IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 3, 2019, accepted January 15, 2019, date of publication January 23, 2019, date of current version February 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2894535

A Privacy-Preserving Edge Computation-Based Face Verification System for User Authentication

XIANG WANG[1], HEYU XUE[1], XUEFENG LIU[1], AND QINGQI PEI[2], (Member, IEEE)

[1]State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071, China
[2]Shaanxi Key Laboratory of BlockChain and Security Computuing, Xian 710071, China

Corresponding author: Xiang Wang (wangxiang@xidian.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0800601, in part by the Key Basic Research Plan in Shaanxi Province under Grant 2017ZDXM-GY-014, and in part by the Key Program of NSFC-Tongyong Union Foundation under Grant U1636209.

ABSTRACT The face recognition has become a common means of identity authentication because of the advantages of uniqueness, non-invasive and not easy to be stolen. The outsourcing of face recognition to the service provider is a typical manner nowadays. However, it raises vital concerns about the privacy of outsourcing server due to the sensitivity of face data. Therefore, a frame of identity authentication based on the technology of privacy-preserving face recognition is presented in this paper. The convolutional neural network is used for face feature extraction. To overcome the issue of privacy leaked, a method of secure nearest neighbor that can compute the cosine similarity over encrypted feature vectors is proposed. What's more, the edge computing is introduced in our frame to increase the authentication efficiency by removing some operations from the cloud to the edge of the Internet. Moreover, we also propose a secret sharing homomorphism technology which is used for distributed computing to improve the fault-tolerance of our identity authentication system. The experimental results show that the proposed schemes are secure and effective.

INDEX TERMS Face recognition, edge computing, privacy protection.

I. INTRODUCTION

In recent years, biometrics identification such as face, iris, fingerprints and DNA receive a significant attention, especially in the field of human identification and authentication [1]. Compared with the traditional password-based authentication, biometrics has the the advantages of uniqueness, distinctive, mobility, user friendliness and not being transferable [2]. Face verification, as the most popular technology in biometrics, is widely used in authentication systems because its non-invasive and not needing the cooperation of users when scanning face images compared with the identification technology of fingerprint and iris [3]. However, in addition to the advantages mentioned above, there are also many challenges about the technology of biometrics identification such as privacy and security. It is a very typical choice to perform face verification by the cloud service providers such as face++ [4], but in this situation, one must upload the face image to the cloud server of service providers. Face image is extremely sensitive information, for it contains much privacy. Therefore, the issue such as face information could be collected, analyzed and misused without the permission of the users arose if uploading the face image without encrypting. Therefore, the emphases of researches are focused on how to achieve face recognition in a privacy-preserving manner [5]. The issues that related to the privacy-preserving of biometric recognition can be easier if being solved with edge computing [6]. Nowadays, fog and edge computing have received considerable attention from the areas of science and research. In the scenario mentioned above, the privacy protection provided by the cloud server is limited, and it can not be guaranteed that the security of the privacy data uploaded to the cloud server from being violated. However, the advantage of edge computing allows the edge computing nodes to process important data such as encryption so that it could be avoided unloading the sensitive data to the cloud server directly. Thus, the introduction of edge computing in our authentication system based on face verification can meet the privacy-preserving demand easier compared with traditional cloud server model.

In this paper, we proposed a frame of identity authentication based on technology of privacy-preserving face recognition in which secure nearest neighbor scheme and secret sharing is used for protecting the privacy of face information and DeepID [17] based on convolutional neural network (CNN) is used for face feature extraction. What's more, edge computing is introduced in our system for sharing some tasks of traditional cloud server, because edge computing can offload some computing tasks from the cloud and perform some privacy-preserving operations by the preprocessing. In addition, if some operations that can be completed within the edge computing, the numbers of interactive between the users and cloud server can be decrease greatly. Therefore, the introduction of edge computing in our authentication system based on privacy-preserving face verification can improve our efficiency compared with traditional cloud server model.

