# A Novel NIZK-based Privacy Preserving Biometric Identification Scheme for Internet of Things

Lin You, Qiang Zhu, Gengran Hu

***Abstract*—With the popularity of biometric-based identity authentication in the field of the Internet of Things, more and more attention has been paid to the privacy protection of biometric data. Gunasinghe et al. presented the PrivBioMTAuth which is the first authentication solution from mobile phones to protect user's privacy by performing interactive zero-knowledge proof. However, PrivBioMTAuth still requires considerable storage overhead and communication overhead during the registration phase. Meanwhile, the user's biometric images and password need to be revealed to the identity provider. In this paper, we present an authentication solution for Internet of Things with fully succinct verification, significantly lower storage overhead and communication overhead. Different from PrivBioMTAuth, we rely on the non-interactive zero knowledge arguments given in Groth's work to reduce the proof size and simplify the verification complexity. In addition, we focus on multi-exponentiation arguments based on Bayer et al.'s work to ensure the truth of the operation results provided by the identity provider.**

***Index Terms*—Biometrics, Privacy Protection, NIZK, Identification, IoT.**

## I. Introduction

WITH the miniaturization of Internet of Things (IoT) devices and the improvement of computing power, the development of user identity authentication has been promoted to a certain extent for Internet of Things. A growing number of online service providers prefer to adopt biometric-based remote identity authentication scheme for Internet of Things, such as e-commerce organizations [1], online banking institutions [2] and stock finance companies. However, the proposed biometric-based authentication systems for Internet of Things are usually based on plaintext matching and it may course the problem of user privacy leak. In order to address the leakage of user's sensitive information about the biometrics, many identity authentication schemes with privacy protection have been proposed [3]–[9].

In traditional remote identity authentication online services, the users need to register their biometrics with the service providers before performing identity authentication protocol. Typically, the service providers offer a variety of biometric-based authentication mechanisms, such as fingerprints, faces, voiceprints, etc. Once the biometric-based identity authentication system receives the biometric images sent by the user, it will use a specific extraction mechanism to extract the user's biometric information which can be referred to as a biometric template. Furthermore, a set of binary strings are randomly selected as labels for the biometric templates and stored in the biometric template database in plaintext. When the user is required for identity authentication, his/her biometric is extracted and is matched with the template stored in the database. The authentication will succeed if the match is passed, otherwise it will fail. Privacy protection of biometric templates and the security of the authentication process are critical for users. If the biometric template database or the biometric template provided at authentication are stolen, the pretender may use the user's identity to engage in illegal activities unless the service providers adopt a revocable biometric template technology [11]. Moreover, service providers may make use of the user's biometric data for big data analysis to achieve other purposes, such as AI face swap, criminal behavior. Therefore, it is very important to protect the user's biometric data in the biometric-based identity authentication systems for Internet of Things.

A potential method for solving this issue is to encrypt the biometric templates and perform authentication of the user's identity in the ciphertext space. Nevertheless, the users need to register their identity information with a trusted authority, which can be referred to as an identity provider (IDP). Once recieving the identity authentication messages from the users, the SP needs to request the IDP for the user's encrypted biometric template. In such identity authentication architecture, the users are not required to register and disclose their biometric information with the SP. This method can realize the privacy protection of user's sensitive information without exposing user's biometric data. However, such an identity authentication solution may have the risk of indirectly revealing user's privacy. Because the IDP is involved in each transaction, it can infer sensitive information, such as user's transaction patterns with different service providers [5].

User-centric identity management architecture can tackle the above issues and does not involve the IDP during the authentication phase. In such identity authentication architecture, after registering biometric data with the IDP, the users can be authenticated directly by the service providers without involving a trusted third party. In the PrivBioMTAuth [5], when registering an identity with the IDP by biometric images and a password from mobile phones, the user is given an identity token (IDT) and some secure artifacts which can be used to regenerate the secrets during the authentication phase. It can be achieved with the use of interactive zero knowledge proofs [12] and cryptographic commitments [13]. Unfortunately, there are some drawbacks to this solution. In practical applications, considerable storage overhead and communication overhead are required in the registeration phase. Besides, the user's

This research is partially supported by the National Natural Science Foundation of China (No. 61772166) and the Key Program of the Natural Science Foundation of Zhejiang Province of China (No. LZ17F020002).

Lin You, Qiang Zhu and Gengran Hu are with the School of Cyberspace Security, Hangzhou Dianzi University, Hangzhou, 310018, China (e-mail: mryoulin@gmail.com; drqiangzhu@gmail.com; grhu@hdu.edu.cn)

biometric images and password are revealed to the IDP in plaintext which still have the risk of leaking user sensitive information.

To overcome the drawbacks, we propose a novel NIZK-based privacy preserving biometric identification scheme for Internet of Things. The main contributions of our work can be summarized as follows:

(1) We provide a efficient, privacy-preserving and user centric authentication protocol for online identity authentication of Internet of things users. Service providers can verify user's identity without involving the user's biometric information. Moreover, the SP cannot derive any useful sensitive information from the user's identity proof $\pi$ besides the truth of the user identity.

(2) The authentication scheme has the relatively low communication overhead and computation complexity during the authentication phase. Furthermore, it avoids the considerable communication overhead and local storage overhead of cryptographic authentication artifacts during the registration phase.

(3) We give the security analysis of our identity authentication scheme and evaluate its efficiency. The result shows that our solution is efficient and privacy-preserving.

The rest of this paper is organized as follows: In Section II, we introduce the main concepts used in our identity authentication scheme. In Section III, we introduce the system model and security model. We present the details of our identity authentication scheme in Section IV. In Section V and Section VI, the security analysis and performance evaluation are presented respectively. Finally, we discuss related work in Section VII and draw the conclusions in Section VIII.

## II. Background

### A. Bilinear Pairing

For three cyclic groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ (all of which we shall write multiplicatively) of the same prime order $p$, a bilinear pairing $e$ is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

- Bilinearity: For $g \in \mathbb{G}_1$, $h \in \mathbb{G}_2$, and $a, b \in \mathbb{Z}_p$, we have $e(g^a, h^b) = e(g, h)^{a \cdot b}$
- Non-degeneracy: $e(g, h) \neq 1_{\mathbb{G}_T}$
- Computability: $e(p, q)$ can be computed efficiently for all $p \in \mathbb{G}_1$, $q \in \mathbb{G}_2$.

### B. Quadratic Arithmetic Program(QAP)

Here, we define QAP in terms of Groth [14] and Gennaro et al. [16] in a relation $R$. The QAP have the following description:

$$R = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, k, \{u_i(X), v_i(X), w_i(X)\}_{i=0}^{m}, t(X)).$$

The bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ describes a finite field $\mathbb{F}$, $1 \leq k \leq m$, and the polynomials $\{u_i(X), v_i(X), w_i(X)\}_{i=0}^{m}$ represents the set of three linearly independent polynomials defined in QAP, as shown below:

$$\sum_{i=0}^{m} a_i u_i(X) \cdot \sum_{i=0}^{m} a_i v_i(X) \equiv \sum_{i=0}^{m} a_i w_i(X) \bmod t(X)$$

where $u_i(X), v_i(X)$ and $w_i(X)$ have strictly lower degree than the degree of $t(X)$. A quadratic arithmetic program with this property defines a binary relation $R$ as follows ,where we define $a_0 = 1$,

$$R = \left\{ (\phi, w) \;\middle|\; \begin{array}{l} \phi = (a_1, a_2, \ldots, a_k) \in \mathbb{F}^k \\ w = (a_{k+1}, a_{k+2}, \ldots, a_m) \in \mathbb{F}^{m-k} \\ \\ \sum_{i=0}^{m} a_i u_i(X) \cdot \sum_{i=0}^{m} a_i v_i(X) \equiv \sum_{i=0}^{m} a_i w_i(X) \bmod t(X) \end{array} \right\}.$$

We say $\mathcal{R}$ is the relationship generator of QAP if it generates relations $R$ of the form given above with fields of size larger than $2^{\lambda-1}$.

### C. Eigenfaces face recognition algorithm

Eigenface algorithm [17] is a traditional face recognition scheme designed based on principal component analysis (PCA). It projects the original data from $n$-dimensional to $k$-dimensional subspace, where $k < n$. The eigenface face recognition algorithm includes two stages: *Computing the Eigenfaces Subspace* and *Projecting an Image Onto Eigefaces Subspace*. In what follows, we discuss how the specific steps of these two stages make the performed.

1) *Computing the Eigefaces Subspace*: First, we need to randomly select a face dataset. Each face image in the selected training dataset is represented as a $p \times q$ matrix of pixel values. Then the matrix corresponding to each image will be converted into a vector of $p * q$ columns. Let $X = \{x_1, x_2, \ldots, x_N\}$ be a matrix containing $N$ vectors which represent the corresponding face images in the face dataset.
   a) Compute the mean: $\psi = \frac{1}{n} \sum_{i=1}^{n} x_i$
   b) Compute the covariance matrix: $S = \frac{1}{n} \sum_{i=1}^{n} (x_i - \psi)(x_i - \psi)^T$
   c) Compute the eigen vectors $v_i$ and eigen values $\lambda_i$. Their relationship is as follows: $v_i = \lambda_i v_i$
   d) Normalize the eigen vectors.
   e) Arrange these eigen values in descending order, and select the $k$ eigen vectors corresponding to the largest $k$ eigen values. The eigenface subspace composed of these $k$ eigen vectors is referred to as $W$.
2) *Projecting an Image Onto Eigenfaces Subspace*: Given a face image $I$ whose features need to be calculated, the features of the face image are extracted through the eigen face subspace $W$ and the mean image $X_{mean} (= \psi)$. The face features are extracted according to the following steps.
   a) Normalize the image: $I_N = \frac{I}{\|\mathbf{I}\|}$
   b) Substract the mean image of the training set: $I_S = I_N - X_{mean}$
   c) Project $I_S$ onto $W$: $F_I = W^T I_S$

   $F_I$ is the projection of image $I$ on eigenfaces subspace $W$. It can be seen as a set of features of an image $I$.

### D. Paillier encryption

The Paillier encryption algorithm [18] has homomorphic encryption properties whose security is based on the hardness of solving discrete logarithms. The algorithm involves an

encryption party and decryption party, and can be described as the following three phases.

*Setup*($Gen(\ell)$): Given a security parameter $\ell \in \mathbb{Z}^+$, the encryptor randomly generates two independent $\ell$-bit prime numbers $p$ and $q$, and satisfy $\gcd(pq, (p-1)(q-1)) = 1$. Then the encryptor computes $n = pq$, $\lambda = lcm(p-1, q-1)$ and randomly chooses $g \in \mathbb{Z}^*_{n^2}$. The public key $pk$, $(n, g)$ is published while $\lambda$ is saved as a private key $sk$.

