# Outsourcing LDA-Based Face Recognition to an Untrusted Cloud

Yanli Ren, Zhuhuan Song, Shifeng Sun, Joseph K. Liu, and Guorui Feng

**Abstract**—Face recognition has been extensively employed in practice, such as attendance system and public security. Linear discriminant analysis (LDA) algorithm is one of the most significant ones in the field of face recognition, but it is very difficult for many clients to employ it in their resource-constrained devices (e.g., smartphones and notebook computers). Outsourcing computation provides a promising method for clients to perform heavy tasks with limited computing power. In this paper, we design a protocol of outsourcing LDA-based face recognition to an untrusted cloud, which can help the client to complete the operations of matrix inversion (MI), matrix multiplication (MM) and eigenvalue decomposition (ED) simultaneously. The proposed outsourcing protocol can hide the private data of the client from the cloud. More importantly, the client can verify whether the outsourcing results are correct or not with probability one and so it is impossible for the server to deceive the client. In addition, the proposed protocol greatly decreases the computational complexity of the client thus enabling the client to complete LDA algorithm efficiently. Finally, we implement the protocol and give a comprehensive evaluation. The experimental results demonstrate that the client obtain great computing savings and the face recognition accuracy in the proposed protocol is almost identical to the original LDA algorithm.

**Index Terms**—Cloud computing, linear discriminant analysis, face recognition, secure outsourcing, privacy-preserving

## 1 Introduction

In the last two decades, owing to the fast development of deep learning and machine learning, face recognition has been employed in many fields, such as attendance system [1], public security [2] and identity authentication [3]. By using face recognition system, we can catch criminals to maintain social security. However, the algorithm of face recognition often suffers from high computational cost since it features a lot of matrix operations. As we know, an image is stored in the form of a matrix. For many small terminals, it is difficult for them to complete the whole process of face recognition due to their limited computing power.

Recently, the continuous improvement of cloud computing has made a lot of changes to the Internet. Cloud computing can help those devices with limited resources to complete computing tasks, and can also reduce their computing costs [4]. Function-as-a-Service (FaaS) is a kind of cloud computing services that allows the clients with restricted computing power to complete complex calculations [5]. Building an application following FaaS is one way of achieving a serverless architecture. FaaS is popular in on-demand services since it can be shut down and save the computational cost once it is not in use [6]. Therefore, many small-scale terminals prefer to outsource the collected face images to the cloud, and execute the function of face recognition with the help of the cloud server.

However, outsourcing computation will bring a series of security risks and challenges if the original data needed to be calculated is directly transmitted to the cloud. Some rogue servers may collect private information and carry out illegal activity [7], [8]. The first challenge is the disclosure of private data. The private information leaked may bring incalculable losses to the client. Therefore, the client should not only encrypt the original data, but also ensure that the outputs associated with the original data are well protected from the server. The second challenge is verifiability. As we know, the outsourcing services are provided by a third party which may not be completely trusted and return wrong computing results in order to save computing resources. Moreover, even if the third party is fully trusted, there may still be some calculation errors. In view of the above situations, a verification algorithm is desirable to enable the client to check the correctness of the outsourcing results. The third challenge is efficiency. The client needs to ensure that outsourcing computation can greatly reduce the computation costs and save computing in contrast to the original computation task. Otherwise, it is meaningless to execute outsourcing computation for a limited terminal. Therefore, a reliable protocol must be secure, verifiable and efficient.

Many secure outsourcing protocols have been proposed until now. In [9], Ren *et al.* proposed secure verifiable outsourcing protocol based on co-occurrence matrix for feature extraction, which used only one cloud server to complete the whole process. Feng *et al.* [10] proposed an outsourcing scheme of orthogonal tensor singular value decomposition (SVD) to compress high-order Big Data by using garbled circuits, which can be widely employed in the network security

- *Yanli Ren, Zhuhuan Song, and Guorui Feng are with the School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China. E-mail: {renyanli, songzhuhuan, grfeng}@shu.edu.cn.*
- *Shifeng Sun is with the School of Cyber Science and Engineering, Shanghai Jiao Tong University, China. E-mail: shifeng.sun@monash.edu.*
- *Joseph K. Liu is with the Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia. E-mail: joseph.liu@monash.edu.*

*Manuscript received 13 October 2020; revised 10 June 2021; accepted 28 April 2022. Date of publication 3 May 2022; date of current version 13 May 2023.*
*The work was supported in part by the National Natural Science Foundation of China under Grants U1736120 and 62072295, in part by the Natural Science Foundation of Shanghai under Grant 20ZR1419700 and 22ZR1481000, and in part by the Henan Key Laboratory of Network Cryptography Technology under Grant LNCT2021-A13.*
*(Corresponding author: Yanli Ren.)*
*Digital Object Identifier no. 10.1109/TDSC.2022.3172143*

and network forensics. Fu *et al.* [11] outsourced non-negative matrix factorization to a malicious cloud and solved security problems. The proposed scheme adopts Paillier homomorphism to protect images. Xia *et al.* [12] put forward an efficient protocol of outsourcing local binary pattern (LBP). In this protocol, the server can directly extract encrypted LBP features for the application. Zhu *et al.* [13] designed a novel framework of outsourcing location-based services to a semi-honest cloud, where query privacy, identity privacy and the verifiability of the outsourcing results are all well realized. He *et al.* [14] presented a secure searchable protocol, where the cloud can search over encrypted data. Zhao *et al.* [15] proposed a secure and verifiable computation protocol which used sparse matrices to protect the inputs and outputs. However, the above protocols can only be used to solve specific problems, and they cannot be applied to other problems, such as the outsourcing computation of face recognition.

There are many classical face recognition algorithms, such as eigenface algorithm [16], LBP algorithm [17] and LDA algorithm [18], [19]. By using these algorithms, we can easily complete the application of face recognition and facilitate our life. LDA algorithm has been commonly adopted in the field of machine learning, which can be employed to execute image classification and reduce image dimension. To address the matter of insufficient feature extraction, some improved LDA-based algorithms are introduced in [20], [21], and redefines the between-class scatter matrix. The client needs to calculate matrix inversion (MI), matrix multiplication (MM) and eigenvalue decomposition (ED) in the LDA algorithm, and the computational complexity of these steps is $O(n^3)$. It is very difficult for some terminal devices to do the corresponding calculations for large-scale face images, which may lead to the failures of face recognition. Therefore, it is essential and meaningful to use an outsourcing protocol to execute the LDA-based face recognition algorithm in practice.

Currently, some outsourcing protocols for face recognition have been proposed for decreasing the computational loads of the clients. In [22] and [23], Lei *et al.* proposed two outsourcing protocols of matrix inversion and matrix determinant. These two protocols employ matrix transformations on the original matrices to hide the private information of the images. In [24], Zhou *et al.* proposed two outsourcing schemes of ED and SVD and applied these to principal component analysis (PCA) algorithm, which is the first outsourcing protocol of ED and SVD. However, the malicious server can get the eigenvalues and eigenvectors by counting the greatest common divisor of the encrypted data. Zhang *et al.* [25] outsourced PCA-based face recognition, and mainly introduced two outsourcing protocols of matrix multiplication and eigenvalue decomposition. However, the client and the cloud needs to interact three times in these protocols, where the client should execute three encryptions and decryptions to protect the private data. To the best of our knowledge, no outsourcing algorithm for LDA-based face recognition has been proposed until now.

*Our Contributions*. We propose a non-interactive outsourcing protocol of LDA-based face recognition in this paper. The main contributions are as follows:

1) The client only needs one encryption and decryption to complete the whole outsourcing process. The computational complexity can be reduced from $O(n^3)$ to $O(n^2)$. The proposed protocol can both reduce communication costs and shorten computation time of the client. Compared with the previous ones, the protocol can outsource three kinds of matrix computations by only one encryption, which can greatly reduce interactions between the client and the cloud server.
2) The original inputs and the true outsourcing results are also private for the server. In detail, the inputs are unknown for the server by multiplying elementary matrices. The outsourcing results of our protocol are demonstrated to be computationally indistinguishable from a random vector and matrix, which means the outputs are also well protected.
3) The outsourcing results can be verified efficiently by the client. The client can verify whether the results are correct by using the verification algorithm, and the probability of verification successfully is $1-\frac{1}{C_n^k}$. With the increment of $n$ and $k$, the probability is infinitely close to 1, which is a non-negligible probability for the incorrect results to pass the verification algorithm.

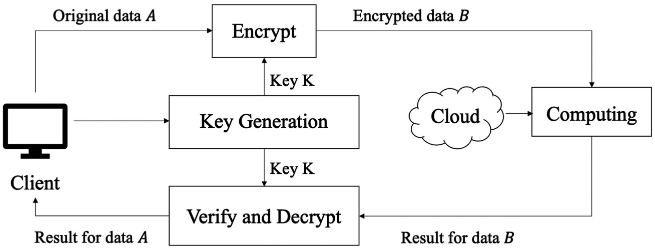


Fig. 1. Secure outsourcing computation model.

The rest of our paper is organized as follows. In Section 2, we first show the system model and framework, and then introduce the LDA-based face recognition and some theorems on eigenvalue decomposition. Section 3 presents the proposed outsourcing protocol of LDA-based algorithm. We prove that the proposed outsourcing protocol is secure, verifiable and efficient in Section 4. The experimental results are presented in Section 5. Finally, we conclude the paper in Section 6.

## 2 Models and Definitions

In this section, we first review the system model and framework of outsourcing computation, and then introduce the algorithm of LDA-based face recognition and some theorems on eigenvalue decomposition.

