental results prove the significance and validity of SviaB.

SviaB can be experimented with on other biometric modalities such as fingerprint, face, finger vein etc. to ensure the privacy of templates and trust in the computed result.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline
\textbf{Method} & \textbf{Database} & \textbf{m} & \textbf{Smart contract deployment} & \textbf{Hash computation and store it} & \textbf{Hash verification} & \textbf{Computation of distance} & \\
& & & \textbf{Cost} & \textbf{ET} & \textbf{Cost} & \textbf{ET} & \textbf{Cost} & \textbf{ET} & \\
\hline
BMIAE [21] & DB1 & 1280 & 673,389 & 1 s & 87741 gas & 1.5 s & 79740 gas & 2 s & 2751053 gas & 6 s & \\
& & & ($0.3326$) & & ($0.4656$) & & ($0.4230$) & & ($1.4594$) & \\
DB2 & 2560 & 673593 gas & 1 s & 1753485 gas & 5 s & 1673154 gas & 4 s & 5885037 gas & 10 s & \\
& & & ($0.3573$) & & ($0.9302$) & & ($0.8876$) & & ($3.1220$) & \\
DB3 & 2560 & 673593 gas & 1 s & 1753485 gas & 5 s & 1673154 gas & 4 s & 5885037 gas & 10 s & \\
& & & ($0.3573$) & & ($0.9302$) & & ($0.8876$) & & ($3.1220$) & \\
SviaB & DB1, DB2, DB3 & 64 & 412072 gas & 1 s & 81049 gas & 1 s & 40934 gas & <1 s & 82956 gas & <1 s & \\
& & & ($0.2956$) & & ($0.0581$) & & ($0.0294$) & & ($0.0595$) & \\
& & & 128 & 412136 gas & 1 s & 84218 gas & 1.5 s & 59081 gas & 1 s & 143226 gas & 1 s & \\
& & & ($0.2957$) & & ($0.0604$) & & ($0.0424$) & & ($0.1028$) & \\
& & & 256 & 412244 gas & 1 s & 136915 gas & 2 s & 96206 gas & 2 s & 269162 gas & 3 s & \\
& & & ($0.2958$) & & ($0.0982$) & & ($0.0609$) & & ($0.1931$) & \\
& & & 512 & 412466 gas & 1 s & 210644 gas & 4 s & 169907 gas & 4 s & 423825 gas & 4.5 s & \\
& & & ($0.2959$) & & ($0.1511$) & & ($0.1219$) & & ($0.3041$) & \\
\hline
\end{tabular}
\caption{Encryption and decryption time (average in seconds); DB1: CASIA-V3-Interval, DB2: IITD, DB3: SDUMLA-HMT}
\end{table}

Abbreviations: BMIAE, BC-based multi-instance iris authentication using additive ElGamal homomorphic encryption; SviaB, secure and verifiable multi-instance iris remote authentication using blockchain.
research must be performed in terms of storing the template information in BC in an optimized manner using a private BC framework or an IPFS.

| Compute Distance | CASIA-V3-Interval | IITD | SDUMLA-HMT |
|------------------|--------------------|------|------------|
| BMIAE [21]       | Enc/Dec            | 1/1  | 1/1        | 1/1        |
|                  | Mult               | 2/1  | 2559       | 5119       |
|                  | Exp                | 0    | 0          | 0          |
| SviaB            | Enc/Dec            | 1/1  | 1/1        | 1/1        |
|                  | Mult               | 2mp1| 255        | 255        |
|                  | Exp                | m    | 128        | 128        |

**Abbreviations:** BMIAE, BC-based multi-instance iris authentication using additive ElGamal homomorphic encryption; SviaB, secure and verifiable multi-instance iris remote authentication using blockchain.

## Table 10 Comparison of SviaB with state-of-the-art approaches (EER in terms of %)

|                     | Dwivedi, R. et al. [47] |         |
|---------------------|-------------------------|---------|
| CASIA-V3-Interval   |                         | 0.43    |
| Punithavathi, P. et al. [49] |                   | 1.9    |
| Soliman, R.F. et al. [50] |                       | 0.63   |
| Zhao, D. et al. [51]   |                         | 1.03   |
| Sadhya, D. et al. [52]  |                         | 0.105  |
| Mahesh, M. et al. [53]   |                         | 0.39   |
| Soliman, R.F et al. [54] |                       | 0.36   |
| Mahesh, M. et al. [21]  |                         | 0.13   |
| SviaB                |                         | 0.31   |
| IITD                 |                         | 0.43   |
| Rathgeb, C., Busch, C. [55] |                   | 3.3    |
| Punithavathi, P. et al. [49] |               | 0.7    |
| Gomez-Barrero, M. et al. [56] |              | 1.4    |
| Mahesh, M. et al. [52]  |                         | 0.88   |
| Mahesh, M. et al. [53]   |                         | 0.88   |
| SviaB                |                         | 0.86   |

**Abbreviations:** EER, equal error rate; SviaB; secure and verifiable multi-instance iris remote authentication using blockchain.