*Encryption*: The encryptor creates the ciphertext of $M \in \mathbb{Z}_n$ by choose $r \in \mathbb{Z}^*_n$ at random and computing:

$$c = \varepsilon_{pk}(M; r) = g^M r^n \bmod n^2.$$

*Decryption*: To decrypt the ciphertext $c$, the decryptor can define $L(x) = \frac{x-1}{n}$. Then plaintext can be calculated by

$$M = \zeta_{sk}(c) = \frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n.$$

### E. Pedersen Commitment

The Pederson commitment [19] is a security commitment scheme and it has perfectly hiding and computational binding. It's perfectly hiding does not depend on any difficult assumptions. It's computational binding relies on the Discrete Logarithm Assumption (DLA). It's structure is divided into three parts:

*Setup:* The committer chooses two large prime numbers $p$ and $q$, such that $q$ divides $p-1$. $\mathbb{G}_q$ is the interger group of order $q$, and is sub group of $\mathbb{Z}_p$, where $\mathbb{Z}_p$ represents the group of integers of order $p$. Then, the committer selects elements $g$ which is a generator of $\mathbb{G}_q$ and $h$ which an element of $\mathbb{G}_q$, such that given $g$ and $h$ it is computationally hard to find $\log_g h$, and publishes public parameter $ck = (p, q, g, h)$.

*Commit:* The committer picks $r \in \mathbb{Z}_q$ randomly, and creates a commitment to element $x \in \mathbb{Z}_q$ by computing

$$C = com_{ck}(x; r) = g^x h^r \bmod p.$$

*Open:* To open the commitment, the committer sends $(x, r)$ to the verifier. The verifier checks whether $C = g^x h^r$ to verify the truth of the commitment. Accept if they are equal, otherwise reject the commitment.

## III. SYSTEM MODEL AND SECURITY MODEL

### A. Notations

For the sake of expression, we show some notations and their descriptions in Table I.

TABLE I: Notations and their descriptions

| Notations | Descriptions |
|---|---|
| $\ell$, $\lambda$ | security parameter |
| $p$, $q$ | a large prime |
| $\mathbb{G}_1$, $\mathbb{G}_2$ | multiplicative cyclic groups with order $p$ |
| $e$ | a bilinear pairing map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ |
| $g$, $h$ | a generator of group $\mathbb{G}_1$, $\mathbb{G}_2$ |
| $[a]_1$, $[b]_2$ | $g^a$, $h^b$ |
| $[c]_T$ | $e(g, h)^c$ |
| $k \cdot [a]_i$ | $[k \cdot a]_i$ |
| $[a]_1 \cdot [b]_2$ | $[ab]_T$ |
| $[a]_i + [b]_i$ | $[a + b]_i$ |
| $sk$, $pk$ | the private key and public key of the user |
| $ck$ | the commitment key of the user |
| $\varepsilon_{pk}()$, $\zeta_{sk}()$ | the encryption and decryption functions of the user |
| $com_{ck}()$ | a commitment scheme of the user |
| $\boldsymbol{xy}$ | $(x_1 y_1, \ldots, x_n y_n)$ |
| $xy$ | $(xy_1, \ldots, xy_n)$ |
| $\boldsymbol{x \cdot y}$ | $\sum_{i=1}^n x_i y_i$ |
| $\boldsymbol{C^a}$ | $\prod_{i=1}^n C_i^{a_i}$ |
| $\boldsymbol{C^e}$ | $(C_1^e, C_2^e, \ldots, C_k^e)$ |
| $H()$ | a cryptographic hash function |
| $\Phi()$ | a shift function which converts value to field element |
| $\mathbb{H}$ | ciphertext space |
| $\mathbb{G}$ | commitment space |
| $\mathbb{Z}_p^*$ | a prime field with non-zero elements |
| $\boldsymbol{\Gamma}$ | the biometric image of the user |
| $\boldsymbol{\Gamma}'$ | the encrypted biometric image of the user |
| $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k)$ | the eigen vectors calculated by the IDP |
| $(\boldsymbol{u}'_1, \ldots, \boldsymbol{u}'_k)$ | the eigen vectors processed by shift function |
| $(\theta'_1, \ldots, \theta'_k)$ | the transformation result of projecting the user's encrypted biometric image onto the eigen faces subspace |

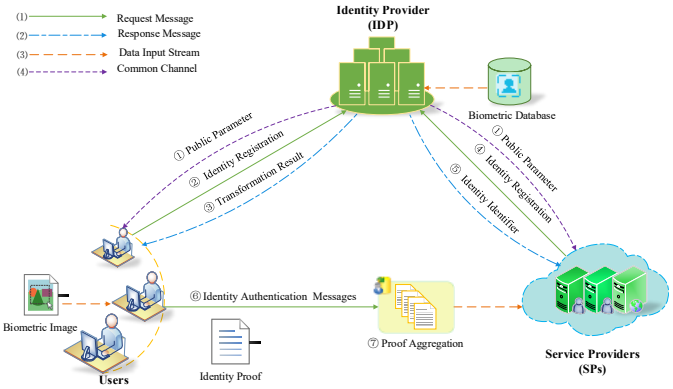


Fig. 1: The system model

### B. System Model

Our system model involves three entities: user, identity provider (IDP) and service provider (SP), as shown in Fig.1.

(1) User: After biometric images collected by the IoT devices, the user encrypts the biometric image and the password, picks a random value $e$, computes the digital signature of their merged hash value, and sends all values, public key $pk$ and hash function to the IDP. When receiving the transformation result of projecting the user's encrypted biometric image onto the eigen faces subspace $W$ generated by IDP, the user will verify whether the operation results generated by the IDP is calculated based on the original biometric image sent by himself. Then, the user decrypts the obtained transformation result and calculates exclusive witness related to himself. Finally, the user constructs an identity proof $\pi$ based on witness, common reference string $\sigma$ and statement $\phi$ to proof his identity quickly.

(2) Identity provider: The IDP is a third-party service agency. Usually, it can be seen as a semi-honest entity. For example, the IDP honestly handles SP's identity registration request at the same time and provides the SP with generic identity identifier $\phi$. Moreover, the system initialization is often implemented by the IDP by generating and sending public parameters to registered service providers and users respectively. Unfortunately, the IDP may not use the encrypted biometric image sent by the user for calculations or try to

analyze the encrypted biometric and the encrypted password entered by the user to obtain the user's sensitive information.

(3) Service provider: The SP is a general online service provider which provides user identity authentication services using the identity proof $\pi$ sent by the user, common reference string $\sigma$ and generic identity identifier $\phi$. Before providing user identity authentication service, the SP should pre-register in IDP through its public key, name, e-mail and other identity information. Then the SP will receive the generic identity identifier $\phi$ provided by the IDP. After receiving an authentication message containing identity proof $\pi$ generated by the user, the SP uses common reference string $\sigma$ and generic identity identifier $\phi$ to compute a single pairing product equations to judge whether user's identity is valid.

### C. Definition

*Definition 1*: A novel NIZK-based privacy preserving biometric identification scheme for Internet of Things involves the following six algorithms: *Setup*, *UserReg*, *CalcEigen*, *EigenVerify*, *ProofGen*, *ProofVerify*. The specific details of these algorithms are as follows:

1) **Setup**$(1^\lambda, \Omega)$ is a setup algorithm run by the IDP. It takes as input a security parameter $\lambda$ and the biometric dataset $\Omega$. It outputs quadratic arithmetic program $R$, common reference string $\sigma$, a set of committed values $\boldsymbol{c}_A$ and the SP's identity identifier $\phi$.
2) **UserReg**$(1^\ell, \mathbf{\Gamma}, pw)$ is a user identity registration algorithm run by the user. It take as input the user's biometric image, password $pw$ and a security parameter $\ell$. It outputs parameter $v$, the digital signature $\varepsilon_{sk}(v'; \varsigma)$ of the hash value of $v$, the user's public key $pk$ and hash function $H()$, where $v$ is the combination of encrypted biometric image, the encrypted password $pw$ and random values $e$, and $v'$ is the hash value of $v$.
3) **CalcEigen**$(v'', pk, H())$ is a transformation result generation algorithm run by the IDP. It take as input parameter $v''$, the user's public key $pk$ and hash function $H()$. It outputs a set of vectors $\boldsymbol{c}_b$, $\boldsymbol{E}$, parameter $C$, a transformation result $(\theta'_1, \ldots, \theta'_k)$ and statement $\phi$.
4) **EigenVerify**$(x, C, \boldsymbol{E}, \boldsymbol{c}_A, \boldsymbol{c}_b)$ is a transformation result verification algorithm run by the user and the IDP. It take as input random challenge value $x$, a set of vectors $\boldsymbol{E}$, $\boldsymbol{c}_b$, a set of committed values $\boldsymbol{c}_A$ and parameter $C$. The user can verify the truth of the operation results generated by the IDP.
5) **ProofGen**$(R, \sigma, \theta'_1, \ldots, \theta'_k, \phi)$ is an identity proof generation algorithm run by the user. It takes as input quadratic arithmetic program $R$, common reference string $\sigma$, the transformation result $(\theta'_1, \ldots, \theta'_k)$ and statement $\phi$. It outputs an identity proof $\pi$ that can be used to demonstrate user's identity.
6) **ProofVerify**$(R, \sigma, \phi, \pi)$ is a user identity verification algorithm run by the SP. It takes as input quadratic arithmetic program $R$, common reference string $\sigma$, generic identity identifier $\phi$ and the user's identity proof $\pi$. The SP can verify the truth of the user identity.

### D. Security requirements

In our identity authentication scheme, we consider the IDP is a semi-honest entity. The IDP honestly publishes system public parameters, handles SP's identity registration request at the same time and provides the SP with generic identity identifier $\phi$. Whereas, it could substitute some or all of the ciphertext of the encrypted biometric image without being detected by user, and may attempt to analyze the user's biometric information. Nevertheless, the IDP can not obtain the transformation result of the user's original biometric image projected on the eigen faces subspace $W$ and the intermediate value of the operation process. Moreover, the SP can be seen as a untrusted entity. After receiving an authentication message from the user, it may be curious about the sensitive information about the biometrics in the identity proof generated by the user. Therefore, the user's original biometric image and password should not be exposed to the IDP and the SP. In order to guarantee the security of user's sensitive information about the biometrics, the definition of security that meets the above requirements is as follows:

*Definition 2*: We say a novel NIZK-based privacy preserving biometric identification scheme for Internet of Things has the property of computational soundness if it satisfies the following conditions: whenever a non-uniform polynomial time adversary $\mathcal{A}$ can generate a valid identity proof $\pi$ to pass the verification of the challenger $\mathcal{C}$ with non-negligible probability, there exists a non-uniform polynomial time knowledge extractor $\mathcal{X}_\mathcal{A}$ that can compute a witness, which gets full access to the adversarys state, including any random coins.