### 2.1 System Model and Framework

#### 2.1.1 System Model

We assume that the client needs to execute some heavy operations, but it is very difficult for him to complete the task because of restricted local computing power. For the client with limited ability, a potential solution is to outsource data to the cloud. As depicted in Fig. 1, the (sensitive) data is set as $A$. The client expects the cloud not to get any private information related to $A$, so he encrypts the data $A$ with a private key $K$ to get the encrypted data $B$.

Next, the client sends data $B$ to the cloud which performs the corresponding calculations on $B$. The client verifies the result of data $B$ after the cloud server sends it back. If yes, the client decrypts the encrypted result to obtain the computation result based on original data $A$; otherwise, the result is rejected.

#### 2.1.2 Threat Models

During the process of outsourcing computation, the client may face many security threats which mostly come from the malicious attacks of the cloud server. In general, there exists three kinds of threat models in the outsourcing computation [26].

*Lazy but Honest Model*. The cloud server will faithfully perform every step required by the client in this model, but it may return random results as the true ones for saving computation resource.

*Honest but Curious Model*. The cloud server will honestly perform every step and return the results to the client in this model. However, the cloud server may analyze data and attempt to collect some private information.

*Malicious Model*. The cloud server will not only randomly sends results to the client, but also want to collect some private information by analyzing data from the client. Clearly, the malicious model is the strongest one, and what we consider in this paper.

#### 2.1.3 Design Goals

The proposed outsourcing protocol needs to meet the design goals shown below.

*Correctness*. The client can finally acquire the true results through decrypting the encrypted results, if the cloud server and the client execute the outsourcing protocol faithfully.

*Privacy*. When the protocol runs, the cloud has no access to obtain the client's true data and the true computational results from the ciphertext and outsourcing results.

*Verifiability*. After receiving the results, the client utilizes the verification algorithm to verify whether it is true with a great probability and any incorrect results cannot pass the verification algorithm.

*Efficiency*. Compared with direct calculation, outsourcing computation can enormously decrease the computing overheads for the clients. Else, outsourcing these computation tasks to the cloud server is meaningless.

#### 2.1.4 Framework

The outsourcing protocol commonly includes the following algorithms in order to achieve the above design goals.

*Key Generation*$(1^\kappa)$. On input a security parameter $\kappa$, the algorithm generates a secret key $K$, which will be employed to encrypt the original data and decrypt the results returned from the cloud server.

*Encryption*$(A, K)$. The client adopts the secret key $K$ to encrypt the raw data $A$, and gets the ciphertext data $B$ which will be sent to the cloud server.

*Computation*$(B)$. The cloud server performs the computations on the ciphertext data $B$ and returns the results $\alpha$ to the client.

*Verification*$(\alpha)$. The client employs the verification algorithm to verify the computation results $\alpha$. If the results pass the verification algorithm, the client accepts them; otherwise, the client rejects them.

*Decryption*$(\alpha, K)$. The client adopts secret key $K$ to decrypt the verified results $\alpha$ and gets the final results $\alpha'$.

### 2.2 LDA-Based Face Recognition Algorithm

#### 2.2.1 A Brief Introduction of LDA Algorithm

LDA algorithm has been extensively employed in image compression, face recognition and feature extraction. It is often called the classic Fisher linear discriminant analysis method. According to [18], [19], [20], [21], we briefly introduce the steps of LDA-based face recognition algorithm.

Consider that there exists some samples of $M$ classes and the number of all samples is $N$. We assume that the set of samples in each class is $X_i$ and $N_j$ is the number of samples in each class, where $i \in [1, N]$ and $j \in [1, M]$. We also consider all the samples are $X = \{x_1, x_2, \ldots, x_N\}$. After projection, $Y_i$ is the set of samples in the class and $m_j$ is the average value of each class. According to the above data, we can do the calculation as follows.

Computing the average value of all the mean vectors:

$$m = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{1}{N}\sum_{j=1}^{M} N_j m_j. \tag{1}$$

The within-class scatter matrix can be calculated:

$$S_w = \sum_{i=1}^{M}\sum_{x\in X_i}(x - m_i)(x - m_i)^T. \tag{2}$$

The between-class scatter matrix can be computed:

$$S_b = \sum_{i=1}^{M} N_i(m_i - m)(m_i - m)^T. \tag{3}$$

Calculate the matrix S:

$$S = S_w^{-1} S_b. \tag{4}$$

Computing the eigenvalue $\sigma_i$ and the corresponding eigenvector $w_i$ of matrix $S$, and then selecting the eigenvectors corresponding to the $L(L \leq M - 1)$ maximum eigenvalues to form the projection matrix $W_{opt}$, and then making it satisfy the Fisher decision criteria:

$$W_{opt} = argmax\frac{|w^T S_b w|}{|w^T S_w w|}. \tag{5}$$

Finally, the test samples are projected on the projection matrix. When the images are classified, a distance measure classifier can be selected to recognize the face.

#### 2.2.2 Necessity of Outsourcing LDA Algorithm

In the algorithm of face recognition, the client needs to preprocess the input image, including histogram equalization, geometric normalization and other operations. Then the client needs to covert the two-dimensional image matrix into a column vector. However, the dimension may become very high during the conversion. For example, the column vector is $100 \times 100 = 10000$ dimension, while the dimension of within-class scatter matrix $S_w$ is $10000 \times 10000$ if a $100 \times 100$

face image is converted. In [22], [24] and [25], it has been demonstrated that local devices may cost a lot of time to do the MI, MM and ED with such a large-scale matrix. Therefore, it is difficult for the client with restricted computing power to execute the calculation locally.

To overcome the problem of LDA-based face recognition, we propose an outsourcing protocol of the LDA algorithm. The original inputs are the matrices $S_w$ and $S_b$, and then the cloud calculates the eigenvalues and eigenvectors of the matrix $S_w^{-1}S_b$, which will greatly reduce the local computation overhead of the client.

### 2.3 Eigenvalue Decomposition

Before introducing eigenvalue decomposition, we first introduce eigenvalues and eigenvectors of matrices. For a given real matrix $A \in \mathbb{R}^{n\times n}$, we assume a non-zero vector $x$ and a real number $\lambda$ can make Eq. (6) hold. Then we call $x$ and $\lambda$ are the eigenvector and eigenvalue of matrix $A$, separately.

$$Ax = \lambda x. \tag{6}$$

Eigenvalue decomposition decomposes a matrix into a product of its eigenvalues and eigenvectors as the form of:

$$AX = X\Lambda, \tag{7}$$

where $\Lambda \in \mathbb{R}^{n\times n}$ is a diagonal matrix and all its diagonal elements are the eigenvalues of matrix $A$, while each column of matrix $X \in \mathbb{R}^{n\times n}$ is the eigenvector of matrix $A$ [27].

Eigenvalues and eigenvectors have many applications in machine learning, such as data compression and latent semantic analysis and face recognition. According to [25] and [27], we introduce two theorems about eigenvalues and eigenvectors of matrices.

**Theorem 1.** *[27] : Let $G \in \mathbb{R}^{n\times n}$ be an upper or a lower triangular matrix, and $G_1, G_2, \ldots, G_n$ are the elements on the diagonal of $G$, and then the eigenvalues of $G$ is $G_1, G_2, \ldots, G_n$.*

**Theorem 2.** *[25] : Let $A \in \mathbb{R}^{n\times n}$ be a real matrix and the eigenvalues and corresponding eigenvectors of $A$ are $\lambda_1, \lambda_2, \ldots, \lambda_n$ and $x_1, x_2, \ldots, x_n$. Let $G \in \mathbb{R}^{n\times n}$ be an upper or a lower triangular matrix, and its eigenvalues $G_1, G_2, \ldots, G_n$ are the elements on its main diagonal and its corresponding eigenvectors are $g_1, g_2, \ldots, g_n$. If we let $O \in \mathbb{R}^{n\times n}$ be a zero matrix and then mark $V = \begin{pmatrix} A & O \\ O & G \end{pmatrix}$, the eigenvalues of $V$ are $\epsilon_1 = \lambda_1, \epsilon_2 = \lambda_2, \ldots, \epsilon_n = \lambda_n, \epsilon_{n+1} = G_1, \epsilon_{n+2} = G_2, \ldots, \epsilon_{2n} = G_n$, and the eigenvectors of $V$ are $v_1 = \begin{pmatrix} x_1 \\ o \end{pmatrix}$, $v_2 = \begin{pmatrix} x_2 \\ o \end{pmatrix}, \ldots, v_n = \begin{pmatrix} x_n \\ o \end{pmatrix}, v_{n+1} = \begin{pmatrix} o \\ g_1 \end{pmatrix}, v_{n+2} = \begin{pmatrix} o \\ g_2 \end{pmatrix}, \ldots, v_{2n} = \begin{pmatrix} o \\ g_n \end{pmatrix}$.*

*where $o \in \mathbb{R}^{n\times 1}$ is a zero vector.*

### 2.4 Computational Indistinguishability

The definition of computational indistinguishability [28] will be introduced in this subsection.

**Definition 1.** *Set $R \in \mathbb{R}^{n\times n}$ to be a random matrix with its elements in the $j$th column taken from a uniform distribution in the range of $[-R_j, R_j], \forall j \in [1, n]$. We say that matrices $Q$ and $R$ are computationally indistinguishable for each probabilistic polynomial time distinguisher $D$, if there is a negligible function $\mu$ making the following Eq. holds:*

$$\forall i, j, \left|Pr\left[D(q_{i,j}) = 1\right] - Pr\left[D(r_{i,j}) = 1\right]\right| < \mu \tag{8}$$

*where $q_{i,j}$ and $r_{i,j}$ are the elements in the $i$th row and $j$th column of matrices $Q$ and $R$, respectively. When the input is detected as a random matrix which is distributed uniformly between the range $[-R_j, R_j]$, distinguisher $D$ outputs one and zero otherwise.*

Definition 1 shows that a malicious adversary cannot distinguish the elements of matrix $Q$ from $R$. That is to say, it is impossible for the malicious adversary to get any effective information from matrix $Q$.