The main contributions of this paper can be summarized as follows:

- We for the first time use edge computing for authentication system based on privacy-preserving face verifications and it can effectively reduce the numbers of interactive between the users and cloud. The experiment result shows that the efficiency of our system is better than traditional cloud computing in some special scenarios.
- A secure nearest neighbor scheme is proposed to protect the face feature vector in edge computing servers, which can enable the edge computing server to carry out face verification directly with the encrypted face feature vectors without decrypting them.
- We propose a scheme of secret sharing homomorphism for dividing the face feature vectors into n secret shadows and storing in n servers respectively. The technology of homomorphism guarantees the sum of results computed by any t of n edge computing servers with their secret shadows equals to the result computed by the original feature vectors. Furthermore, the secret sharing homomorphism algorithm we employed in this paper can make our system enjoy the advantages of availability and fault-tolerance.

The remainder of the paper is organized as follows. The related works are presented in Section 2. Section 3 introduces the details of the system model, threat models, design goals and the background knowledge. The proposed method is described in Section 4. Section 5 presents the security analysis of our system. Section 6 evaluates the performance of the proposed scheme. Finally, we draw conclusions in Section 7.

II. RELATED WORKS

In recent years, many privacy-preserving methods for biometric data recognition were proposed. The combination of cryptographic primitives such as homomorphic encryption and garbled circuits are mostly used for protect the biometric data in these methods. A secure protocol for privacy preserving biometric identification was proposed in [7] that can achieve security against semi-honest adversaries. The security model is designed by the scheme of homomorphic encryption, oblivious transfer and garbled circuit. Hamming distance and Euclidean distance are used for measuring the similarity of biometric information. A privacy-preserving face recognition scheme was given in [8] that improved the performance of computation and communication efficiency. Eigenfaces biometric recognition is adopted to realize face recognition. The distance computation is achieved by Homomorphic Encryption and distance comparison is by garbled circuit. Bringer *et al.* [9]–[11] described several kinds of privacy-preserving biometric identification and all operations about biometric data are in the encrypted domain to prevent from privacy leaks. In [9], a protocol for outsourcing identification of encrypted biometric data to untrusted server was proposed, a new method of oblivious RAM was adopted for iris recognition and proved to be effective. The proposal only rely on standard symmetric encryption technology. The result shows that it is the only one that can deal with large databases and utilize all the opportunities of cloud computing. Two methods of privacy-preserving computation about Hamming distance were presented in [10], the first one is based on 1-out-of-2 Oblivious Transfer which can achieve full security in the semi-honest adversaries but one-sided security in the malicious adversaries, and the second one is an improvement on the first one that based on Committed Oblivious Transfer which can achieve full security against malicious adversaries. In [11], the privacy-preserving computation method was further improved that can be used for computing Euclidean distance, Hamming distance, Mahalanobis distance and scalar product of different biometric traits such as iris, face and fingerprint.

The technique of secure multiparty computation is also often adopted to protect the privacy in biometric data recognition and more than one cloud servers are usually used in these methods. A secure outsourced face recognition method under federated environment was given in [12] that can efficiently protect privacy data of the users under the semi-honest model. Face recognition based on the Eigenfaces algorithm is performed by two semi-bonest and non-colluding cloud servers in a privacy-preserving manner. Paillier cryptosystem is used in designing the secure model which can guarantee the result of the system keep same as that in the standard Eigenfaces algorithm. Reference [13] also proposed a secure face-verification system, deep neural networks is used to extract the face features and two servers are involved in this system, a data server stores the encrypted face features of the user and a verification server is used for performing face verification. Paillier encryption is used for protecting the face features and all data is transmitted in ciphertext, so no parties can decrypt it except for the verification server. A secure multiparty computation technique about face recognition was also employed in CloudID [14], the combination of homomorphic encryption and garbled circuits are used to carry out the encrypted face recognition without decrypting. A k-d tree structure is proposed in CloudID which can not only help to handle the biometrics variations in encrypted domain, but also improve the recognition efficiency of the system.