*Definition 3*: We say a novel NIZK-based privacy preserving biometric identification scheme for Internet of Things has the property of perfect zero-knowledge if the SP cannot derive any useful sensitive information from the user's identity proof $\pi$ besides the truth of the user identity and the IDP only calculates the encrypted biometric image, and cannot obtain the user's sensitive information about the biometrics from the operation results.

### E. Design Goal

For the sake of meet the aforementioned system model and security requirements, our solution should simultaneously fulfill the following security and performance goals:

1) Perfect completeness:
   a) Registration information completeness: to ensure that once the user sends identity registration request to the IDP, the request message must be able to pass the verification of the IDP.
   b) Sensitive information completeness: to ensure that once the IDP uses the encrypted biometric image and the encrypted password sent by the user to calculate the transformation result of projecting the user's encrypted biometric image onto the eigen faces subspace $W$, it must be able to pass the verification of the user and make the user believe in the truth of the operation results.
   c) Authentication completeness: to ensure that once the user has obtained the transformation result generated by

the IDP, the identity proof $\pi$ constructed by the user must pass the verification of the SP.

2) Computational knowledge soundness: to ensure that if the user does not register his identity with the IDP, he cannot pass the SP's verification and if the IDP deceives the user and does not use the user's biometric image for calculations, it cannot pass the user's verification.
3) Perfect zero-knowledge: to ensure that if the user sends identity authentication message to the SP, the SP cannot obtain any useful sensitive information from the user's identity proof $\pi$ besides the truth of the user identity. The IDP only calculates the encrypted biometric image, and cannot obtain the user's sensitive information about the biometrics from the operation results.
4) Low communication overhead and low computation complexity: to ensure that if the user sends identity authentication message to the SP, the size of an identity proof generated by the user is only 3 group elements. Meanwhile, the SP's verification consists of checking a single pairing product equations using 3 pairings in total. Our authentication scheme should be able to efficiently handle large-scale authentication messages using relatively low communication overhead and computation complexity.

## IV. Identity Authentication Scheme

### A. An Overview

In order to achieve privacy preserving biometric-based identity authentication for Internet of Things, we mainly take into account the idea of zero-knowledge proof to realize the protection of user sensitive information. Nevertheless, it is infeasible to directly use zero-knowledge proof during the identity authentication phase. Firstly, we know that the prover can generate a proof to convince the verifier that the validity of the assertion, according to the properties of zero-knowledge proof. However, there is no combination directly of zero-knowledge proof and identity authentication under the condition of protecting the user's sensitive information about the biometrics. To address this issue, Gunasinghe et al. used cryptographic commitment schemes and interactive zero-knowledge proofs to implement privacy protection of the users' biometrics [5]. Nevertheless, this solution requires storing the trained classifier on the user's mobile phones, which will inevitably occur considerable storage and communication overhead during the registration phase. Simultaneously, the interaction process between the user and the service provider will inevitably lead to man-in-the-middle attacks. Secondly, the interactive zero-knowledge proof only convince the original verifier that the truth of the user identity. In order to convince multiple verifiers that the truth of the user's identity and prevent collusion between the user and the verifier, the user needs to interact with each verifier, which will produce excessive verification cost and cause low efficiency. Thirdly, the user's original biometric images and password are revealed in the process of registering their identity with the IDP. Nevertheless, the IDP may use the user's sensitive information about the biometrics for data analysis, or the identity token returned to the user is not generated based on the biometric images sent by the user.

In order to tackle the aforementioned issues, we introduce pairing-based non-interactive zero knowledge (NIZK) arguments [14] during the authentication phase and construct a batch authentication method for identity proofs based on NIZK arguments. In addition, we optimize the multi-exponentiation arguments [15], which makes it convenient for user to verify that the truth of the operation results provided by the IDP with lower communication overhead. Our solution has achieved the following goals in total: i) it avoids divulging the user's biometrics during the authentication phase and registration phase; ii) it avoids storing the user's biometrics in the IDP and the SP; iii) it does not involve the IDP during the authentication phase; iv) the identity proof size is only 3 group elements, and the SP only need to check a single pairing equations using 3 pairings in total to verify the truth of the user identity; v) it ensures that the operation results generated by the IDP based on the encrypted biometric image and encrypted password provided by the user.

In our identity authentication scheme, the IDP randomly selects the data in a face dataset as the biometric training set, and uses the eigenface algorithm to calculate the mean $\psi$ of the biometric images and the $k$ eigen vectors corresponding to the $k$ larger eigen values. Then the IDP uses pederson commitment scheme to commit $k$ eigen vectors which processed by the shift function, and calculates the statement $\phi$, which can be seen as the identity identifier for the SP. After receiving the identity registration request sent by the SP, the IDP sends identity identifier to him in a secret way. In addition, the user sends random value, the encrypted biometric image, the encrypted password, the digital signature of their merged hash value, public key and hash function to the IDP to ensure the validity of the request message. When above message sent by the user are valid, the IDP will generate a special set of vectors which can be used to calculate the transformation result of projecting the user's encrypted biometric image onto the eigen faces subspace $W$. Lastly, the IDP uses multi-exponentiation arguments to make the user believe that the operation results calculated based on the user's biometric image. After receiving the transformation result generated by the IDP, the user decrypts it and calculates the witness related to himself. Before allowing the user to execute transactions, the SP can verify the truth of the user identity using the generic identity identifier, where the verification of the SP only requires checking a single pairing product equation using 3 pairings in total. Further, after receiving authentication messages from multiple users, the SP can aggregate the identity proof sent by users into 4 elements, and then verify the 4 elements once to check whether there are illegal users forged identities. The details description will be shown in the following subsection.

### B. Description of the Proposed Scheme

In our identity authentication scheme, the user's biometric image can be represented by a row vector $\mathbf{\Gamma} = (x_1, \ldots, x_N)$, where $x_i \in \mathbb{Z}_p^*$ denotes the $i$-th pixel value of the user's biometric image. Before sending the identity registration request to the IDP, the user usually encrypts biometric image

$\mathbf{\Gamma}$ to obtain a set of ciphertext $\mathbf{\Gamma}' = (x'_1, \ldots, x'_N)$, where $x'_i = \varepsilon_{pk}(x_i; s_i)$, $1 \le i \le N$, and $s_i$ is a random value selected by the user, belonging to $\mathbb{Z}_p^*$. Meanwhile, the $k$ eigen vectors corresponding to the largest $k$ eigen values calculated by the IDP can be expressed as $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k)$, where $\boldsymbol{u}_i$ is an $N$-dimensional column vector. In order to calculate the transformation result of projecting the user's encrypted biometric image onto the eigen faces subspace $W$ and facilitate users to verify the operation results generated by IDP, the IDP needs to calculate the eigen vectors $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k)$ through a shift function $\Phi(x)$ to obtain a new set of vectors $(\boldsymbol{u}'_1, \ldots, \boldsymbol{u}'_k)$, where $\boldsymbol{u}'_{ij} \in \mathbb{Z}_p^*$, $1 \le i \le k$, $1 \le j \le N$. By adding all the elements in the each vector after shift processing, $k$ elements $(a_1, \ldots, a_k)$ are obtained, where $a_i \in \mathbb{Z}_p^*$, $1 \le i \le k$. In binary relation $R$, we assign the values of $k$ variables $(a_1, \ldots, a_k)$ to the statement $\phi$. Furthermore, the committed value of a set of vectors $(\boldsymbol{u}'_1, \ldots, \boldsymbol{u}'_k)$ can be expressed as $(com_{ck}(\boldsymbol{u}'_1; r_1), \ldots, com_{ck}(\boldsymbol{u}'_k; r_k))$. It can be used by the user to judge whether the IDP has performed the correct calculations, where $r_i$ is a random value selected by the user, belonging to $\mathbb{Z}_p^*$, $1 \le i \le k$. To simplify notation, we write $\boldsymbol{c}_A = com_{ck}(A; \boldsymbol{r})$ for the vector $(c_{A_1}, \ldots, c_{A_k}) = (com_{ck}(\boldsymbol{u}'_1; r_1), \ldots, com_{ck}(\boldsymbol{u}'_k; r_k))$ when $A$ is a matrix with column vectors $(\boldsymbol{u}'_1, \ldots, \boldsymbol{u}'_k)$.

The details of our authentication scheme are as follows.

1) **Setup**$(1^\lambda, \Omega)$
   a) The IDP randomly selects the data in a face dataset $\Omega$ as the biometric training set, and uses the eigenface algorithm to calculate the mean $\psi$ of the biometric images and the $k$ eigen vectors $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k)$ corresponding to the $k$ larger eigen values.
   b) The IDP uses a shift function $\Phi(x)$ to convert all the values in the eigen vectors into finite field to get a set of vectors $(\boldsymbol{u}'_1, \ldots, \boldsymbol{u}'_k)$, where $\boldsymbol{u}'_{ij} \in \mathbb{Z}_p^*$, $1 \le i \le k$, $1 \le j \le N$. Then, the IDP uses pederson commitment scheme to commit this set of vectors to get a set of committed value $\boldsymbol{c}_A = (c_{A_1}, \ldots, c_{A_k})$, where $c_{A_i} \in \mathbb{G}$, $1 \le i \le k$. By adding all the elements in the each vector after shift processing, $k$ elements $(a_1, \ldots, a_k)$ are obtained, where $a_i \in \mathbb{Z}_p^*$, $1 \le i \le k$. When receiving the identity registration request sent by the SP, the IDP will send the generic identity identifier $\phi = (a_1, \ldots, a_k)$ to him in a secret way.
   c) The IDP constructs a relation generator $\mathcal{R}$ that given a security $\lambda$ in unary return relations of the form $R = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h, k, \{u_i(x), v_i(x), \omega_i(x)\}_{i=0}^m, t(x))$ with $|p| = \lambda$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are multiplicative cyclic groups of prime order $p$, a generator $g$ of $\mathbb{G}_1$, a generator $h$ of $\mathbb{G}_2$, a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.
   d) The IDP picks elements $\alpha, \beta, \gamma, \delta, x \leftarrow \mathbb{Z}_p^*$, defines $\tau = (\alpha, \beta, \gamma, \delta, x)$ and calculates common reference string $\sigma = ([\sigma_1]_1, [\sigma_2]_2)$, where