## 3 Non-Interactive Secure Outsourcing of LDA-Based Face Recognition

We propose a protocol of outsourcing LDA-based face recognition in this section. The client inputs an encrypted matrix, and then gets the eigenvalues and eigenvectors of the original matrix by using the proposed outsourcing protocol.

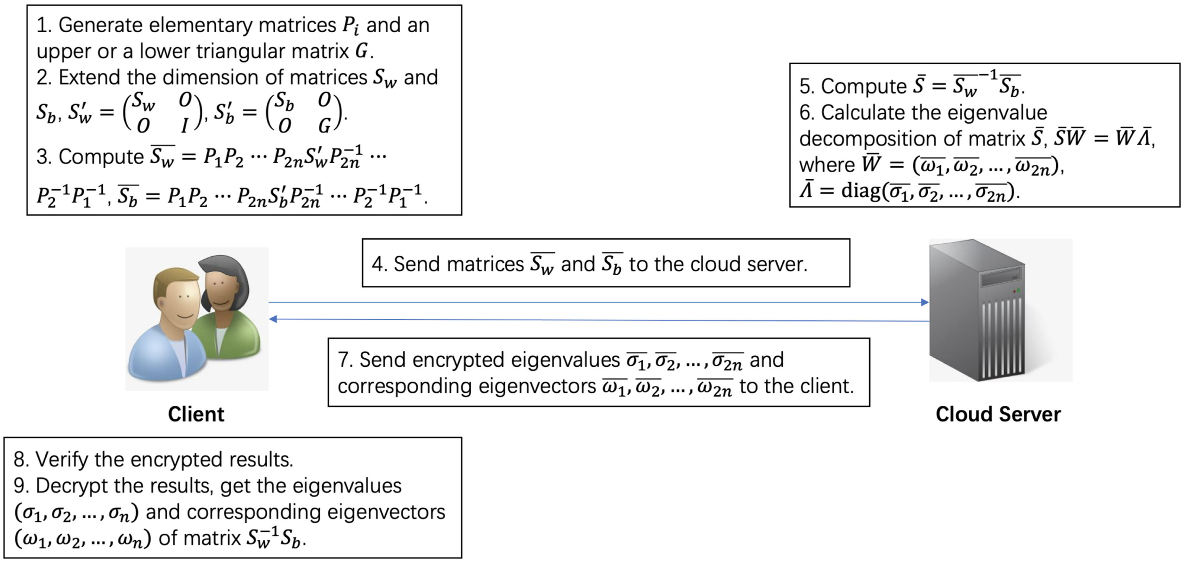
### 3.1 Non-Interactive Outsourcing of LDA-Based Algorithm

In an interactive outsourcing protocol, the client first encrypts the matrix $S_w \in \mathbb{R}^{n\times n}$ and sends the ciphertext of $S_w$. Then he decrypts the results returned by the cloud and gets $S_w^{-1}$. Second, the client submits the ciphertexts of matrices $S_w^{-1}$ and $S_b \in \mathbb{R}^{n\times n}$ and then decrypts the outsourcing results and gets $S = S_w^{-1}S_b$. Third, the client encrypts $S$ to get $\overline{S}$ and the server does eigenvalue decomposition (ED) to get the eigenvalues and eigenvectors of $\overline{S}$. At last, the client decrypts the results and acquires the eigenvalues and eigenvectors of $S_w^{-1}S_b$.

**Algorithm 1.** Elementary Matrix Generation

```
Input: The number $2n$.
Output: Elementary matrices $P_i, i = 1, 2, \ldots, 2n$.
 1: Set $\alpha_1 = 2n$ and $Y = \{1, 2, \ldots, 2n-1\}$.
 2: for $i = 1 : 2n$ do
 3:     Let $P_i$ be an identity matrix and $p_i$ be a random number
        in the interval $(-2^p, 2^p)$, where $p$ is a positive constant.
 4:     Set $\beta_i = i$, and if $\beta_i$ is in set $Y$, delete $\beta_i$ from $Y$.
 5:     if $i > 1$ then
 6:         Set $\alpha_i$ to be an integral number in set $Y$.
 7:     end if
 8:     Delete $\alpha_i$ from set $Y$.
 9:     Add $\beta_i$ to set $Y$ if $\beta_i$ is deleted at 4.
10:     Let the element in the $\beta_i$th row and the $\alpha_i$th column of
        matrix $P_i$ be $p_i$.
11: end for
```

During the whole outsourcing process, the client does three encryptions and decryptions, and executes three interactions with the server, which needs very high computational and communication costs. To decrease the communication and computational overloads for the client, a non-interactive outsourcing protocol of LDA-based algorithm is presented in this section. Fig. 2 shows the detailed process of the proposed outsourcing protocol.



1. Generate elementary matrices $P_i$ and an upper or a lower triangular matrix $G$.
2. Extend the dimension of matrices $S_w$ and $S_b$, $S'_w = \begin{pmatrix} S_w & O \\ O & I \end{pmatrix}$, $S'_b = \begin{pmatrix} S_b & O \\ O & G \end{pmatrix}$.
3. Compute $\overline{S_w} = P_1P_2\cdots P_{2n}S'_wP_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}$, $\overline{S_b} = P_1P_2\cdots P_{2n}S'_bP_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}$.

4. Send matrices $\overline{S_w}$ and $\overline{S_b}$ to the cloud server.

5. Compute $\bar{S} = \overline{S_w}^{-1}\overline{S_b}$.
6. Calculate the eigenvalue decomposition of matrix $\bar{S}$, $\bar{S}\bar{W} = \bar{W}\bar{\Lambda}$, where $\bar{W} = (\overline{\omega_1}, \overline{\omega_2}, \dots, \overline{\omega_{2n}})$, $\bar{\Lambda} = \text{diag}(\overline{\sigma_1}, \overline{\sigma_2}, \dots, \overline{\sigma_{2n}})$.

7. Send encrypted eigenvalues $\overline{\sigma_1}, \overline{\sigma_2}, \dots, \overline{\sigma_{2n}}$ and corresponding eigenvectors $\overline{\omega_1}, \overline{\omega_2}, \dots, \overline{\omega_{2n}}$ to the client.

8. Verify the encrypted results.
9. Decrypt the results, get the eigenvalues $(\sigma_1, \sigma_2, \dots, \sigma_n)$ and corresponding eigenvectors $(\omega_1, \omega_2, \dots, \omega_n)$ of matrix $S_w^{-1}S_b$.

Fig. 2. Specific flow chart of the proposed outsourcing protocol.

### 3.1.1 Key Generation

The client employs Algorithm 1 to obtain elementary matrices $P_i \in \mathbb{R}^{2n\times 2n}$.

We can get the structure of matrix $P_i$ according to Algorithm 1:

$$P_i = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & \vdots & \ddots & & \\ & p_i & \cdots & 1 & \\ & & & & 1 \end{pmatrix}, i = 1, 2, \dots, 2n. \tag{9}$$

According to this structure, the diagonal elements of $P_i$ are all 1, the element of row $\beta_i$ and column $\alpha_i$ is $p_i$ and the rest elements are all 0.

Since the matrix $P_i$ is an elementary matrix, it is easy for the client to get its inverse:

$$P_i^{-1} = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & \vdots & \ddots & & \\ & -p_i & \cdots & 1 & \\ & & & & 1 \end{pmatrix}, i = 1, 2, \dots, 2n. \tag{10}$$

We set that $p_i$ is a random variable chosen from the interval $(-2^p, 2^p)$, where $p$ is a positive constant, so its probability density function is presented in the following equation:

$$f_{P_i}(p_i) = \begin{cases} \frac{1}{2^{p+1}} & -2^p < p_i < 2^p \\ 0 & otherwise. \end{cases} \tag{11}$$

### 3.1.2 Encryption

The client first extends the dimension of matrix $S_w \in \mathbb{R}^{n\times n}$. According to Theorem 2 in Section 2.3, the client computes $S'_w = \begin{pmatrix} S_w & O \\ O & I \end{pmatrix}$, where $O \in \mathbb{R}^{n\times n}$ and $I \in \mathbb{R}^{n\times n}$ is a zero matrix and an identity matrix, respectively. Then according to Algorithm 1, the client generates $2n$ elementary matrices $P_i \in \mathbb{R}^{2n\times 2n}$ and then calculates:

$$\overline{S_w} = P_1P_2\cdots P_{2n}\begin{pmatrix} S_w & O \\ O & I \end{pmatrix}P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}. \tag{12}$$

Next, the client extends the dimension of matrix $S_b \in \mathbb{R}^{n\times n}$. According to Theorem 2, the client gets $S'_b = \begin{pmatrix} S_b & O \\ O & G \end{pmatrix}$, where matrix $G \in \mathbb{R}^{n\times n}$ is a triangular matrix randomly generated by the client and the elements on the diagonal of $G$ are $G_1, G_2, \dots, G_n$. Then the client uses the same elementary matrices $P_i$ and calculates:

$$\overline{S_b} = P_1P_2\cdots P_{2n}\begin{pmatrix} S_b & O \\ O & G \end{pmatrix}P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}. \tag{13}$$

Finally, the client sends matrices $\overline{S_w} \in \mathbb{R}^{2n\times 2n}$ and $\overline{S_b} \in \mathbb{R}^{2n\times 2n}$ to the cloud.

### 3.1.3 Computation

After receiving the matrices $\overline{S_w}$ and $\overline{S_b}$, the cloud computes $\overline{S} = \overline{S_w}^{-1}\overline{S_b}$ and eigenvalue decomposition of matrix $\overline{S}$, and then sends the eigenvalues $\overline{\sigma_i}$ and eigenvectors $\overline{w_i}$ of matrix $\overline{S}$ back.