Moving from biometric recognition to privacy-preserving content-based image retrieval (PCBIR) that the technology which is similar to the privacy-preserving biometric data recognition. Xia *et al.* [15] implemented an scheme of PCBIR outsouring with an honest-but-curious cloud server. A secure

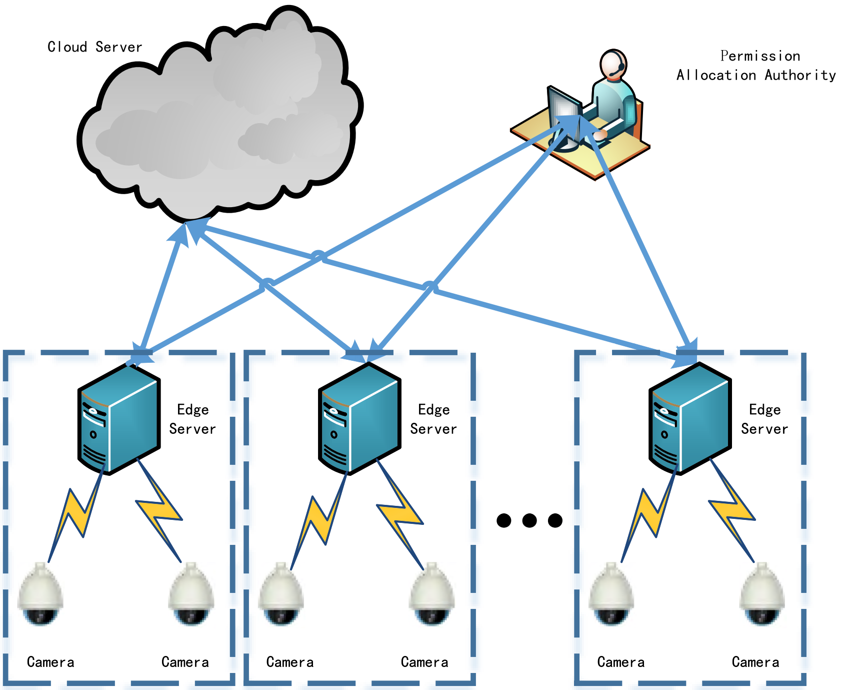


FIGURE 1. System overview.

nearest neighbor scheme was proposed to protect the feature vectors and enable the cloud server to compared the Euclidian distance of encrypted feature vectors efficiently without the additional communication burdens. The typical visual descriptors which are defined in MPEG-7 are employed for feature extraction and technology of locality-sensitive hashing is adopted to cluster the similar images that helps to improve the search efficiency greatly. Xu *et al.* [16] firstly proposed a method of orthogonal decomposition for PCBIR in cloud environment that the image was divided into two different component that encryption and feature extraction are performed separately. Therefore, what can be accessed by the cloud server provider are features of encrypted images and it can compare them with queried images without decrypting them. One difference from other methods is that this method has no special requirements to encryption algorithms.

III. PROBLEM STATEMENT

In this section, the system model and overview, threat model, design goals and background will be described in detail.

A. SYSTEM MODEL

The face privacy protection and authentication system we proposed in this paper involves three different types of entities: the cloud server (it refers to the service provider), n edge computing servers connected with several cameras and the permission allocation authority, as illustrated in Figure 1. The cameras are used for collecting face images of the users in our system. The permission allocation authority in our system is responsible for distributing permission for all the users registered in our system. The cloud server and edge computing servers work together to complete the identity registration and authentication.