$$\sigma_1 = \left( \begin{array}{c} \alpha, \beta, \delta, \{x^i\}_{i=0}^{n-1}, \left\{ \frac{\beta u_i(x) + \alpha v_i(x) + \omega_i(x)}{\gamma} \right\}_{i=0}^{k} \\ \left\{ \frac{\beta u_i(x) + \alpha v_i(x) + \omega_i(x)}{\delta} \right\}_{i=k+1}^{m}, \left\{ \frac{x^i t(x)}{\delta} \right\}_{i=0}^{n-2} \end{array} \right)$$

$$\sigma_2 = (\beta, \gamma, \delta, \{x^i\}_{i=0}^{n-1}).$$

   e) The IDP publishes quadratic arithmetic program $R$, common reference string $\sigma$, a set of committed values $\boldsymbol{c}_A$, and the SP's identity identifier $\phi$.
2) **UserReg**$(1^\ell, \mathbf{\Gamma}, pw)$
   a) The user chooses a security parameter $\ell$, obtains public and private keys by running $Gen(\ell)$, and picks a secure cryptographic hash function $H()$, where $H : \{0,1\}^* \to \mathbb{Z}_n^*$.
   b) The user obtains facial image of size $n \times n$ through the IoT devices. Then the user can preprocess the captured biometric image, such as image graying, image size adjustment, histogram equalization. Finally, the obtained biometric image is converted into an $N = n * n$ dimensional vector $\mathbf{\Gamma} = (x_1, \ldots, x_N)$. The user encrypts the obtained biometric image to get a set of ciphertext $\mathbf{\Gamma}' = (x'_1, \ldots, x'_N)$.
   c) The user selects a random value $e \in \mathbb{Z}_p^*$, enters a password $pw \in \mathbb{Z}_p^*$, and sets $v = \mathbf{\Gamma}' || e || pw'$, where $pw'$ is the encrypted value of $pw$. Then the user calculates identity information by computing $v'' = v || \varepsilon_{sk}(v'; \varsigma)$, where $\varepsilon_{sk}(v'; \varsigma)$ is the digital signature of $v'$, and $v'$ is the hash value of $v$. Finally, the information of $v''$, the user's public key $pk$, and hash function $H()$ are added to the request message to register the identity with the IDP.
3) **CalcEigen**$(v'', pk, H())$
   a) After receiving the identity registration request sent by the user, the IDP checks the validity of the user's identity information by verifying whether the following equation holds.

$$\zeta_{pk}(\varepsilon_{sk}(v'; \varsigma)) = Hash(\mathbf{\Gamma}' || e || pw') \tag{1}$$

   If the equation (1) holds, the IDP parses $v''$ to obtain the user's encrypted biometric image $\mathbf{\Gamma}'$, random value $e$ and the encrypted password $pw'$.
   b) The IDP uses the random value $e$ generated by the user to construct a special set of vectors $(\boldsymbol{C}_1, \ldots, \boldsymbol{C}_k)$, where $\boldsymbol{C}_i = (pw' \cdot \boldsymbol{C})^{i \cdot e}$, $1 \le i \le k$, and $\boldsymbol{C}$ is the ciphertext of the difference between the user's original biometric image $\mathbf{\Gamma}$ and the mean $\psi$ of the selected images in the face dataset. Then, the IDP calculates a set of ciphertexts $(\theta'_1, \ldots, \theta'_k)$ using the inner product of the corresponding vectors in $(\boldsymbol{C}_1, \ldots, \boldsymbol{C}_k)$ and $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k)$, as follows:

$$\theta'_i = \boldsymbol{C}_i^{\boldsymbol{u}'_i}. \tag{2}$$

   It can be seen as the transformation result of projecting the user's encrypted biometric image onto the eigen faces subspace $W$. For convenience, we represent the processed eigen vectors $(\boldsymbol{u}'_1, \ldots, \boldsymbol{u}'_k)$ as $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k)$, where $k = \mu m'$.
   c) The IDP chooses $\rho \in \mathbb{Z}_q$ at random, computes $C = \varepsilon_{pk}(1; \rho) \prod_{i=1}^{k} \boldsymbol{C}_i^{\boldsymbol{a}_i}$, and picks $\boldsymbol{b} = (b_0, \ldots, b_{\mu-2})$, $\boldsymbol{s}, \boldsymbol{\tau} \leftarrow \mathbb{Z}_q^{\mu-1}$, and sets $b_{\lfloor \mu/2 \rfloor} = 0$, $s_{\lfloor \mu/2 \rfloor} = 0$, $\tau_{\lfloor \mu/2 \rfloor} = \rho$. Computing for $k = 0, \ldots, \mu - 2$

$$c_{b_k} = com_{ck}(b_k; s_k)$$

$$E_k = \varepsilon_{pk}(b_k; G^{\tau_k}) \prod_{\ell=0}^{m'-1} \prod_{\substack{i=1,j=1 \\ j=(k+1-\mu)+i}}^{\mu,\mu} \boldsymbol{C}_{\mu\ell+i}^{\boldsymbol{a}_{\mu\ell+j}} \cdot \boldsymbol{C}_{\mu\ell+j}^{\boldsymbol{a}_{\mu\ell+i}}.$$

Then, the IDP sends $(\theta'_1, \ldots, \theta'_k)$, $\boldsymbol{c}_b = (c_{b_0}, ..., c_{b_{\mu-2}})$, $\boldsymbol{E} = (E_0, ..., E_{\mu-2})$, $C$, and statement $\phi$ to the user.

4) **EigenVerify**$(x, C, \boldsymbol{E}, \boldsymbol{c}_A, \boldsymbol{c}_b)$
i) The user wants to verify whether the operation results sent by the IDP was generated from the user's encrypted biometric image. He randomly chooses a challenge value $x \leftarrow \mathbb{Z}_q^*$ and sends it to the IDP.
ii) After receiving the user's challenge value $x$, the IDP makes the following calculations:
   a) Sets $\boldsymbol{x} = (1, x, ..., x^{\mu-2})^T$ and sends $s = \boldsymbol{s} \cdot \boldsymbol{x}$, $b = \boldsymbol{b} \cdot \boldsymbol{x}$ and $\rho' = \boldsymbol{\tau} \cdot \boldsymbol{x}$ to the user.
   b) Defines $\boldsymbol{C}'_1, ..., \boldsymbol{C}'_{m'}$ and $c_{A'_1}, \ldots, c_{A'_{m'}}$ and $y^i$ and $C''$ by

$$\boldsymbol{C}'_\ell = \prod_{i=1}^{\mu} \boldsymbol{C}_{\mu(\ell-1)+i}^{x^{\mu-i}} \qquad y^i = x^{m'-1} \cdot (x^i - x^{-i})$$

$$c_{A'_\ell} = \prod_{j=1}^{\mu} c_{A_{\mu(\ell-1)+j}}^{x^{j-1}} \qquad C'' = \varepsilon_{pk}(-b; G^{-\rho'}) \boldsymbol{E}^{\boldsymbol{x}}$$

   c) Computes for $\ell = 1, 2, \ldots, m'$

$$\boldsymbol{a}'_\ell = \sum_{j=1}^{\mu} x^{j-1} \boldsymbol{a}_{\mu(\ell-1)+j} \quad r'_\ell = \sum_{j=1}^{\mu} x^{j-1} r_{\mu(\ell-1)+j}$$

$$\upsilon_\ell = \prod_{j=1}^{m'-1} \boldsymbol{C}_{\mu(\ell-1)+j}^{\sum_{i=1}^{i+j\le m'} y^i \cdot \boldsymbol{a}_{\mu(\ell-1)+i+j}}$$

   d) Sends $\upsilon = \prod_{\ell=1}^{m'} \upsilon_\ell, \boldsymbol{a}'_1, \ldots, \boldsymbol{a}'_{m'}, \varepsilon_{pk}(1; \rho)^{x^{-1+\mu}}$ and $r'_1, \ldots, r'_{m'}$ to the user.
iii) The user checks whether $\boldsymbol{c}_b \in \mathbb{G}^{\mu-1}$, and $E_0, ..., E_{\mu-2} \in \mathbb{H}$, and $b, s \in \mathbb{Z}_q$. Accept if for $\ell = 1, 2, \ldots, m'$

$$c_{A'_\ell} = com_{ck}(\boldsymbol{a}'_\ell; r'_\ell) \tag{3}$$

and

$$c_{b_{\lfloor \mu/2 \rfloor}} = com_{ck}(0; 0) \tag{4}$$

$$\boldsymbol{c}_b^{\boldsymbol{x}} = com_{ck}(b; s) \tag{5}$$

$$\varepsilon_{pk}(1; \rho)^{x^{-1+\mu}} \prod\nolimits_{\ell=1}^{m'} \boldsymbol{C}'^{\boldsymbol{a}'_\ell}_\ell = \upsilon \cdot C'' \cdot C^{x^{-1+\mu}} \tag{6}$$

5) **ProofGen**$(R, \sigma, \theta'_1, \ldots, \theta'_k, \phi)$
a) After receiving the transformation result of projecting the user's encrypted biometric image onto the eigen faces subspace $W$, the user decrypts it to obtain the corresponding plaintext data $(\theta_1, \ldots, \theta_k)$ as follows:

$$\theta_i = \frac{\zeta_{sk}(\theta'_i)}{i \cdot e} \tag{7}$$

where $\zeta_{sk}()$ is the decryption function of the paillier algorithm, $1 \le i \le k$, and $sk$ is the user's private key.
b) The user uses the decrypted data $(\theta_1, \ldots, \theta_k)$ to calculate exclusive witness $(pw, \theta_1, \ldots, \theta_k, c_1, \ldots, c_k, O)$ by the following equation:

$$c_i = k\theta_i \cdot a_i \tag{8}$$

$$pw + \theta_1 \cdot a_1 + 2\theta_2 \cdot a_2 + \cdots + k\theta_k \cdot a_k = O \tag{9}$$

For ease of description, we present the user's witness as $(a_{k+1}, \ldots, a_m)$.
c) The user randomly selects elements $r, s \leftarrow \mathbb{Z}_p^*$ and computes identity proof $\pi = ([A]_1, [C]_1, [B]_2)$, where

$$A = \alpha + \sum_{i=0}^{m} a_i u_i(x) + r\delta$$

$$B = \beta + \sum_{i=0}^{m} a_i v_i(x) + s\delta$$

$$C = \frac{\sum_{i=k+1}^{m} a_i(\beta u_i(x) + \alpha v_i(x) + w_i(x)) + h(x)t(x)}{\delta} + As + Br - rs\delta$$

d) Finally, the user outputs the identity proof $\pi = ([A]_1, [C]_1, [B]_2)$ to the SP.