### 3.1.4 Verification

After receiving the results, the client needs to verify whether the results are true or not.

Since the computational complexity of matrix-vector multiplication is $O(n^2)$, we design Algorithm 2 to decrease the computational complexity on the premise of correctness.

### 3.1.5 Decryption

The client can recover the eigenvectors of matrix $S_w^{-1}S_b$ according to the following equation:

$$w = P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}\overline{w}. \tag{14}$$

Next, the client computes the eigenvalues $\overline{\sigma_i}$. Theorem 1 proves that the eigenvalues of a triangular matrix $G$ are the elements on its diagonal. Since the client generates this triangular matrix $G$ and knows its diagonal elements, he only needs to remove the eigenvalues $G_1, G_2, \ldots, G_n$ of $G$, and the rest eigenvalues $\overline{\sigma_i}$ are the eigenvalues of matrix ${S_w}^{-1}S_b$.

**Algorithm 2.** Verification Algorithm

```
Input: The unchecked results $\overline{W} = (\overline{w_1}, \overline{w_2}, \ldots, \overline{w_{2n}})$ and $\overline{\Lambda} = diag(\overline{\sigma_1}, \overline{\sigma_2}, \ldots, \overline{\sigma_{2n}})$.
Output: Accept the results or not.
for $i = 1 : k$ do
    The client randomly chooses $\overline{w_i}$ and $\overline{\sigma_i}$ in $\overline{W}$ and $\overline{\Lambda}$.
    The client computes $U = \overline{S_b}\overline{w_i} - \overline{\sigma_i}\overline{S_w}\overline{w_i}$.
    if $U \neq (0, 0, \ldots, 0)^T$ then
        Reject $\overline{W}$ and $\overline{\Lambda}$, and return them to cloud.
    end if
end for
Take $\overline{W}$ and $\overline{\Lambda}$ as the correct results.
```

### 3.2 Summary of the Proposed Protocol

The proposed protocol is summarized as follows.

*Key Generation*. The client generates elementary matrices $P_i$, an identity matrix and an upper or a lower triangular matrix.

*Encryption*. The client encrypts matrix $S_w$ and $S_b$ as follows and sends $\overline{S_w}$ and $\overline{S_b}$ to the cloud.

$$\overline{S_w} = P_1P_2\cdots P_{2n}\begin{pmatrix} S_w & O \\ O & I \end{pmatrix}P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}$$

$$\overline{S_b} = P_1P_2\cdots P_{2n}\begin{pmatrix} S_b & O \\ O & G \end{pmatrix}P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}$$

*Computation*. The cloud computes $\overline{S_w}^{-1}$, $\overline{S} = \overline{S_w}^{-1}\overline{S_b}$, and ED of matrix $\overline{S}$, and then returns its encrypted eigenvalues and eigenvectors back.

*Verification*. The client randomly chooses $k$ pairs of eigenvalues and eigenvectors, and does the calculations. If the answers pass the verification algorithm, the client accepts them, or the client refuses them.

*Decryption*. The client uses (14) to obtain the eigenvectors and uses Theorem 2 to obtain the eigenvalues.

### 3.3 Correctness of the Outsourcing Protocol

As described above, the client encrypts matrices $S_w$ and $S_b$ and sends encrypted matrices $\overline{S_w}$ and $\overline{S_b}$ to the cloud server.

First, the cloud computes the inverse of $\overline{S_w}$. According to Eq. (12), $\overline{S_w}^{-1}$ is shown as follows:

$$\overline{S_w}^{-1} = P_1P_2\cdots P_{2n}\begin{pmatrix} S_w^{-1} & O \\ O & I \end{pmatrix}P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}. \tag{15}$$

Next, the cloud computes the multiplication of matrices $\overline{S_w}^{-1}$ and $\overline{S_b}$. The specific process is as follows:

$$\begin{aligned}\overline{S} &= \overline{S_w}^{-1}\overline{S_b} \\ &= P_1P_2\cdots P_{2n}\begin{pmatrix} S_w^{-1} & O \\ O & I \end{pmatrix}P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1} \\ &\cdot P_1P_2\cdots P_{2n}\begin{pmatrix} S_b & O \\ O & G \end{pmatrix}P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1} \\ &= P_1P_2\cdots P_{2n}\begin{pmatrix} S_w^{-1}S_b & O \\ O & G \end{pmatrix}P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}.\end{aligned} \tag{16}$$

Finally, the cloud computes ED of matrix $\overline{S}$, and returns the results back.

After the client receives the results, he randomly selects some pairs of eigenvalues and eigenvectors, and calculates the Eq. (6). The proposed verification algorithm can ensure the efficiency and correctness of verification.

Then the client decrypts the results. According to the relationships between matrices, eigenvalue and eigenvector mentioned in Section 2.3, we can get the following expression:

$$\overline{S}\overline{w} = \overline{\sigma}\overline{w}. \tag{17}$$

According to (16) and (17):

$$P_1P_2\cdots P_{2n}\begin{pmatrix} S_w^{-1}S_b & O \\ O & G \end{pmatrix}P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}\overline{w} = \overline{\sigma}\overline{w}. \tag{18}$$

Therefore, we get the following equation:

$$\begin{aligned}\begin{pmatrix} S_w^{-1}S_b & O \\ O & G \end{pmatrix}P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}\overline{w} &= P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}\overline{\sigma}\overline{w} \\ &= \overline{\sigma}P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}\overline{w}.\end{aligned} \tag{19}$$

Let $w = P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}\overline{w}$, we can get:

$$\begin{pmatrix} S_w^{-1}S_b & O \\ O & G \end{pmatrix}w = \overline{\sigma}w. \tag{20}$$

According to Theorem 2, the eigenvalues $\overline{\sigma}$ of matrix $\begin{pmatrix} S_w^{-1}S_b & O \\ O & G \end{pmatrix}$ are composed of the eigenvalues of matrices $S_w^{-1}S_b$ and $G$. The eigenvectors $w$ can be obtained by $w = P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}\overline{w}$.

## 4 Analysis of the Proposed Outsourcing Protocol

We analyze the proposed outsourcing protocol, and prove it is secure, verifiable and efficient in this section. In detail, we prove that the inputs and outputs are computational indistinguishable from a random matrix and vector. More importantly, the client can check the errors with a non-negligible probability. Besides, in terms of efficiency, the computational overhead of the client can be decreased from $O(n^3)$ to $O(n^2)$.

### 4.1 Security Analysis

*Input Privacy*. The inputs of the proposed protocol are matrices $S_w$ and $S_b$. As shown in Section 3, the client encrypts them as follows:

$$\overline{S_w} = P_1P_2\cdots P_{2n}\begin{pmatrix} S_w & O \\ O & I \end{pmatrix}P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}$$

$$\overline{S_b} = P_1P_2\cdots P_{2n}\begin{pmatrix} S_b & O \\ O & G \end{pmatrix}P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}.$$

The client multiplies the matrix by elementary matrices to conceal the private information of a matrix. Concretely, the matrix $Q \in \mathbb{R}^{2n\times 2n}$ can be hidden as follows:

$$\overline{Q} = P_1P_2\cdots P_{2n}QP_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}, \tag{21}$$

where $P_i \in \mathbb{R}^{2n\times 2n}(i = 1, 2, \ldots, 2n)$ are all elementary matrices as presented in Section 3. We suppose the elements in matrix $Q$ are set in the interval $(-2^r, 2^r)$, where $r$ is a positive constant.

To prove the privacy of matrix $\overline{Q}$, we need to disassemble it into the following two parts:

$$\widehat{Q} = P_1P_2\cdots P_{2n}Q, \tag{22}$$

$$\overline{Q} = \widehat{Q}P_{2n}^{-1}\cdots P_2^{-1}P_1^{-1}. \tag{23}$$

In [24], Zhou *et al.* proved that left multiply a matrix by an elementary matrix is identical to the row transformation of the matrix, and right multiply a matrix by an elementary matrix is identical to column transformation of the matrix. Then, after the transformation of (22), every element in matrix $\widehat{Q}$, denoted by $\widehat{q}_{i,j}$, can be calculated as:

$$\widehat{q}_{i,j} = q_{i,j} + p_iQ_{i,j}, \tag{24}$$

where $Q_{i,j} = q_{i',j}(i' \in [1, 2n], i \neq i')$, $q_{i,j}$ and $q_{i',j}$ are two different elements in the matrix $Q$ and $p_i$ is a random variable which is randomly chosen from the interval $(-2^p, 2^p)$ as shown in Section 3.1 in detail. Next, we set $Z_i$ to be the theoretical maximum of $\{Q_{i,j} \,|\, j = 1, 2, \ldots, 2n\}, \forall i \in [1, 2n]$. Therefore, $\widehat{q}_{i,j}$ is in the range of $(-2^r - 2^p \cdot Z_i, 2^r + 2^p \cdot Z_i)$. Thus, we can now draw a theorem with regard to the computational indistinguishability of matrix $\widehat{Q}$ and a random matrix where the row elements are all taken from a uniform distribution.

**Theorem 3.** *[25] : Set $R \in \mathbb{R}^{2n\times 2n}$ to be a random matrix with its elements in row $i$ taken from a uniform distribution in the range of $(-2^p \cdot Z_i, 2^p \cdot Z_i), \forall i \in [1, 2n]$, and $\widehat{Q}$ is defined as (22). We say that matrices $\widehat{Q}$ and $R$ are computationally indistinguishable.*

As mentioned above, since left multiply a matrix by an elementary matrix is identical to the row transformation of the matrix and right multiply a matrix by an elementary matrix is identical to column transformation of the matrix, (22) performs row operations and (23) performs column operations on a matrix. Followed by Theorem 3, we can propose Theorem 4 with regard to the computational indistinguishability of matrix $\overline{Q}$ and a random matrix where the column elements are taken from a uniform distribution.