## Table 11 Comparison of SviaB with existing approaches (in terms of separability measure (δ-prime))

|                  | CASIA-V3-Interval | IITD |
|------------------|--------------------|------|
| Sadhya, D. et al. [52]  | 2.39              | 2.92 |
| Walia, G.S. et al. [57] | -                 | 1.9578 |
| Mahesh, M. et al. [21]   | 4.3784            | 4.3786 |
| SviaB                | 4.3257            | 4.3877 |

**Abbreviations:** SviaB; secure and verifiable multi-instance iris remote authentication using blockchain.

## TABLE 9 Computational cost in terms of number of operations (Enc/Dec—encryption/decryption, Mult—multiplications, Exp—exponentiations)

|                      | Compute Distance | CASIA-V3-Interval | IITD | SDUMLA-HMT |
|----------------------|------------------|--------------------|------|------------|
| BMIAE [21]           | Enc/Dec          | 1/1                | 1/1  | 1/1        |
|                      | Mult             | 2F-1              | 2559 | 5119       |
|                      | Exp              | 0                 | 0    | 0          |
| SviaB                | Enc/Dec          | 1/1                | 1/1  | 1/1        |
|                      | Mult             | 2n-1              | 255  | 255        |
|                      | Exp              | m                 | 128  | 128        |

**Abbreviations:** BMIAE, BC-based multi-instance iris authentication using additive ElGamal homomorphic encryption; SviaB, secure and verifiable multi-instance iris remote authentication using blockchain.

## REFERENCES

1. Ross, A.A., Nandakumar, K., Jain, A.K.: Handbook of Multibiometrics: Human Recognition Systems, vol. 6. Springer, Dordrecht. https://cds.cern.ch/record/1990241 (2006)
2. Jain, A.K., Flynn, P., Ross, A.A.: Handbook of biometrics. Springer Verlag, New York (2007)
3. Saini, R., Rana, N.: Comparison of various biometric methods. Int. J. Adv. Sci. Technol. 2(1), 24–30 (2014)
4. Daugman, J.: How Iris recognition works. IEEE Trans. Circ. Syst. Video Technol. 14(1), 21–30 (2004)
5. Prabhakar, S., Pankanti, S., Jain, A.K.: Biometric recognition: security and privacy concerns. IEEE Secur. Priv. 1(2), 33–42 (2003)
6. Maiorana, E., Hine, G.E., Campisi, P.: Hill-climbing attacks on multi-biometrics recognition systems. IEEE Trans. Inf. Forensics Secur. 10(5), 900–915 (2014)
7. Venugopal, S., Savvides, M.: How to generate spoofed irises from an Iris code template. IEEE Trans. Inf. Forensics Secur. 6(2), 385–395 (2011)
8. Jain, A.K., Nandakumar, K., Nagar, A.: Biometric template security. EURASIP J. Adv. Signal Process. 113, 1–17 (2008)
9. Rs 500, 10 Minutes, and You Have Access to Billion Aadhaar Details. https://www.tribuneindia.com/news/nation/rs-500-10-minutes-and-you-have/-access-to-billion-aadhaar-details/523361.html. (2018) Accessed 7 July 2019
10. Rathgeb, C., Uhl, A.: A survey on biometric cryptosystems and cancelable biometrics. EURASIP J. Inf. Secur. 2011(1), 1–25 (2011)
11. Ignatenko, T., Willems, F.M.J.: Information leakage in fuzzy commitment schemes. IEEE Trans. Inf. Forensics Secur. 5(2), 337–348 (2010)
12. Fontaine, C., Galand, F.: A survey of homomorphic encryption for nonspecialists. EURASIP J. Inf. Secur. 15–25 (2007)
13. Upmanyu, M., et al.: Blind authentication: a secure crypto-biometric verification protocol. IEEE Trans. Inf. Forensics Secur. 5(2), 253–268 (2010)
14. Troncoso-Pastoriza, J.R., et al.: Fully private noninteractive face verification. IEEE Trans. Inf. Forensics Secur. 8(7), 1101–1114 (2013)
15. Simoens, K., et al.: A framework for analysing template security and privacy in biometric authentication systems. IEEE Trans. Inf. Forensics Secur. 7(2), 833–841 (2012)
16. Gasti, P., et al.: Secure, fast, and energy-efficient outsourced authentication for smartphones. IEEE Trans. Inf. Forensics Secur. 11(11), 2556–2571 (2016)
17. Abidin, A., et al.: Efficient verifiable computation of XOR for biometric authentication. In: International Conference on Cryptology and Network Security, pp. 284–298, Milan (November 2016)
18. Gomez-Barrero, M., et al.: Multi-biometric template protection based on Homomorphic Encryption. Pattern Recogn. 67, 149–163 (2017)
19. Zhou, K., Ren, J.: PassBio: privacy-preserving user-centric biometric authentication. IEEE Trans. Inform. Forensics Secur. 13(12), 3050–3063 (2018)
20. Hu, S., et al.: Outsourced biometric identification with privacy. IEEE Trans.Inf. Forensics Secur. 13(10), 2448–2463 (2018)
21. Kumar, M.M., Prasad, M.V.N.K., Raju, U.S.N.: BMIAE: blockchain-based multi-instance iris authentication using additive ElGamal homomorphic encryption. IET Biom. 9(4), 165–177 (2020)
22. Pascal, P.: Public-key cryptosystems based on composite degree residuosity classes. In: International Conference on the Theory and Applications of Cryptographic Techniques, pp. 223–238. Prague (1999)