6) **ProofVerify**$(R, \sigma, \phi, \pi)$
a) The SP parses $\pi = ([A]_1, [C]_1, [B]_2) \in \mathbb{G}_1^2 \times \mathbb{G}_2$, and verifies the truth of the user identity as follows:

$$[A]_1 \cdot [B]_2 = \sum_{i=0}^{k} a_i \left[ \frac{\beta u_i(x) + \alpha v_i(x) + \omega_i(x)}{\gamma} \right]_1 \cdot [\gamma]_2 + [\alpha]_1 \cdot [\beta]_2 + [C]_1 \cdot [\delta]_2 \tag{10}$$

If the equation (10) holds, the SP authenticates the user and allows the user to perform any transactions.
b) Furthermore, the SP can batch validate the truth of the user identity. The aggregation method of the identity proof is as follows:
First, the SP initializes 4 elements $(A, B, C, D)$ and their corresponding values are all zero. Then, we assume that $([A_i]_1, [B_i]_2, [C_i]_1)$ represents the identity proof sent by the $i$-th user. After receiving identity authentication messages from $N$ users, we have

$$\begin{cases} [D]_T := [D]_T + [A]_1 \cdot [B_i]_2 + [B]_2 \cdot [A_i]_1 \\ [A]_1 := [A]_1 + [A_i]_1 \\ [B]_2 := [B]_2 + [B_i]_2 \\ [C]_1 := [C]_1 + [C_i]_1 \end{cases} \tag{11}$$

In this way, the SP can aggregate the identity proofs sent by the users. Finally, by judging whether the following equation is true.

$$[A]_1 \cdot [B]_2 = [D]_T + [C]_1 \cdot [\delta]_2 + [N \cdot \alpha]_1 \cdot [\beta]_2 + \sum_{i=0}^{k} a_i \left[ \frac{\beta u_i(x) + \alpha v_i(x) + \omega_i(x)}{\gamma} \right]_1 \cdot [N \cdot \gamma]_2 \tag{12}$$

If the above equation holds, the SP verifies the identity of all users. Otherwise, there are illegal users who forged the proof.

## V. Security Analysis

In this section, we demonstrate that our identity authentication scheme is secure and privacy-preserving, and analyze the properties of Completeness, Soundness, Zero-knowledge.

*Theorem 1(Perfect completeness)*: Our identity authentication scheme satisfies the following three core properties:

1) When the user sends identity registration request to the IDP, the request message must be able to pass the verification of the IDP.
2) When the IDP uses the encrypted biometric image and the encrypted ppassword sent by the user to calculate the transformation result of projecting the user's encrypted biometric image onto the eigen faces subspace $W$, it must be able to pass the verification of the user and make the user believe in the truth of the operation results.
3) When the user has obtained the transformation result generated by the IDP, the identity proof $\pi$ constructed by the user must pass the verification of the SP.

*Proof*:

1) Given parameter $v''$, the user's public key $pk$ and hash function $H()$, the verification equation will hold in *CalcEigen* phase. Using the properties of discrete logarithm, the verification equation (1) can be proved correct. The detailed description of the proof is as follows:

$$\begin{aligned}\zeta_{pk}(\varepsilon_{sk}(v';\varsigma)) &= \zeta_{pk}(\varepsilon_{sk}(Hash(v);\varsigma))\\ &= Hash(v)\\ &= Hash(\mathbf{\Gamma}'||e||pw').\end{aligned}$$

2) Given parameters $x$, $C$, and a set of vectors $\boldsymbol{c}_A$, $\boldsymbol{c}_b$, $\boldsymbol{E}$, the verification equation (3), (4), (5) and (6) will hold in *EigenVerify* phase. For the sake of description, we present $com_{ck}()$ as $\nu()$.

a) Using the properties of discrete logarithm, the verification equation (3) can be proved correct. The detailed description of the proof is as follows:

$$\begin{aligned}&c_{A'_\ell} = \textstyle\prod_{j=1}^{\mu} c_{A_{\mu(\ell-1)+j}}^{x^{j-1}}\\ &= c_{A_{\mu(\ell-1)+1}} \cdot c_{A_{\mu(\ell-1)+2}}^{x} \cdots c_{A_{\mu(\ell-1)+\mu}}^{x^{\mu-1}}\\ &= \nu(\boldsymbol{a}_{\mu(\ell-1)+1}; r_{\mu(\ell-1)+1}) \cdot \nu(x\boldsymbol{a}_{\mu(\ell-1)+2}; xr_{\mu(\ell-1)+2})\\ &\quad \cdots \nu(x^{\mu-1}\boldsymbol{a}_{\mu(\ell-1)+\mu}; x^{\mu-1}r_{\mu(\ell-1)+\mu})\\ &= \nu(\textstyle\sum_{j=1}^{\mu} x^{j-1}\boldsymbol{a}_{\mu(\ell-1)+j}; \sum_{j=1}^{\mu} x^{j-1}r_{\mu(\ell-1)+j})\\ &= \nu(\boldsymbol{a}'_\ell; r'_\ell).\end{aligned}$$

b) Using the properties of discrete logarithm, the verification equation (5) can be proved correct. The detailed description of the proof is as follows:

$$\begin{aligned}&\boldsymbol{c}_{\boldsymbol{b}}^{\boldsymbol{x}} = \textstyle\prod_{i=0}^{\mu-2} c_{b_i}^{x^i}\\ &= c_{b_0} \cdot c_{b_1}^{x^1} \cdots c_{b_{\mu-3}}^{x^{\mu-3}} \cdot c_{b_{\mu-2}}^{x^{\mu-2}}\\ &= \nu(b_0; s_0) \cdots \nu(x^{\mu-2}b_{\mu-2}; x^{\mu-2}s_{\mu-2})\\ &= \nu(b_0 + \cdots + x^{\mu-2}b_{\mu-2}; s_0 + \cdots + x^{\mu-2}s_{\mu-2})\\ &= \nu(\textstyle\sum_{i=0}^{\mu-2} x^i b_i; \sum_{i=0}^{\mu-2} x^i s_i)\\ &= \nu(\boldsymbol{b}\cdot\boldsymbol{x}; \boldsymbol{s}\cdot\boldsymbol{x})\\ &= \nu(b; s).\end{aligned}$$

c) Using the properties of discrete logarithm, the verification equation (6) can be proved correct. The detailed description of the proof is as follows:

$$\begin{aligned}&\upsilon \cdot C'' \cdot C^{x^{-1+\mu}} = \upsilon \cdot \varepsilon_{pk}(-b; G^{-\rho'}) \cdot \boldsymbol{E}^{\boldsymbol{x}} \cdot C^{x^{-1+\mu}}\\ &= \prod_{\ell=1}^{m'} \prod_{j=1}^{m'-1} \boldsymbol{C}_{\mu(\ell-1)+j}^{\sum_{i=1}^{i+j\le m'} y^i \boldsymbol{a}_{\mu(\ell-1)+i+j}} \cdot \varepsilon_{pk}(-b; G^{-\rho'})\\ &\quad \cdot \prod_{k=0}^{\mu-2} {E_k}^{x^k} \cdot \varepsilon_{pk}(1;\rho)^{x^{-1+\mu}} \prod_{i=1}^{k} \boldsymbol{C}_i^{x^{-1+\mu}\cdot\boldsymbol{a}_i}\\ &= \prod_{\ell=1}^{m'} \prod_{j=1}^{m'-1} \boldsymbol{C}_{\mu(\ell-1)+j}^{\sum_{i=1}^{i+j\le m'} y^i \boldsymbol{a}_{\mu(\ell-1)+i+j}} \cdot \varepsilon_{pk}(-\boldsymbol{b}\cdot\boldsymbol{x}; G^{-\boldsymbol{\tau}\cdot\boldsymbol{x}})\\ &\quad \cdot \prod_{k=0}^{\mu-2} \prod_{\ell=0}^{m'-1} \prod_{\substack{i=1,j=1\\ j=(k+1-\mu)+i}}^{\mu,\mu} (\boldsymbol{C}_{\mu\ell+i}^{\boldsymbol{a}_{\mu\ell+j}} \cdot \boldsymbol{C}_{\mu\ell+j}^{\boldsymbol{a}_{\mu\ell+i}})^{x^k}\\ &\quad \cdot \varepsilon_{pk}(1;\rho)^{x^{-1+\mu}} \prod_{i=1}^{k} \boldsymbol{C}_i^{x^{-1+\mu}\cdot\boldsymbol{a}_i} \cdot \prod_{k=0}^{\mu-2} \varepsilon_{pk}(b_k; G^{\tau_k})^{x^k}\\ &= \varepsilon_{pk}(-\boldsymbol{b}\cdot\boldsymbol{x}; G^{-\boldsymbol{\tau}\cdot\boldsymbol{x}}) \cdot \varepsilon_{pk}(\boldsymbol{b}\cdot\boldsymbol{x}; G^{\boldsymbol{\tau}\cdot\boldsymbol{x}}) \cdot \varepsilon_{pk}(1;\rho)^{x^{\mu-1}}\\ &\quad \cdot \prod_{\ell=1}^{m'} \prod_{j=1}^{m'-1} \boldsymbol{C}_{\mu(\ell-1)+j}^{\sum_{i=1}^{i+j\le m'} (x^{m'-1+i} - x^{m'-1-i})\boldsymbol{a}_{\mu(\ell-1)+i+j}}\\ &\quad \cdot \prod_{k=0}^{\mu-2} \prod_{\ell=0}^{m'-1} \prod_{\substack{i=1,j=1\\ j=(k+1-\mu)+i}}^{\mu,\mu} \boldsymbol{C}_{\mu\ell+i}^{x^k\cdot\boldsymbol{a}_{\mu\ell+j}} \cdot \boldsymbol{C}_{\mu\ell+j}^{x^k\cdot\boldsymbol{a}_{\mu\ell+i}}\\ &\quad \cdot \prod_{i=1}^{k} \boldsymbol{C}_i^{x^{-1+\mu}\cdot\boldsymbol{a}_i}\\ &= \varepsilon_{pk}(1;\rho)^{x^{-1+\mu}} \prod_{\ell=1}^{m'} (\prod_{i=1}^{\mu} \boldsymbol{C}_{\mu(\ell-1)+i}^{x^{\mu-i}})^{\sum_{j=1}^{\mu} x^{j-1}\boldsymbol{a}_{\mu(\ell-1)+j}}\\ &= \varepsilon_{pk}(1;\rho)^{x^{-1+\mu}} \prod_{\ell=1}^{m'} \prod_{i=1}^{\mu} \boldsymbol{C}_{\mu(\ell-1)+i}^{\sum_{j=1}^{\mu} x^{\mu-i+j-1}\boldsymbol{a}_{\mu(\ell-1)+j}}\\ &= \varepsilon_{pk}(1;\rho)^{x^{-1+\mu}} \prod_{\ell=1}^{m'} \boldsymbol{C}'^{\boldsymbol{a}'_\ell}_{\ell}.\end{aligned}$$