We suppose that the elements in matrix $\widehat{Q}$ are valued between the interval $(-2^t, 2^t)$, where $t$ is a positive constant. According to (23), every element in matrix $\overline{Q}$, denoted by $\overline{q}_{i,j}$, can be calculated as:

$$\overline{q}_{i,j} = \widehat{q}_{i,j} - p_i\widehat{Q}_{i,j}, \tag{25}$$

where $\widehat{Q}_{i,j} = \widehat{q}_{i,j'}(j' \in [1, 2n], j \neq j')$, $\widehat{q}_{i,j}$ and $\widehat{q}_{i,j'}$ are two different elements in the matrix $\widehat{Q}$. Set $Z'_i$ to be the theoretical maximum of $\{Q_{i,j} \,|\, i = 1, 2, \ldots, 2n\}, \forall j \in [1, 2n]$. Therefore, $\overline{q}_{i,j}$ is in the range of $(-2^t - 2^p \cdot Z'_i, 2^t + 2^p \cdot Z'_i)$.

**Theorem 4.** *Set $R \in \mathbb{R}^{2n\times 2n}$ to be a random matrix with its elements in column $j$ taken from a uniform distribution in the range of $(-2^p \cdot Z'_i, 2^p \cdot Z'_i), \forall j \in [1, 2n]$, and $\overline{Q}$ is defined as (23). We say that matrices $\overline{Q}$ and $R$ are computationally indistinguishable.*

**Proof.** To prove Theorem 4, we should demonstrate that any $\overline{q}_{i,j}$ and $r_{i,j}(\forall i, j \in [1, 2n])$ are computationally indistinguishable for matrices $\overline{Q}$ and $R$ as shown in Definition 1. Concretely, we need to prove that it is impossible for any probabilistic polynomial time distinguisher $D$ to distinguish $\overline{q}_{i,j}$ from $r_{i,j}$, $\forall i, j \in [1, 2n]$, unless there exists a negligible probability. That means we have to calculate the specific probability of successful verification and prove this probability is infinity close to 1.

We have demonstrated that the values from matrices $\overline{Q}$ and $R$ are in the range of $(-2^t - 2^p \cdot Z'_i, 2^t + 2^p \cdot Z'_i)$ and $(-2^p \cdot Z'_i, 2^p \cdot Z'_i)$, respectively. Thus, when given a sampled $x = \overline{q}_{i,j}$, the best strategy for distinguisher $D$ is to return $b \leftarrow \{0, 1\}$ with the same chance if $-2^p \cdot Z'_i < x < 2^p \cdot Z'_i$ and one if $x \leq -2^p \cdot Z'_i$ or $x \geq 2^p \cdot Z'_i$. Thus, when $x = \overline{q}_{i,j}$, the success probability of the distinguisher can be computed as follows:

$$\begin{aligned}
&Pr[D(\overline{q}_{i,j})] = 1\\
&= \frac{1}{2}Pr\big[-2^p \cdot Z'_i < \overline{q}_{i,j} < 2^p \cdot Z'_i\big]\\
&\quad + Pr\big[\overline{q}_{i,j} \leq -2^p \cdot Z'_i\big] + Pr\big[\overline{q}_{i,j} \geq 2^p \cdot Z'_i\big]\\
&= \frac{1}{2}\big(1 - Pr\big[\overline{q}_{i,j} \leq -2^p \cdot Z'_i\big] - Pr\big[\overline{q}_{i,j} \geq 2^p \cdot Z'_i\big]\big)\\
&\quad + Pr\big[\overline{q}_{i,j} \leq -2^p \cdot Z'_i\big] + Pr\big[\overline{q}_{i,j} \geq 2^p \cdot Z'_i\big]\\
&= \frac{1}{2} + \frac{1}{2}Pr\big[\overline{q}_{i,j} \leq -2^p \cdot Z'_i\big] + \frac{1}{2}Pr\big[\overline{q}_{i,j} \geq 2^p \cdot Z'_i\big]
\end{aligned}$$

where

$$\begin{aligned}
Pr\big[\overline{q}_{i,j} \geq 2^p \cdot Z'_i\big] &= Pr\Big[\widehat{q}_{i,j} + p_i\widehat{Q}_{i,j} \geq 2^p \cdot Z'_i\Big]\\
&\leq Pr\big[2^t + |p_i|Z'_i \geq 2^p \cdot Z'_i\big]\\
&= Pr\left[|p_i| \geq 2^p - \frac{2^t}{Z'_i}\right]\\
&= \frac{2^{t-p}}{Z'_i}
\end{aligned}$$

Analogously, we acquire that $Pr[\overline{q}_{i,j} \leq 2^p \cdot Z'_i] \leq \frac{2^{t-q}}{Z'_i}$. Hence, the successful probability for distinguisher $D$ when $x = \overline{q}_{i,j}$ is shown in the following equation:

$$0 \ < \ Pr\big[D\big(\overline{q}_{i,j} = 1\big)\big] \leq \frac{1}{2} + \frac{2^{t-p}}{Z'_i}. \tag{26}$$

If $x = r_{i,j}$, we acquire that $Pr[D(r_{i,j} = 1)] = \frac{1}{2}$.

Based on (8), for any $i, j \in [1, 2n]$, the following equation can be achieved:

$$Pr\big[D\big(\overline{q}_{i,j} = 1\big)\big] - Pr\big[D\big(r_{i,j} = 1\big)\big] \leq \frac{2^{t-p}}{Z'_i}. \tag{27}$$

Notice that $Z'_i \geq 2^t$. Thus, we obtain:

$$\mu(p) = \frac{2^{t-p}}{Z'_i} \leq \frac{2^{t-p}}{2^t} = \frac{1}{2^p}, \tag{28}$$

which is a negligible function. Therefore, the proof is finished. □

According to Theorems 3 and 4, matrices $\overline{S_w}$ and $\overline{S_b}$ are computationally indistinguishable with a random matrix $R$. Therefore, we can say that the inputs $S_w$ and $S_b$ of the proposed protocol are secure and well protected.

*Output Privacy.* The cloud computes eigenvalue decomposition of the matrix $\overline{S}$ and sends the results back. The decryption process of eigenvectors is as follows:

$$w = P_{2n}^{-1} \cdots P_2^{-1} P_1^{-1} \overline{w}. \tag{29}$$

where $\overline{w}$ is the encrypted eigenvectors and $w$ is the decrypted eigenvectors. That is:

$$\overline{w} = P_1 P_2 \cdots P_{2n} w. \tag{30}$$

Suppose that the values of the elements in vector $w$ are in the interval $(-2^b, 2^b)$, where $b > 0$ and $b$ is a constant. Therefore, according to (30), every element in vector $\overline{w}$ can be denoted by $\overline{w_i}$:

$$\overline{w_i} = w_i + p_i w_i, \tag{31}$$

where $W_i = w_j$ or $\overline{w}_j$, and $w_i, w_j (j \in [1, 2n], i \neq j)$ are two different elements in vector $w$. Set $Z_i''$ to be the theoretical maximum of $W_i, \forall i \in [1, 2n]$. Therefore, $\overline{w}_i$ is in the range of $(-2^b - 2^p \cdot Z_i'', 2^b + 2^p \cdot Z_i'')$. We draw a theorem with regard to the computational indistinguishability of vector $\overline{w}$ and a random vector where the elements are sampled from a uniform distribution.

**Theorem 5.** *Set $r \in \mathbb{R}^{2n \times 1}$ to be a random vector with its elements in row i taken from a uniform distribution in the range of $(-2^p \cdot Z_i'', 2^p \cdot Z_i'')$ and $\overline{w}$ is defined as (30). Then we can say that vectors $\overline{w}$ and r are computationally indistinguishable.*

**Proof.** The proof is omitted here since it is similar to that of Theorem 4. □

According to Theorem 5, vector $\overline{w}$ is computationally indistinguishable from a random vector $r$, which means the cloud server cannot acquire any effective information of vector $\overline{w}$. Thus, vector $\overline{w}$ is well protected. In order to protect the eigenvalues $\sigma$, we use a random triangular matrix $G$, and set $V = \begin{pmatrix} S_w^{-1} S_b & O \\ O & G \end{pmatrix}$. According to the Section 2.3, the eigenvalues of matrix $V$ is composed of the eigenvalues of matrices $S_w^{-1} S_b$ and $G$. Since the cloud has no idea about the eigenvalues of matrix $G$, it cannot recover the eigenvalues of matrix $S_w^{-1} S_b$, which are hidden by the eigenvalues of the matrix $G$. With the image dimension expands, the dimension of matrix $G$ increases, and the number of eigenvalues also increases, so it is more and more difficult for the cloud to decrypt the eigenvalues. However, the client knows the elements of matrix $G$, and it is easy for the client to recovery the eigenvalues $\sigma$. Therefore, the eigenvalues and eigenvectors of matrix $S_w^{-1} S_b$ are secure and well protected.

### 4.2 Verification of the Outsourcing Results

In this subsection, we prove that the proposed outsourcing protocol meets robust cheating resistance [22], which means that the client can check the errors with a non-negligible probability.