23. Delgado-Mohatar, O., et al.: Blockchain and biometrics: a first look into Opportunities and challenges. In: International Congress on Blockchain and Applications, pp. 169–177. Avila (2019)

24. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science. 313(5786), 504–507 (2006)

25. Osadchy, M., et al.: Sei-fi-a system for secure face identification. In: IEEE Symposium on Security and Privacy, pp. 239–254. Oakland (2010)

26. Rahulamathavan, Y., et al.: Facial expression recognition in the encrypted domain based on local Fisher discriminant analysis. IEEE Trans. Affect. Comput. 4(1), 83–92 (2012)

27. V. Sedvenka, J., et al.: Secure outsourced biometric authentication with performance evaluation on smartphones. IEEE Trans. Inf. Forensics Secur. 10(2), 384–396 (2014)

28. Haghhighat, M., Zonouz, S., Abdel-Mottaleh, M.: CloudID: trustworthy cloud-based and cross-enterprise biometric identification. Expert Syst. Appl. 42(21), 7905–7916 (2015)

29. Xiang, C., et al.: Privacy-preserving face recognition with outsourced computation. Soft Comput. 20(9), 3735–3744 (2016)

30. Hahn, C., Hur, J.: Efficient and privacy-preserving biometric identification in cloud. ICT Express. 2(3), 135–139 (2016)

31. Kumar, S., et al.: Privacy preserving security using biometrics in cloud computing. Multiim. Tools Appl. 77(9), 11017–11039 (2018)

32. Zhu, H., et al.: Efficient and privacy-preserving online fingerprint authentication scheme over outsourced data. IEEE Trans. Cloud Comput. 6(1), 1–11 (2018)

33. Barni, M., et al.: SEMBA: secure multi-biometric authentication. IET Biometrics. 8(6), 411–421 (2019)

34. Guo, S., Xiang, T., Li, X.: Towards efficient privacy-preserving face recognition in the cloud. Signal Process. 164, 320–328 (2019)

35. Delgado-Mohatar, O., et al.: Biometric template storage with blockchain: a first look into cost and performance tradeoffs. In: The IEEE Conference on Computer Vision and Pattern Recognition Workshops. Long Beach (2019)

36. Delgado-Mohatar, O., et al.: Blockchain meets Biometrics: Concepts, Application to Template Protection, and Trends. arXiv preprint arXiv:2003.09262 (2020)

37. Mohsin, A.H., et al.: Based blockchain-PSO-AES techniques in finger vein biometrics: A novel verification secure framework for patient authentication. Computer Standards & Interfaces. 66, 1–14 (2019). http://dx.doi.org/10.1016/j.csi.2019.04.002

38. Rathgeb, C., et al.: Design decisions for an iris recognition SDK. In: Bowyer, K., Burge, M.J. (eds.) Handbook of Iris Recognition, second edition, Advances in Computer Vision and Pattern Recognition (2016)

39. Wang, Y., Yao, H., Zhao, S.: Autoencoder based dimensionality reduction. Neurocomputing. 184, 232–242 (2016)

40. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behaviour of distance metrics in high dimensional space. In: International Conference on Database Theory, pp. 420–434. Berlin (2001)

41. ISO/IEC 24745:2011 Information Technology-Security Techniques-Biometric Information Protection. https://www.iso.org/standard/52946.html (2011). Accessed 01 August 2019