3) Given an identity proof $\pi$ generated by the user, the verification equation (10) will hold in *ProofVerify* phase. The detailed description of the proof is as follows:

$$\begin{aligned}&[A]_1 \cdot [B]_2 = ([\alpha]_1 + [\textstyle\sum_{i=0}^{m} a_i u_i(x)]_1 + [r\delta]_1)\\ &\quad \cdot ([\beta]_2 + [\textstyle\sum_{i=0}^{m} a_i v_i(x)]_2 + [s\delta]_2)\\ &= [\alpha\cdot\beta]_T + [\alpha \cdot \textstyle\sum_{i=0}^{m} a_i v_i(x)]_T + [\alpha\cdot s\delta]_T\\ &\quad + [r\delta \cdot s\delta]_T + [\textstyle\sum_{i=0}^{m} a_i u_i(x) \cdot \sum_{i=0}^{m} a_i v_i(x)]_T\\ &\quad + [s\delta \cdot \textstyle\sum_{i=0}^{m} a_i u_i(x)]_T + [r\delta \cdot \sum_{i=0}^{m} a_i v_i(x)]_T\\ &\quad + [r\delta\cdot\beta]_T + [\beta \cdot \textstyle\sum_{i=0}^{m} a_i u_i(x)]_T\\ &= [\alpha\cdot\beta]_T + [\textstyle\sum_{i=0}^{k} a_i(\beta u_i(x) + \alpha v_i(x) + w_i(x))]_T\end{aligned}$$

$$
\begin{aligned}
&+ [h(x)\cdot t(x)]_T + [s\delta \cdot \sum_{i=0}^{m} a_i u_i(x)]_T + [r\delta \cdot s\delta]_T \\
&+ [r\delta \cdot \beta]_T + [r\delta \cdot \sum_{i=0}^{m} a_i v_i(x)]_T + [\alpha \cdot s\delta]_T \\
&+ [\sum_{i=k+1}^{m} a_i(\beta u_i(x) + \alpha v_i(x) + w_i(x))]_T \\
&= [\alpha \cdot \beta]_T + [\sum_{i=0}^{k} a_i(\beta u_i(x) + \alpha v_i(x) + w_i(x))]_T \\
&+ [A \cdot s\delta]_T + [B \cdot r\delta]_T - [r\delta \cdot s\delta]_T + [h(x)\cdot t(x)]_T \\
&+ [\sum_{i=k+1}^{m} a_i(\beta u_i(x) + \alpha v_i(x) + w_i(x))]_T \\
&= [\frac{\sum_{i=0}^{k} a_i(\beta u_i(x) + \alpha v_i(x) + w_i(x))}{\gamma} \cdot \gamma]_T \\
&+ [C\cdot\delta]_T + [\alpha\cdot\beta]_T \\
&= \sum_{i=0}^{k} a_i \left[\frac{\beta u_i(x) + \alpha v_i(x) + \omega_i(x)}{\gamma}\right]_1 \cdot [\gamma]_2 \\
&+ [\alpha]_1 \cdot [\beta]_2 + [C]_1 \cdot [\delta]_2.
\end{aligned}
$$

4) When receiving the identity authentication messages sent by multiple users, the SP can use the batch validation method of *ProofVerify* algorithm to check the truth of the user identity, thereby reducing unnecessary computational overhead. Using the properties of bilinear pairing, the verification equation (11) and (12) can be proved correct. The detailed description of equation (12) is as follows:

$$
\begin{aligned}
&\sum_{\substack{i=1,j=1\\ i\neq j}}^{N,N} [A_i]_1 \cdot [B_i]_2 \\
&= \sum_{i=2}^{N} ([A_i]_1 \cdot \sum_{j=1}^{i-1} [B_i]_2 + [B_i] \cdot \sum_{j=1}^{i-1} [A_i]_1) \\
&= \sum_{i=2}^{N} ([A_i]_1 \cdot \sum_{j=1}^{i-1} [B_i]_2) + \sum_{i=2}^{N} ([B_i]_2 \cdot \sum_{j=1}^{i-1} [A_i]_1) \\
&= \sum_{i=2}^{N} [B]_2 \cdot [A_i]_1 + \sum_{i=2}^{N} [A]_1 \cdot [B_i]_2 \\
&= [D]_T
\end{aligned}
\tag{13}
$$

$$
\begin{aligned}
&[A]_1 \cdot [B]_2 = [A\cdot B]_T \\
&= [(A_1 + A_2 + \cdots + A_N)\cdot(B_1 + B_2 + \cdots + B_N)]_T \\
&= [\sum_{i=1}^{N} A_i \cdot B_i]_T + [\sum_{\substack{i=1,j=1\\ i\neq j}}^{N,N} A_i \cdot B_j]_T \\
&= \sum_{i=1}^{N} [A_i]_1 [B_i]_2 + [\sum_{\substack{i=1,j=1\\ i\neq j}}^{N,N} A_i \cdot B_j]_T \\
&= [\frac{\sum_{\ell=1}^{k} a_i(\beta u_i(x) + \alpha v_i(x) + w_i(x))}{\gamma} \cdot \gamma \cdot N]_T \\
&+ [N\cdot\alpha\beta]_T + [\sum_{i=1}^{N} C_i \cdot \delta] + [\sum_{\substack{i=1,j=1\\ i\neq j}}^{N,N} A_i \cdot B_j]_T \\
&= [\sum_{\ell=1}^{k} \frac{a_i(\beta u_i(x) + \alpha v_i(x) + w_i(x))}{\gamma}]_1 \cdot [\gamma\cdot N]_2 \\
&+ [N\cdot\alpha]_1 \cdot [\beta]_2 + [C]_1 \cdot [\delta] + [\sum_{\substack{i=1,j=1\\ i\neq j}}^{N,N} A_i \cdot B_j]_T \\
&= [\sum_{\ell=1}^{k} \frac{a_i(\beta u_i(x) + \alpha v_i(x) + w_i(x))}{\gamma}]_1 \cdot [\gamma\cdot N]_2 \\
&+ [N\cdot\alpha]_1 \cdot [\beta]_2 + [C]_1 \cdot [\delta] + [D]_T
\end{aligned}
$$

the last identity is by equation (13).

*Theorem 2* (*Computational knowledge soundness*) Suppose the $q$-PKE assumption and $q$-CPDH assumption [10] is hard in bilinear groups, and the IDP can honestly execute the operations to generate the public parameters required by the system. In our authentication scheme, for a non-uniform polynomial time adversary $\mathcal{A}$ or an untrusted service provider, it is computationally infeasible to forge a valid identity proof that can pass the verifier's verification if the biometric information is not registered in IDP.

*Proof*:

The detailed process of the proof can be found in [14].

*Theorem 3* (*Perfect zero-knowledge*) The SP cannot derive any useful sensitive information from the user's identity proof $\pi$ besides the truth of the user identity and the IDP only calculates the encrypted biometric image, and cannot obtain the user's sensitive information about the biometrics from the operation results and the intermediate value of the operation process.

*Proof*:

1) In our identity authentication scheme, the user needs to use sensitive information about the biometrics to generate identity proof $\pi$ to demonstrate the truth of his identity before being allowed to execute any transactions. We know that the parameters $r$ and $s$ are chosen randomly by the user, and common reference string $\sigma$ generated by the IDP and the identity proof $\pi$ received by the SP are unpredictable. Therefore, the SP cannot obtain the user's sensitive information about the biometrics through the identity proof $\pi$ generated by the user. It means that the SP cannot derive any useful information from the user's identity proof besides the truth of the user identity.
2) After receiving the encrypted biometric image from the user, the IDP first calculates the difference between the user's encrypted biometric and the mean of the selected images in the face dataset. Then, the IDP calculates the transformation result of projecting the user's encrypted biometric image onto the eigen faces subspace $W$. Due to the entire operation is performed in the ciphertext space, it means that the user cannot analyze the user's sensitive information about the biometrics from any computing process. That is to say, the IDP only calculates the encrypted biometric image, and cannot obtain the user's sensitive information about the biometrics from the operation results and the intermediate value of the operation process. Therefore, our identity authentication scheme is *perfect zero knowledge*.

## VI. Performance Evaluation

In this section, we analyze the computational overhead and communication overhead of our scheme. Then we implement the scheme and evaluate its performance in experiments.

### A. Performance Analysis and Comparison

For the convenience of description, we use new notations to denote the following operations. We assume that $\mathbb{G}$ means one group element, *P* denotes one pairing operation, *exp* means one exponentiation operation, $Mul$ denotes one multiplication operation in $\mathbb{Z}_p^*$, and $Add$ denotes one addition operation in

TABLE II: The computational overhead of our scheme is compared with other schemes

| | Our scheme | e-Finga | PrivBioMTAuth |
|---|---|---|---|
| User comp. | $3n+m-k$ $\mathbb{G}_1$ $exp$, $n$ $\mathbb{G}_2$ $exp$, $k$ $Add$<br>$2N+3k+4m+4$ $Mul$, $N+3k+m+4$ $exp$ | $N'+1$ $exp$, $2N'$ $Mul$ | 3 $Mul$, 2 $exp$, 4 $Add$ |
| SP/OASer comp. | $3P$, $k$ $\mathbb{G}_1$ $exp$ | $N'P$, $N'+1$ $Mul$ | 3 $Mul$, 7 $exp$ |
| IDP/TA comp. | $2k+m'k+4\mu-3$ $exp$, $2k-2m'$ $Add$<br>$k^2+3\mu+k-2$ $Mul$ | $2N'+1$ $exp$, $2N'$ $Mul$ | 1 $Mul$, 2 $exp$, $T$ |

$\mathbb{Z}_p^*$. $k$ is the number of the eigen vectors selected by the IDP. $m$ is the number of the wires in arithmetic circuit. $n$ is the number of the multiplication gates in arithmetic circuit. $N'$ is the dimension of the FingerCode. $T$ is the computational overhead incurred to train the machine learning model.