**Theorem 6.** *The adversary can cheat the client to accept wrong outsourcing results with a negligible probability in the proposed protocol.*

**Proof.** There are two steps to prove this theorem.

First, we need to prove that any correct results can pass this verification algorithm successfully. As shown in (32), if the results are correct, $\overline{S_b}\overline{w_i}$ equals to $\overline{\sigma_i}\overline{S_w}\overline{w_i}$. Therefore, $U$ is always a zero vector and any correct results can pass this verification algorithm successfully.

$$U = \overline{S_b}\overline{w_i} - \overline{\sigma_i}\overline{S_w}\overline{w_i} \tag{32}$$

Second, we prove that any incorrect results cannot pass the verification algorithm with a non-negligible probability. Let *Prob* be the probability of verification successfully. According to the formula for permutation and combination, *Prob* can be estimated by:

$$Prob = 1 - \frac{1}{C_n^k}, \tag{33}$$

where $k$ is the number of selecting eigenvalues and eigenvectors in the verification algorithm and $n$ is the dimension of the matrix $\overline{S_w}^{-1}\overline{S_b}$.

Now we analyze why (33) holds. As shown in Section 3, the client will select $k$ from $n$ eigenvalues and eigenvectors, which means that the cloud could cheat the client with a probability of $1/C_n^k$. For example, the *Prob* is 0.99 when $n = 100$, $k = 1$ and it is $1 - 5 \times 10^{-5}$ when $n = 200$, $k = 2$. Therefore, with the increase of $N$ and $k$, the *Prob* will also be larger. According to our calculation, when $n = 1000$, $k = 5$, *Prob* is almost equal to 1, which means it is almost impossible for the cloud to cheat the client if 5 pairs of eigenvalues and eigenvectors are randomly chosen. Therefore, if there are some errors in the returned outsourcing results, it is impossible for the errors to pass our verification algorithm. □

According to the proof of Theorem 6, the probability for the wrong results to pass the verification algorithm is at most $1/C_n^k$. It is obvious that larger $k$ can make our verification algorithm more successfully, but it can also need more computational costs for the client. Therefore, $k$ effects the probability of verification and computational efficiency.

### 4.3 Efficiency Analysis

There are five algorithms in the proposed protocol. In this subsection, we separately analyze the overheads at client-side and cloud-side.

*Client-Side Overhead.* As discussed in Section 3, the client has to execute four algorithms including Key Generation, Encryption, Verification and Decryption. The client needs to generate a series of elementary matrices and an upper or a lower triangular matrix during the process of Key Generation. It is evident that the computational complexity is $O(n)$. In Encryption algorithm, the client expands the dimension of matrix and then multiplies the original matrix by elementary matrices, and its computational complexity is $O(n^2)$. In Verification algorithm, matrix-vector multiplication is the highest computational complexity which is $O(n^2)$. Finally, the client needs to calculate the formula (16) and obtain the true eigenvalues and eigenvectors in Decryption algorithm,

TABLE 1
Comparison of Computing and Communication Cost

| Protocol | Matrix Operations | Non-interactive | Client-side Overhead | Cloud-side Overhead | Input Privacy | Output Privacy | Verifiability |
|---|---|---|---|---|---|---|---|
| Zhou *et al.* [24] | ED | No | $O(n^2)$ | $O(n^3)$ | No | No | Yes |
| Zhang *et al.* [25] | MM, ED | No | $O(n^2)$ | $O(n^3)$ | Yes | Yes | Yes |
| Our protocol | MI, MM, ED | Yes | $O(n^2)$ | $O(n^3)$ | Yes | Yes | Yes |

and the computational complexity is also $O(n^2)$. In summary, the client-side overhead is $O(n^2)$.

*Cloud-Side Overhead.* Compared with the client, the cloud server only needs to run Computation algorithm. In this algorithm, the cloud computes matrix inversion (MI), matrix multiplication (MM) and eigenvalue decomposition (ED). As we know, the computational complexity of these steps are all $O(n^3)$. Therefore, the cloud-side overhead is $O(n^3)$.

Table 1 shows the comparison among the outsourcing protocols of face recognition. Though the computational overhead at the client-side and the cloud-side overhead in [24], [25] and our proposed protocol are all $O(n^2)$ and $O(n^3)$, respectively, the proposed protocol can outsource more kinds of matrix operations and have fewer times of communication overhead. The proposed outsourcing protocol can complete the operations of MI, MM and ED simultaneously, while [24] can only complete the operation of ED and [25] can only complete the operations of MM and ED. More importantly, the client needs only one encryption and one decryption to execute the outsourcing algorithm in the proposed protocol, which is a non-interactive protocol, while the client needs three encryptions and three decryptions in [24] and [25], which costs the client and the cloud server a lot of communication and computational overheads. The proposed protocol can help the client complete LDA-based face recognition more quickly and accurately. Therefore, using the proposed outsourcing protocol, the client can decrease both the computational complexity and communication overheads of the client.

## 5 Experiments

We have theoretically proven that the proposed protocol can largely reduce the communication and computational costs of the client in the previous section. In this section, the efficiency analysis and performance analysis on face recognition will be demonstrated from the following two experiments.

In the experiments, the client is simulated by MATLAB 2016a by a computer with 8 GB of RAM and Intel Core i5 rated at 1.8 GHZ, and the cloud server is simulated by a MacBook Pro laptop equipped with Intel Core i5 running with 4 cores rated at 1.4 GHz and 16 GB of RAM.

### 5.1 Efficiency Analysis of the Proposed Protocol

In this experiment, to show the efficiency of the proposed outsourcing protocol, we separately define $t_{original}$, $t_{client}$ as the time that the client executes the original LDA-based face recognition algorithm and the outsourcing protocol. According to the above parameters, the performance gain can be defined as $\frac{t_{original}}{t_{client}}$. Generally, the performance gain should be larger than 1, which means the client can save the computing resources, otherwise there is no gain for the client to choose outsourcing.

In Algorithm 1, $p_i$ is a random number chosen from the interval $(-2^p, 2^p)$, where $p$ is a positive constant. In this experiment, we set $p = 10$. As shown in Section 4.2, the verification probability becomes larger as the increase of parameter $k$, but larger $k$ will cost more computational overheads for the client. When $k = 10$, the successful probability is infinitely close to 1. Therefore, we do not need to use larger $k$ to increase the computational overheads of the client. In the following experiments, we will simulate the experiments for $k = 5$ and $k = 10$, respectively. In [24] and [25], Zhou *et al.* and Zhang *et al.* generate random matrices to test the efficiency of their proposed protocols. For a fair comparison, we also generate random matrices with different dimensions.

Tables 2 and 3 mainly present the performance gain of the proposed outsourcing protocol. The client can achieve at least 3.72 performance gain as shown in Table 2. With the expansion of dimension, the client can acquire more performance gain. The client can achieve 26.87 performance gain when the dimension of matrix is 5000. In Table 3, due to the increase of $k$, the client will cost more time on verification. When the dimension of matrix is 5000, the client can achieve 25.89 performance gain.

From Figs. 3, 4, and 5, we separately demonstrate the efficiency of the proposed protocol with those of the previous outsourcing ones and the original LDA-based algorithms. From Fig. 3, we conclude that the computational cost can be greatly decreased and the client can save more time by using the proposed protocol compared with the original algorithm. Next, we compare the cloud-side overhead with other outsourcing protocols as shown in Fig. 4. We find that our proposed protocol and Zhang *et al.* [25] spend more

TABLE 2
Experiment Results of the Outsourcing Protocol When $k = 5$

| No. | Dimension(n) | $t_{original}$(sec) | $t_{client}$(sec) | $t_{cloud}$(sec) | $t_{original}/t_{client}$ |
|---|---|---|---|---|---|
| 1 | 500 | 1.3951 | 0.3747 | 0.4468 | 3.72 |
| 2 | 1000 | 7.7163 | 1.1980 | 2.7853 | 6.44 |
| 3 | 1500 | 27.2441 | 2.7087 | 8.2763 | 10.05 |
| 4 | 2000 | 58.9292 | 4.6446 | 17.6197 | 12.69 |
| 5 | 2500 | 108.9184 | 6.7036 | 39.6343 | 16.25 |
| 6 | 3000 | 183.4635 | 9.2592 | 68.6919 | 19.81 |
| 7 | 3500 | 289.1965 | 13.0359 | 111.1952 | 22.18 |
| 8 | 4000 | 419.4783 | 17.4855 | 167.9263 | 23.99 |
| 9 | 4500 | 607.6743 | 23.8556 | 250.2278 | 25.47 |
| 10 | 5000 | 853.9909 | 31.7768 | 362.3223 | 26.87 |

TABLE 3
Experiment Results of the Outsourcing Protocol When $k = 10$

| No. | Dimension(n) | $t_{original}$(sec) | $t_{client}$(sec) | $t_{cloud}$(sec) | $t_{original}/t_{client}$ |
|---|---|---|---|---|---|
| 1 | 500 | 1.3503 | 0.3925 | 0.4323 | 3.44 |
| 2 | 1000 | 7.6304 | 1.2563 | 2.8711 | 6.07 |
| 3 | 1500 | 26.8636 | 2.8768 | 8.6601 | 9.34 |
| 4 | 2000 | 58.3099 | 4.8931 | 17.6027 | 12.12 |
| 5 | 2500 | 108.6013 | 7.0523 | 39.7892 | 15.40 |
| 6 | 3000 | 183.2340 | 9.6089 | 67.9081 | 19.07 |
| 7 | 3500 | 289.2011 | 13.5359 | 112.2060 | 21.37 |
| 8 | 4000 | 419.6159 | 18.2012 | 165.0293 | 23.05 |
| 9 | 4500 | 606.9187 | 24.7083 | 255.9527 | 24.56 |
| 10 | 5000 | 853.1093 | 32.9499 | 358.3951 | 25.89 |

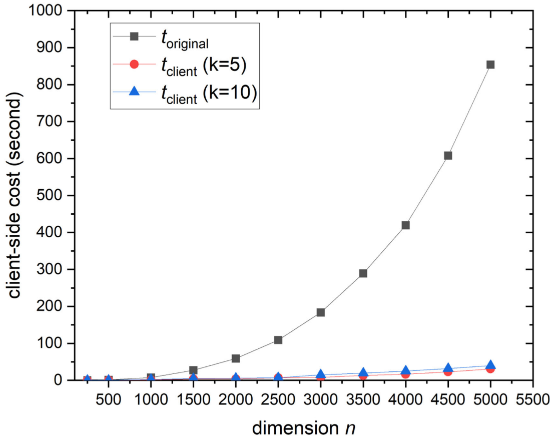


Fig. 3. Client-side time comparison for different $k$.