42. Yin, Y., Liu, L., Sun, X.: SDMLA-HMT: A Multimodal Biometric Database. In: Chinese Conference on Biometric Recognition, pp. 260–268. Beijing (2011)

43. Kumar, A., Passi, A.: Comparison and combination of iris matchers for reliable personal authentication. Pattern Recogn. 43(10), 1016–1026 (2010)

44. Biometrics Ideal Test (CASIA-V3-Interval). http://biometrics.idealtest.org/dbDetailForUser.do?id=4, Accessed 30 December 2018

45. Martinez-Diaz, M., et al.: Hill-climbing and brute-force attacks on biometric systems: a case study in match-on-card fingerprint verification. In: Proceedings of the 40th Annual 2006 International Carnahan Conference on Security Technology, pp. 151–159. Lexington (2006)

46. Hadid, A., et al.: Biometrics Systems under Spoofing Attacks: an evaluation methodology and lessons learnt. IEEE Signal Process. Mag. 32(5), 20–30 (2015)

47. Dwivedi, R., et al.: A privacy-preserving cancellable iris template generation scheme using decimal encoding and look-up table mapping. Comput. Secur. 65, 373–386

48. Lai, Y.L., et al.: Cancellable iris template generation based on Indexing-First-One hashing. Pattern Recogn. 64, 105–117

49. Punithavathi, P., Geetha, S., Sasikala, S.: Generation of cancellable iris template using bi-level transformation. In: Proceedings of the 6th International Conference on Bioinformatics and Biomedical Science, pp. 94–100. Singapore (2017)

50. Soliman, R.F., Amin, M., El-Samie, F.E.A.: A double random phase encoding approach for cancellable iris recognition. Opt. Quant. Electron. 50(6), 326–338 (2018)

51. Zhao, D., et al.: Iris template protection based on randomized response technique and aggregated block information. In: 2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE), pp. 248–258, USA (2018)

52. Sadllya, D., Raman, B.: Generation of cancellable iris templates via randomized bit sampling. IEEE Trans. Inform. Forensics Secur. 14(11), 2972–2986 (2019)

53. Morampudi, M.K., Prasad, M.V., Raju, U.S.N.: Privacy-preserving iris authentication using fully homomorphic encryption. Multimed. Tool. Appl. 79(3), 19215–19237 (2020)

54. Soliman, R.F., Amin, M., Abd El-Samie, F.E.: Cancellable Iris recognition system based on comb filter. Multimed. Tool. Appl. 79(3), 2521–2541 (2020)

55. Rathgeb, C., Busch, C.: Cancellable multi-biometrics: mixing iris-codes based on adaptive bloom filters. Comput. Secur. 42, 1–12 (2014)

56. Gomez-Barreno, M., et al.: Multi-biometric template protection based on bloom filters. Inf. Fusion. 42, 37–50 (2018)

57. Walia, G.S., et al.: Secure multimodal biometric system based on diffused graphs and optimal score fusion. IET Biom. 8(4), 231–242 (2019)

How to cite this article: Morampudi MK, Prasad MVNK, Raju Undi SN. SviaB: Secure and verifiable multi-instance iris remote authentication using blockchain. IET Biometrics. 2021;1–16. https://doi.org/10.1049/bme2.12042

APPENDIX

Smart contract source code.

```solidity
pragma solidity ^0.5.0;

contract bioauth{

    event getRes(uint[F] r1, uint[F] r2);

    struct template {
        bytes32 C1hash;
        bytes32 C2hash;
    }

    mapping(uint => >template) public templates;

    uint public numOfC1Templates = 0;
    uint public numOfC2Templates = 0;

    function getDataC1(uint[] memory a, uint[] memory b) public {
        temp[templates][numOfC1Templates].C1hash = keccak256(ab.encodePacked(a));
        temp[templates][numOfC2Templates].C2hash = keccak256(ab.encodePacked(b));
        numOfC2Templates++;  
    }
```
function compute(uint[] memory c1, uint[] memory c2, uint[] memory c3, uint[] memory c4, uint id) public {
    require(keccak256(abi.encodePacked(c1)) =
        = templates[id].C1hash);
    require(keccak256(abi.encodePacked(c2)) =
        = templates[id].C2hash);
    uint alength = c1.length;
    uint clength = c3.length;
    uint[F] memory result1;
    uint[F] memory result2;
    for(uint i = 0; i < alength; i++) {
        result1[i] = c1[i] * c3[i];
    }
    for(uint i = 0; i < clength; i++) {
        result2[i] = c2[i] * c4[i];
    }
    emit getResult(result1, result2);
}