(1) **Computation overhead comparison**. In Table II, we give a comparison of our scheme with Zhu et al. [8] and Gunasinghe et al. [5] in terms of computational overhead. Assuming that the user's biometric image can be represented by an $N$-dimensional vector, $\boldsymbol{C}^e$ represents one exponentiation operation, and $x \cdot \boldsymbol{a}$ represents one multiplication operation. When generating an encrypted biometric image $\mathbf{\Gamma}' = (x'_1, x'_2, \ldots, x'_N)$, the user needs $2N$ exponentiation operations and $N$ multiplication operations. When the user decrypts the transformation result of projecting the user's encrypted biometric image onto the eigen faces subspace $W$, $k$ multiplication operations and $2k$ exponentiation operations are required. When the user calculates the witness based on the transformation result, $k$ multiplication operations and $k$ addition operations are required. When the user verifies the truth of the transformation result generated by the IDP, $k+3m'+1$ multiplication operations and $k+4m'+4$ exponentiation operations are required. When the user constructs the identity proof, $m+3n-k$ exponentiation operations in $\mathbb{G}_1$ and $n$ exponentiation operations in $\mathbb{G}_2$ are required. In the *CalcEigen* phase, the IDP calculates the transformation result of projecting the user's encrypted biometric image onto the eigen faces subspace $W$, which requires $2k$ exponentiation operations and $k-1$ multiplication operations. When constructing the multi-exponentiation arguments, the IDP requires a total of $2k-2m'$ addition operations, $k^2+3\mu-1$ multiplication operations and $m'k+4\mu-3$ exponentiation operations. In the *ProofVerify* phase, the SP will cost 3 pairing operations and $k$ multiplication operations in $\mathbb{G}_1$ when receiving the authentication message sent by the user.

Next, we will select two different biometric-based authentication schemes for comparison. One of the schemes is to calculate the secure Euclidean distance based on an improved homomorphic encryption technique to achieve identity authentication [8], which is called e-Finga in the rest of the paper. Another scheme is to achieve identity authentication on mobile phones through interactive zero-knowledge proofs and cryptographic commitments, which is called PrivBioMTAuth for convenience. In the e-Finga, we assume that FingerCode is an $N'$-dimensional vector. When generating encrypted biometric information, the computational overhead of the user is $N'+1$ exponentiation operations and $2N'$ multiplication operations. When receiving the user's authentication message, the computational overhead of the TA is $2N'+1$ exponentiation operations and $2N'$ multiplication operations. When generating encrypted templates, the computational overhead of the OASer is $N'$ pairing operations and $N'+1$ multiplication operations. In the PrivBioMTAuth, the user's computational overhead and the SP's computational overhead are mainly concentrated in the authentication phase. The computational overhead of the user is 3 exponentiation operations, 2 multiplication operations and 4 addition operations. When receiving the user's identity authentication message, the computational overhead of the SP is 7 exponentiation operations and 3 multiplication operations. The IDP needs to use machine learning algorithms to train a large number of learning samples during the identity registration phase, which will consume too much computing time. In addition, in order to generate an identity token for the user, the IDP also needs 2 exponentiation operations and 1 multiplication operations. As shown in Table II, we can know that our scheme costs less computational overhead than the other two schemes during the identity registration phase. It means that our scheme is more efficient than other schemes. Simultaneously, we assume that the IDP is a more general semi-honest entity than the third party of other schemes. During the authentication phase, the e-Finga costs more computational overhead than our scheme. Our scheme generates more computational overhead than the PrivBioMTAuth. However, our scheme is based on non-interactive zero-knowledge arguments, and the user only need to generate an identity proof once throughout the network to prove his identity when requesting services from multiple service providers. In terms of computational overhead of the users, our scheme costs less computational overhead than the e-Finga. Nonetheless, our scheme generates more computational overhead than the PrivBioMTAuth. Compared with the other two schemes, our scheme does not need to reveal the user's sensitive information at any phase. Therefore, in the above comparison, we can know that our scheme is more efficient and has privacy protection capabilities.

TABLE III: The communication overhead of our scheme

| Phase | Entity | Communication Overhead |
|---|---|---|
| Setup | IDP | $n+3m$ $\mathbb{G}_1$, $2k$ $\mathbb{G}$ |
| Registration | User, IDP | $N+7+2(\mu+2k)$ $\mathbb{G}$ |
| Authentication | User | 2 $\mathbb{G}_1$, 1 $\mathbb{G}_2$ |

(2) **Communication overhead comparison**. From Section IV, we can see that the communication overhead of our scheme focuses on setup, registration and authentication phases, as shown in Table III. In the setup phase, the IDP needs to generate the common reference string, calculate the commitment value and send identity identifier to the SP. Hence, the

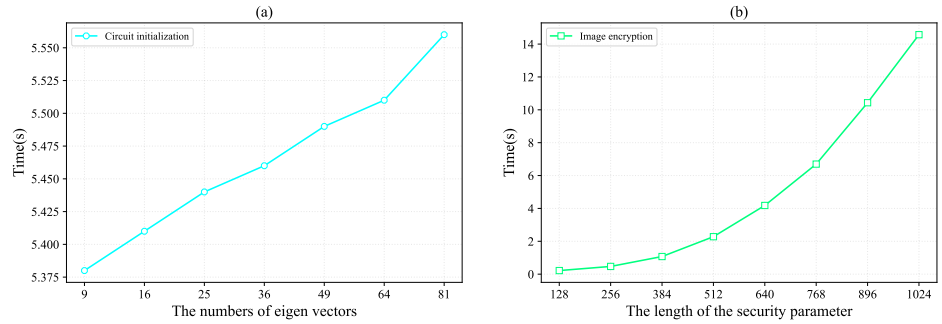


Fig. 2: The computational overhead of Setup phase and UserReg phase

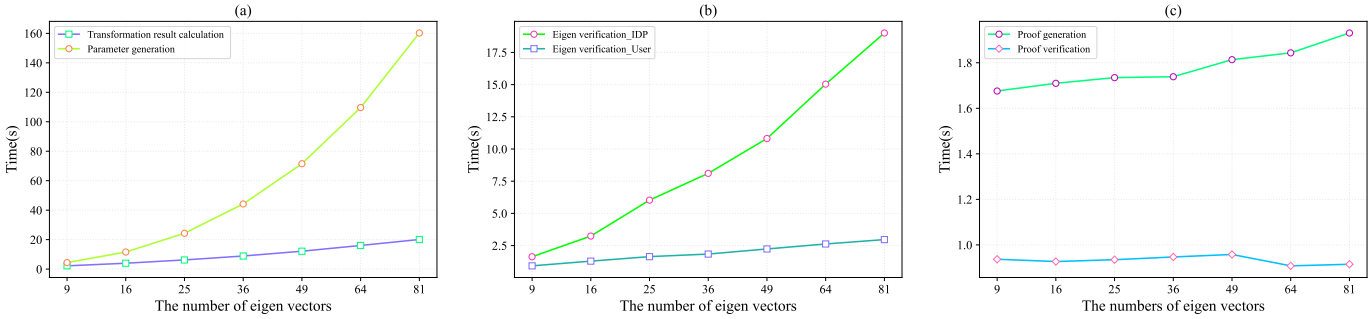


Fig. 3: The computational overhead of EigenVerify phase, ProofGen phase and ProofVerify phase

communication overhead of the setup phase is $n + 3m$ $\mathbb{G}_1$ and $2k$ $\mathbb{G}$. In the registration phase, the multi-exponentiation argument is used to verify the truth of the IDP operation results ,and so the communication overhead of the registration phase is $N + 7 + 2(\mu + 2k)$ $\mathbb{G}$. In the authentication phase, the user needs to construct an identity proof to demonstrate the truth of the identity before executing any transaction with SP. The communication overhead of the authentication phase is 2 $\mathbb{G}_1$ and 1 $\mathbb{G}_2$.

### B. Experimental Evaluation

In this subsection, we evaluate the performance of our identity authentication scheme through a series of experiments. We run these experiments on a desktop with Intel Core i5-7500 CPU, 16 GB memory and Ubuntu 20.04 OS. All of our experiments use the Python language, snarkjs library [33], circomlib library [29] and ORL Faces Database [39]. In our experiments, we use the BLS12-381 curve, set the field size to be 256 bit, and set the captured biometric images and ORL Database image size is 92×112 pixels.

(1) IDP: In our scheme, the computational overhead of the IDP is generated by the *Setup*, *CalcEigen* and *EigenVerify* phases. In the *Setup* phase, the computational overhead of the IDP is determined by the size of the circuit, which is determined by the number of the eigen vectors. As shown in Fig. 2a, we can see that the size of the circuit linearly increases with the number of the eigen vectors. In the *CalcEigen* phase, the factor which impacts the computational overhead is the number of the eigen vectors. Therefore, we set the number of the eigen vectors from 9 to 81. As shown in Fig. 3a, we can see that the time consumed to calculate the transformation result of projecting the user's biometric image onto the eigen faces subspace W ranges from 2.1s to 20s. Meanwhile, the generation of the vector $\boldsymbol{E}$ has a large computational overhead. The time cost of the vector $\boldsymbol{E}$ generation increases with the number of the eigen vectors. In the *EigenVerify* phase, we select the number of the eigen vectors from 9 to 81. As shown in Fig. 3b, we see that the time of parameter generation ranges from 1.6s to 19.0s.

(2) User: In our identity authentication scheme, the computational overhead of the user depends on the *UserReg*, *EigenVerify* and *ProofGen* phases. In the *UserReg* phase, the efficiency of the user's encrypted image depends on the length of the key. In order to further analyze the relationship between the length of the key and the efficiency of the encrypted image, the length of the key is selected from 128 to 1024 bit. As shown in Fig. 2b, the computational overhead of the user increases with the increasing of the length of the key. In the *EigenVerify* phase, the user's computational overhead depends on the execution time of equation (6). The computational complexity of equation (6) is determined by the number of eigen vectors. As shown in Fig. 3b, the number of the eigen vectors are selected from 9 to 81. Obviously, as the number of the eigen vectors increases, more computation time is consumed. In *ProofGen* phase, the number of the eigen vectors varies from 9 to 81. As shown in Fig. 3c, we can see that the computation overheads of the identity proof generation linearly increase with the number of the eigen vectors.

(3) SP: The operation of the SP is mainly in the *ProofVerify* phase. The computational complexity of the SP depends on the number of the eigen vectors. Therefore, different numbers of the eigen vectors were selected to illustrate the execution efficiency of the SP. To further observe the execution efficiency of the SP, the number of the eigen vectors are selected from

9 to 81. As shown in Fig. 3c, the computational overhead of the SP is stable at around 0.9 seconds. Because, our authentication scheme only needs to perform 3 pairings and $k$ exponentiations. If the selected $k$ is not particularly large, it will not affect the execution efficiency of the SP.

## VII. Related Work

The privacy protection of biometric data has always been a hot research direction. Therefore, in order to tackle the problem of disclosure of user's sensitive information about the biometrics, many privacy protection identity authentication schemes have been proposed.