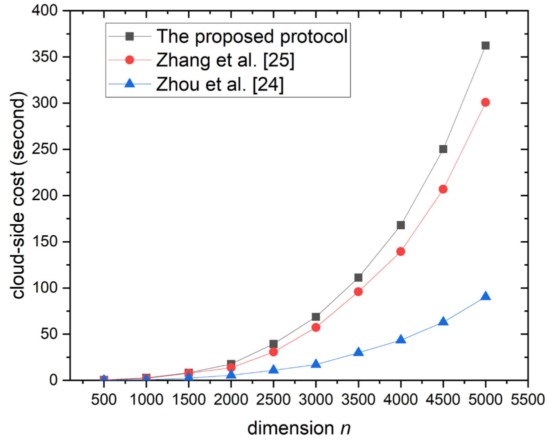


Fig. 4. Comparison of cloud-side time among the outsourcing protocols.

time for cloud-side overhead than Zhou *et al.* [24], since these two protocol extended the dimension of the matrix and can better protect the private information. Besides, our proposed protocol can outsource three kinds of matrix computations at the same time, while the other two can only outsource one kind of matrix computation. Thus, the proposed protocol cost a little more time than the other two at the cloud's side. Finally, we compare the performance gain with other outsourcing protocols in Fig. 5. It is apparent that the proposed outsourcing protocol can acquire more performance gains than those in Zhang *et al.* [25] and Zhou *et al.* [24], and they all outsource three operations, including matrix inversion, matrix multiplication and eigenvalue decomposition. More importantly, the proposed protocol is non-interactive, so the client can also save more communication costs.

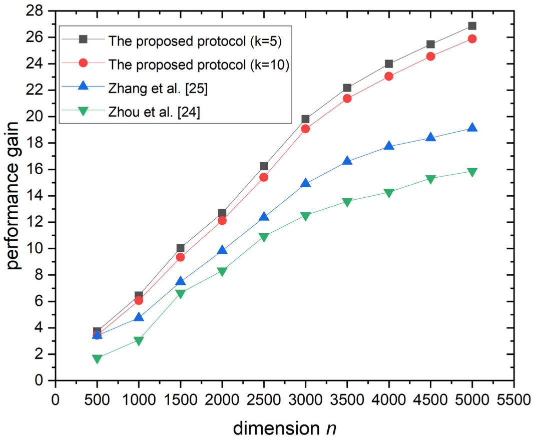


Fig. 5. Comparison of performance gain among the outsourcing protocols.

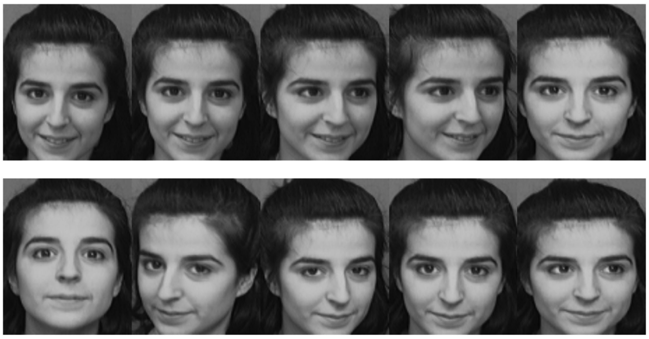


Fig. 6. The image samples of ORL database.

## 5.2 Performance Analysis on Face Recognition

In this experiment, we will demonstrate that the performance of face recognition in the proposed protocol is almost as same as that in the original LDA algorithm. The specific steps of the experiment are as follows.

### 5.2.1 Databases

To show the performance of the proposed outsourcing protocol, we apply the ORL Face Database [12], AR Face Dataset [29] and extended YaleB Face Database [30].

The ORL Face Database is made up of 400 face images of 40 persons and each person has 10 face images. The images contain some changes in posture, expression and facial accessories. All the samples of an object contain 10 normalized gray-scale images. The background of each image is black and its size is 92 × 112. The features and facial expressions of each person are changed, such as laughing or not laughing, wearing glasses or not and eyes closed or open. Some image samples of ORL Face Database are presented in Fig. 6.

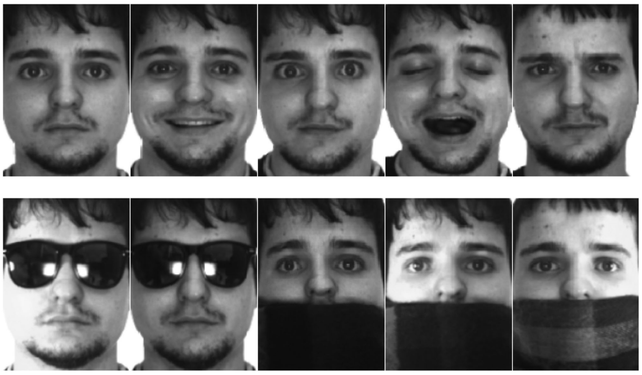


Fig. 7. The image samples of AR face database.

The AR Face Database is made up of more than 3000 images of 126 persons. In this paper, we select 100 persons, and each person with 26 images as the experimental database. The 26 images are collected from two periods. Each period contains 13 images, including 3 wearing sunglasses, 3 wearing scarves, and the other 7 images with light and expression changes. Some images of AR Face Database are presented in Fig. 7.

The extended YaleB Face Database is composed of 21888 images of 38 persons, which are collected from 9 postures and 64 illumination changes. In this paper, we select one of the subdatabases, which contains 38 persons and 64 images per person. We collect 64 kinds of illumination changes under the frontal posture. Some image samples of extended Yale Face Database B are presented in Fig. 8.

#### 5.2.2 Experimental Implementation

In this experiment, we use a two-stage method, which is an improved LDA-based face recognition [20]. First, the dimension of face image is reduced and the matrix $S_w$ is made full rank. Next, LDA algorithm is adopted to extract the image features and carry out face recognition.

According to different databases, we carried out 3 groups of experiments. Specifically, some face images are chosen as training samples, while the rest images are selected as test samples. Besides, the number of samples selected for each person is same. There is no overlap between two groups, which means the test samples are not included in the training samples. To eliminate the randomness of single sample selection, all the experiments were conducted 20 times, and we take the average recognition accuracy as the final recognition accuracy.

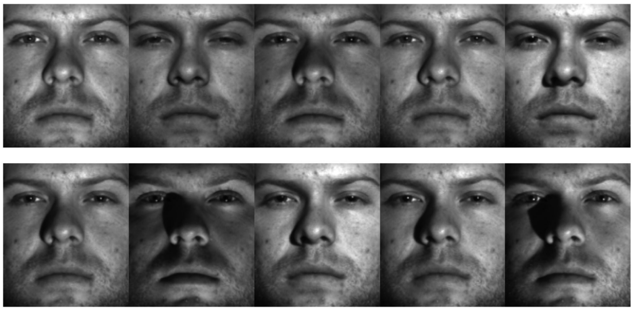


Fig. 8. The image samples of extended YaleB face database.

TABLE 4
Comparison of Accuracy Between the Outsourcing Protocol and the Original One

| Databases | Image Number | Accuracy (LDA-based) | Accuracy (Outsourcing) |
|---|---|---|---|
| ORL Database | 400 | 94.50% | 94.50% |
| AR Database | 2600 | 79.36% | 79.15% |
| extended YaleB | 21888 | 96.80% | 96.58% |

Then, we apply the proposed outsourcing protocol to face recognition algorithm. We conduct the experiment in the same way as above, and get the recognition accuracy by using the proposed protocol.

#### 5.2.3 Experimental Results

We first test the recognition accuracy of the original LDA-based face recognition, and then test the accuracy by using the proposed outsourcing protocol. The test data are presented in Table 4.

From Table 4, we can come to the conclusion that the face recognition accuracy in the proposed outsourcing protocol is nearly as same as that in the original LDA algorithm, which means the features extracted from the proposed protocol is correct and the proposed protocol can simultaneously achieve great cost savings for the client.

Thus, through the above experiments, we can draw two conclusions. First, the proposed protocol can acquire more performance gain than the previous ones. Next, the recognition accuracy of the proposed protocol is nearly as same as that in the original algorithm when it is employed to the LDA-based face recognition, which means that it is feasible and efficient in practice.

In summary, we show that the protocol can achieve great cost savings of computation and communication and keep almost the same accuracy as using LDA-based face recognition directly. Therefore, it is a better choice for a limited client to use the proposed outsourcing protocol to complete the LDA-based face recognition.

## 6 Conclusion

We design a protocol of outsourcing LDA-based face recognition to an untrusted cloud in this paper. By using the proposed protocol, the client needs only one encryption and one decryption to complete the operations of matrix inversion, matrix multiplication and eigenvalue decomposition, which greatly saves the interaction time and reduces the local computational complexity of the client. What's more, the privacy of the inputs and outputs can be hidden by multiplying a series of elementary matrices, which is proved to be secure. Besides, the proposed verification algorithm can check the errors with a non-negligible probability. Most importantly, our proposed protocol obtain more performance improvements compared with the previous outsourcing protocol of face recognition. However, the proposed outsourcing protocol can only be applied to LDA-based face recognition. Other outsourcing protocol of face recognition and machine learning will be researched in the future.