The idea of using biometric data to generate key directly was first proposed in 1994 [20]. Then, Dodis *et al*. [21] proposed the idea of fuzzy extractor which constructed using a secure sketch and a strong extractor. The classical fuzzy extractor requires that the same sampled fingerprint cannot have large errors during registration and verification phase. Subsequently, many improved versions of fuzzy extractors have been proposed, and has been applied in many different scenarios [22], [23], [26]. However, some drawbacks were found when using fuzzy extractors to solve problems in a real scenarios. The reason is that in fuzzy extractors the release of multiple sketches associated with the same biometric features poses security and privacy issues due to the unavoidable information leakage [5], and security analysis have been given in [24] and [25].

In order to protect user's privacy, secure two-party computation setting was introduced to solve biometric data breaches [27], [30]. In this setting, the server and client execute the protocol interactively without revealing their respective saved biometric data during the execution of the agreement. Erkin *et al.* [28] proposed a new homomorphic encryption scheme for outsourcing scenarios, which was constructed by Paillier algorithm and DGK encryption scheme [31]. This method can ensure the security of the user's biometric data during the identity registration and authentication phase, and will not reveal the results of the identity authentication. Xiang *et al.* [32] constructed a privacy-preserving biometric recognition protocol for outsourcing computing based on fully homomorphic encryption. The proposed scheme can reduce considerable online computing overhead when large computation task outsourced to a cloud server. Zhu *et al.* [8] introduced a novel privacy-preserving online fingerprint authentication method based on an improved homomorphic encryption algorithm. It provides secure, accurate and efficient authentication services without revealing user's private information. Although this improved method improves efficiency, the increase in the dimension of FingerCode will inevitably lead to an increase in computational overhead. Unfortunately, Naehrig *et al.* [34] pointed out that homomorphic encryption is impracticality for arbitrary arithmetic computations. Meanwhile, it is not suitable for some resource-constrained scenarios or high real-time application scenarios.

The randomization technique was introduced to solve biometric sensitive information leakage problem. Yuan *et al.* [35] proposed a privacy-preserving biometric identification scheme which outsources encrypted databases to cloud servers, thereby reducing excessive computational overhead. The method uses the random matrix technique to calculate the Euclidean distance under the ciphertext space. Wang *et al.* [36] conducted an in-depth study on the existing biometric identification outsourcing problem, and proposed a new privacy preserving biometric identification protocols based on matrix tracing theory. This scheme improves efficiency by shifting the burden of database owners to the cloud. Zhang *et al.* [37] constructed a privacy-preserving biometric identification scheme, which adds a perturbation term to the biometric data and constructs a new encryption and feature matching algorithms, accommodating higher-level efficiency requirements. Hu *et al.* [38] proposed a privacy-preserving biometric identification scheme which achieves a higher level of privacy using two-server model. This method utilizes homomorphic encryption and batch protocol to achieve privacy protection of biometric data. Liu *et al.* [40] proposed a privacy-preserving and efficient biometric identification scheme which can protect against many types of attacks, using the subtly designed invertible matrix (SDIM) to protect user's private data.

Other aspects, such as zero-knowledge proof-based biometric authentication [5], [41] and the averaged event-related potential-based biometric identity [42] have also been proposed. Nevertheless, all the existing biometric-based authentication schemes suffer from the different drawbacks. In this paper, we discussed how to construct a privacy preserving biometric-based identity authentication scheme for Internet of Things.

## VIII. Conclusion

In this paper, based on NIZK arguments and multi-exponentiation arguments, we present a novel privacy preserving biometric identification scheme for Internet of Things. The advantages of our solution are as follows: i) The user's sensitive information about the biometrics is not exposed during the registration and authentication phase. ii) The identity proof does not reveal anything other than the truth of the user identity. iii) The user only needs to generate an identity proof once throughout the network to prove his identity when requesting services from multiple service providers.

The detailed analyses indicate that our solution can overcome the shortcomings of the existing biometric-based identity authentication schemes and achieve the privacy protection of the user's sensitive information about the biometrics. In the future work, we will focus on biometric-based revocable and regulatory identity authentication for Internet of Things.

## References

[1] A. MacGregor, "Amazon wants to replace passwords with selfies and videos," *Accessed: Jan*, vol. 15, p. 2017, 2016.

[2] N. Cappella, "Hsbc announces biometric banking with voice and fingerprints," February 2016, https://thestack.com/world/2016/02/19/hsbc-voice-biometric-online-banking/.

[3] M. Qi and J. Chen, "Anonymous biometrics-based authentication with key agreement scheme for multi-server environment using ecc," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 27 553–27 568, 2019.

[4] X. Li, K. Wang, J. Shen, S. Kumari, F. Wu, and Y. Hu, "An enhanced biometrics-based user authentication scheme for multi-server environments in critical systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, no. 3, pp. 427–443, 2016.

[5] H. Gunasinghe and E. Bertino, "Privbiomtauth: Privacy preserving biometrics-based and user centric protocol for user authentication from mobile phones," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 1042–1057, 2018.

[6] T. Zhu, Z. Qu, H. Xu, J. Zhang, Z. Shao, Y. Chen, S. Prabhakar, and J. Yang, "Riskcog: Unobtrusive real-time user authentication on mobile devices in the wild," *IEEE Transactions on Mobile Computing*, vol. 19, no. 2, pp. 466–483, 2019.

[7] D. Akdogan, D. K. Altop, L. Eskandarian, and A. Levi, "Secure key agreement protocols: pure biometrics and cancelable biometrics," *Computer Networks*, vol. 142, pp. 33–48, 2018.

[8] H. Zhu, Q. Wei, X. Yang, R. Lu, and H. Li, "Efficient and privacy-preserving online fingerprint authentication scheme over outsourced data," *IEEE Transactions on Cloud Computing*, vol. 9, no. 2, pp. 576–586, 2021.

[9] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4900–4913, 2018.

[10] J. Groth, "Short pairing-based non-interactive zero-knowledge arguments," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2010, pp. 321–340.

[11] V. M. Patel, N. K. Ratha, and R. Chellappa, "Cancelable biometrics: A review," *IEEE Signal Processing Magazine*, vol. 32, no. 5, pp. 54–65, 2015.

[12] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Conference on the theory and application of cryptographic techniques*. Springer, 1986, pp. 186–194.

[13] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Annual international cryptology conference*. Springer, 1991, pp. 129–140.

[14] J. Groth, "On the size of pairing-based non-interactive arguments," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2016, pp. 305–326.

[15] S. Bayer and J. Groth, "Efficient zero-knowledge argument for correctness of a shuffle," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2012, pp. 263–280.

[16] R. Gennaro, C. Gentry, B. Parno, and M. Raykova, "Quadratic span programs and succinct nizks without pcps," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2013, pp. 626–645.

[17] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[18] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1999, pp. 223–238.

[19] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," *Springer-Verlag*, 1991.

[20] A. Bodo, "Method for producing a digital signature with aid of a biometric feature," *German patent DE*, vol. 42, no. 43, p. 908, 1994.

[21] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *International conference on the theory and applications of cryptographic techniques*. Springer, 2004, pp. 523–540.

[22] R. K. Mahendran and P. Velusamy, "A secure fuzzy extractor based biometric key authentication scheme for body sensor network in internet of medical things," *Computer Communications*, vol. 153, pp. 545–552, 2020.

[23] D. Bao and L. You, "Two-factor identity authentication scheme based on blockchain and fuzzy extractor," *Soft Computing*, pp. 1–13, 2021.

[24] K. Simoens, P. Tuyls, and B. Preneel, "Privacy weaknesses in biometric sketches," in *2009 30th IEEE Symposium on Security and Privacy*. IEEE, 2009, pp. 188–203.

[25] M. Blanton and M. Aliasgari, "Analysis of reusability of secure sketches and fuzzy extractors," *IEEE transactions on information forensics and security*, vol. 8, no. 9, pp. 1433–1445, 2013.

[26] J. Ma, B. Qi, and K. Lv, "Threshold reusable fuzzy extractor and an application to joint access control via biometric information," *Information Sciences*, vol. 579, pp. 525–540, 2021.

[27] M. Barni, G. Droandi, and R. Lazzeretti, "Privacy protection in biometric-based recognition systems: A marriage between cryptography and signal processing," *IEEE Signal Processing Magazine*, vol. 32, no. 5, pp. 66–76, 2015.

[28] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *International symposium on privacy enhancing technologies symposium*. Springer, 2009, pp. 235–253.

[29] J. Baylina. *The CircomLib Library*. Accessed: Jan. 28, 2022. [Online]. Available: github.com/iden3/circomlib

[30] J. Bringer, H. Chabanne, and A. Patey, "Privacy-preserving biometric identification using secure multiparty computation: An overview and recent trends," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 42–52, 2013.

[31] I. Damgård, M. Geisler, and M. Krøigaard, "Efficient and secure comparison for on-line auctions," in *Australasian conference on information security and privacy*. Springer, 2007, pp. 416–430.

[32] C. Xiang, C. Tang, Y. Cai, and Q. Xu, "Privacy-preserving face recognition with outsourced computation," *Soft Computing*, vol. 20, no. 9, pp. 3735–3744, 2016.

[33] J. Baylina. *The Snarkjs Library*. Accessed: Jan. 28, 2022. [Online]. Available: github.com/iden3/snarkjs

[34] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, 2011, pp. 113–124.

[35] J. Yuan and S. Yu, "Efficient privacy-preserving biometric identification in cloud computing," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 2652–2660.

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

[37] C. Zhang, L. Zhu, and C. Xu, "Ptbi: An efficient privacy-preserving biometric identification based on perturbed term in the cloud," *Information Sciences*, vol. 409, pp. 56–67, 2017.

[38] S. Hu, M. Li, Q. Wang, S. S. Chow, and M. Du, "Outsourced biometric identification with privacy," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2448–2463, 2018.

[39] *The ORL Database of Faces*. Olivetti Research Laboratory. Accessed: Jan. 3, 2022. [Online]. Available: https://www.kaggle.com/tavarez/the-orl-database-for-training-and-testing

[40] C. Liu, L. Yang, L. Ma, L. Shi, X. Hu, W. Cao, and J. Zhang, "Pebiid: Privacy-preserving and efficient biometric identification for iov dapp," in *2021 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom)*, 2021, pp. 63–72.

[41] A. Bhargav-Spantzel, A. Squicciarini, E. Bertino, X. Kong, and W. Zhang, "Biometrics-based identifiers for digital identity management," in *Proceedings of the 9th Symposium on Identity and Trust on the Internet*, 2010, pp. 84–96.

[42] M. V. Ruiz-Blondet, Z. Jin, and S. Laszlo, "Cerebre: A novel method for very high accuracy event-related potential biometric identification," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 7, pp. 1618–1629, 2016.