## REFERENCES

[1] M. Arsenovic, S. Sladojevic, A. Anderla, and D. Stefanovic, “Facetime – deep learning based face recognition attendance system,” in *Proc. IEEE Int. Symp. Intell. Syst. Inf.*, 2017, pp. 53–58.

[2] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang, and X. Yao, “Security and privacy preservation scheme of face identification and resolution framework using fog computing in Internet of Things,” *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1143–1155, Oct. 2017.

[3] C. Behaine and J. Scharcanski, “Enhancing the performance of active shape models in face recognition applications,” *IEEE Trans. Instrum. Meas.*, vol. 61, no. 8, pp. 2330–2333, Aug. 2012.

[4] X. Li, Y. Zhu, J. Wang, Z. Liu, Y. Liu, and M. Zhang, “On the soundness and security of privacy-preserving SVM for outsourcing data classification,” *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 906–912, Sep./Oct. 2018.

[5] J. Scheuner and P. Leitner, “Function-as-a-service performance evaluation: A multivocal literature review,” *J. Syst. Softw.*, vol. 170, 2020, pp. 110708.

[6] O. Ascigil, A. Tasiopoulos, T. K. Phan, V. Sourlas, I. Psaras, and G. Pavlou, “Resource provisioning and allocation in function-as-a-service edge-clouds,” *IEEE Trans. Services Comput.*, to be published, doi: 10.1109/TSC.2021.3052139.

[7] P. Anu and S. Vimala, “A survey on sniffing attacks on computer networks,” in *Proc. Int. Conf. Intell. Comput. Control*, 2017, pp. 1–5.

[8] M. Agarwal, S. Biswas, and S. Nandi, “Discrete event system framework for fault diagnosis with measurement inconsistency: Case study of rogue dhcp attack,” *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 3, pp. 789–806, May 2019.

[9] Y. Ren, X. Zhang, G. Feng, Z. Qian, and F. Li, “How to extract image features based on co-occurrence matrix securely and efficiently in cloud computing,” *IEEE Trans. Cloud Comput.*, vol. 8, no. 1, pp. 207–219, Jan.–Mar. 2020.

[10] J. Feng, L. T. Yang, G. Dai, W. Wang, and D. Zou, “A secure high-order lanczos-based orthogonal tensor SVD for Big Data reduction in cloud environment,” *IEEE Trans. Big Data*, vol. 5, no. 3, pp. 355–367, Sep. 2019.

[11] A. Fu, Z. Chen, Y. Mu, W. Susilo, Y. Sun, and J. Wu, “Cloud-based outsourcing for enabling privacy-preserving large-scale non-negative matrix factorization,” *IEEE Trans. Services Comput.*, vol. 15, no. 1, pp. 266–278, Jan./Feb. 2022.

[12] Z. Xia, L. Jiang, X. Ma, W. Yang, P. Ji, and N. N. Xiong, “A privacy-preserving outsourcing scheme for image local binary pattern in secure industrial Internet of Things,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 629–638, Jan. 2020.

[13] X. Zhu, E. Ayday, and R. Vitenberg, “A privacy-preserving framework for outsourcing location-based services to the cloud,” *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 1, pp. 384–399, Jan./Feb. 2021.

[14] K. He, J. Guo, J. Weng, J. Weng, J. K. Liu, and X. Yi, “Attribute-based hybrid boolean keyword search over outsourced encrypted data,” *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 6, pp. 1207–1217, Nov./Dec. 2020.

[15] L. Zhao and L. Chen, “Sparse matrix masking-based non-interactive verifiable (outsourced) computation, revisited,” *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 6, pp. 1188–1206, Nov./Dec. 2020.

[16] U. Devani, V. Nikam, and B. Meshram, “Super-fast parallel eigenface implementation on GPU for face recognition,” in *Proc. Int. Conf. Parallel Distrib. Grid Comput.*, 2014, pp. 130–136.

[17] B. Zhang, Y. Gao, S. Zhao, and J. Liu, “Local derivative pattern versus local binary pattern: Face recognition with high-order local pattern descriptor,” *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 533–544, Feb. 2010.

[18] R. Martín-Clemente and V. Zarzoso, “LDA via l1-PCA of whitened data,” *IEEE Trans. Signal Process.*, vol. 68, pp. 225–240, 2020.

[19] M. Akbar *et al.*, “An empirical study for PCA- and LDA-based feature reduction for gas identification,” *IEEE Sensors J.*, vol. 16, no. 14, pp. 5734–5746, Jul. 2016.

[20] J. Seng and K. Ang, “Big feature data analytics: Split and combine linear discriminant analysis (SC-LDA) for integration towards decision making analytics,” *IEEE Access*, vol. 5, pp. 14 056–14 065, 2017.

[21] Q. Ye, J. Yang, F. Liu, C. Zhao, N. Ye, and T. Yin, “L1-norm distance linear discriminant analysis based on an effective iterative algorithm,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 1, pp. 114–129, Jan. 2018.

[22] X. Lei, X. Liao, T. Huang, H. Li, and C. Hu, “Outsourcing large matrix inversion computation to a public cloud,” *IEEE Trans. Cloud Comput.*, vol. 1, no. 1, pp. 78–87, Jan.–Jun. 2013.

[23] X. Lei, X. Liao, T. Huang, and H. Li, “Cloud computing service: The caseof large matrix determinant computation,” *IEEE Trans. Services Comput.*, vol. 8, no. 5, pp. 688–700, Sep./Oct. 2015.

[24] L. Zhou and C. Li, “Outsourcing eigen-decomposition and singular value decomposition of large matrix to a public cloud,” *IEEE Access*, vol. 4, pp. 869–879, 2016.

[25] Y. Zhang, X. Xiao, L. Yang, Y. Xiang, and S. Zhong, “Secure and efficient outsourcing of PCA-based face recognition,” *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1683–1695, 2020.

[26] L. Jiang, C. Xu, X. Wang, B. Luo, and H. Wang, “Secure outsourcing SIFT: Efficient and privacy-preserving image feature extraction in the encrypted domain,” *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 1, pp. 179–193, Jan./Feb. 2020.

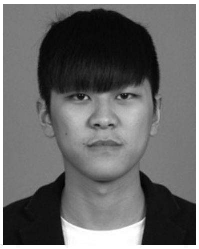
[27] X. Luciani and L. Albera, “Joint eigenvalue decomposition of non-defective matrices based on the LU factorization with application to ICA,” *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4594–4608, Sep. 2015.

[28] S. Salinas, C. Luo, X. Chen, W. Liao, and P. Li, “Efficient secure outsourcing of large-scale sparse linear systems of equations,” *IEEE Trans. Big Data*, vol. 4, no. 1, pp. 26–39, Mar. 2018.

[29] A. Martinez and A. Kak, “PCA versus LDA,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.

[30] A. Georghiades, P. Belhumeur, and D. Kriegman, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.

**Yanli Ren** received the MS degree in applied mathematics from Shaanxi Normal University, China, in 2005, and the PhD degree in computer science and technology from Shanghai Jiao Tong University, China, in 2009. She is currently a professor with the School of Communication and Information Engineering, Shanghai University, China. Her research interests include applied cryptography, secure outsourcing computation, blockchain security, AI security, and network security. She has published more than 80 quality papers, including publications in the *IEEE Transactions on Dependable and Secure Computing*, *IEEE Transactions on Information Forensics and Security*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Multimedia*, AsiaCCS, and INDOCRYPT, etc.



**Zhuhuan Song** is currently working toward the MS degree in the School of Communication and Information Engineering, Shanghai University, Shanghai, China. His research interests include applied cryptography, secure outsourcing computation, and privacy-preserving machine learning.

**Shifeng Sun** received the PhD degree in computer science from Shanghai Jiao Tong University, in 2016 and worked as a visiting scholar with the Department of Computing and Information Systems, University of Melbourne during his PhD study. He is currently an associate professor with the School of Cyber Science and Engineering, Shanghai Jiao Tong University, China. Prior to joining SJTU, he was a lecturer with the Department of Software Systems and Cybersecurity, Monash University, Australia. Before that, he worked as a research fellow with the Monash University and CSIRO, Australia. His research interest centers on cryptography and data privacy, particularly on provably secure cryptosystems against physical attacks, data privacy-preserving technology in cloud storage, and privacy-enhancing technology in blockchain. He has published more than 50 quality papers, including publications in ACM CCS, USENIX SEC, NDSS, EUROCRYPT, PKC, ESORICS, AsiaCCS, FC, the *IEEE Transactions on Dependable and Secure Computing*, and *IEEE Transactions on Knowledge and Data Engineering*, etc.

**Joseph K. Liu** received the PhD degree from the Chinese University of Hong Kong, in 2004. He is currently an associate professor with the Faculty of Information Technology, Monash University. Prior to joining Monash at 2015, he has worked as a research scientist with Institute for Infocomm Research (I2R) in Singapore for more than seven years. His research areas include cyber security, blockchain, IoT security, applied cryptography, and privacy enhanced technology. He has received more than 5,700 citations and his H-index is 43, with more than 170 publications in top venues such as CRYPTO, ACM CCS. He is currently the lead of the Monash Cyber Security Group. He has established the Monash Blockchain Technology Centre at 2019 and serves as the founding director.

**Guorui Feng** received the BS and MS degrees in computational mathematic from Jilin University, China, in 1998 and 2001, respectively, and the PhD degree in electronic engineering from Shanghai Jiaotong University, China, in 2005. From January 2006 to December 2006, he was an assistant professor with East China Normal University, China. During 2007, he was a research fellow with Nanyang Technological University, Singapore. Now, he is with the School of Communication and Information Engineering, Shanghai University, China. His current research interests include image processing, image analysis, and computational intelligence.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.