B. SYSTEM OVERVIEW

In our application scenario, each edge computing server connected with several cameras arranged in a building will achieve identity registration and authentication. The cameras is responsible for collecting face images of the user who want to register or authentication on the edge computing server it connected, and then send it to this edge computing server. In the stage of identity registration, besides the face image, what sent to the edge computing server is still a encrypted vector of permission (only the cloud server can decrypt it) of the people who wants to register on this edge computing server. This permission vector (recorded as L) is n bits distributed by the permission allocation authority. Every element of the vector L represents the permission about those n edge computing servers of the people who want to register. If $L[i] = 1$, this people has the right to enter the building where the ith edge computing server is located, otherwise the people can not get this right. On received the face image, a feature vector f is extracted from the face image based on the SDK (DeepID-based methods which is described in section 3.E) that the cloud service provider installed on this edge computing server. First, this feature vector is protected by secure nearest neighbor algorithm and stored in the local database of the edge computing server. Next, n secret shadows of the face feature vector are generated by the edge computing server using a special (t, n) threshold secret sharing scheme. Then, these n secret shadows are encrypted with a standard encryption algorithm such as Advanced Encryption

System (AES) where each pair of any two edge servers shares a unique key and uploaded to the cloud server along with the encrypted permission vector. In our system, each edge server also shares a unique key with cloud server for secure communication so that it can effectively prevent adversaries from disguising himself as an edge server and and sending false information to the cloud server. Key establishment can be achieved by traditional key agreement protocols (e.g. Diffie-Hellman) whereby two or more parties can agree on a key in such a way that both influence the outcome. Finally, cloud server stored these contents and distribute the ciphertext (encrypted plaintext) of n shares among n edge computing server and make sure that each edge computing server get one secret shadow. In our system, each edge computing server can carry out authentication. In the stage of request for authentication, the face image collection and feature vector extraction are similar to those in the stage of identity registration. First, it should be searched in this edge computing server whether the people is registered locally. If not, this edge computing server will split this face feature vector with the method of (t, n) threshold secret sharing scheme and cooperate with any other $t-1$ edge computing servers and cloud server to get the permission of people who request for authentication.

C. THREAT MODEL

In our face privacy protection and authentication system, we consider the cloud server is semi-trusted or say ‘‘honest-but-curious’’, which means that the cloud server will carry out the protocol what you specify but it will analyse and keep watching all over the intermediate data generated when the protocol executed on the cloud server. What's more, we suppose that n edge computing server are non-colluding and independent with each other. Each edge computing server not only store the face vectors registered on it, but also store secret shadows (generated by secret sharing scheme) of the face feature vectors registered on other edge computing servers respectively. Therefore, each edge computing server will try to get the information of face feature vectors by the secret shadows what they store. In addition, we also consider the adversary maybe invade the cloud server or some of n edge computing servers. Therefore, we should design a scheme that can protect the privacy of the face data in the cloud server and edge computing servers.

In our system, if a people request for authentication on an edge computing server that is not the one he registered on. It is inevitable that the edge computing server can get the face feature vector and the permission on it, but other information about this people such as name, ID, the permission on other edge computing servers and the server he registered on are remain secret. This type of information leakage is not considered in our paper.

D. DESIGN GOALS

In this paper, we design a system used for face privacy protection and authentication based on edge computing. We achieve face recognition with the CNNs-based method and ensure the security of face feature vectors by (t, n) threshold secret sharing scheme and secure nearest neighbor scheme. To implement the model we mentioned above, several goals should be achieved in our system:

1) PRIVACY-PRESERVING

The technology of Biometric-based recognition is a very popular method in the field of authentication. However, biometric data is extremely sensitive, so we need to guarantee the user's privacy and confidentiality from being infringed when biometric data is stored and processed. Firstly, all the face vectors unloaded to the cloud server need to be kept secret to it in our system. And then, we must guarantee that edge computing server can not get any information about the face images in the case of only knowing the secret shadows of face feature vectors. Furthermore, if the adversary invade the cloud server and obtain the data, he can not get any information about face images. In term of the edge computing server, if being invaded, the information of users is still keep secret unless t edge computing servers are invaded at the same time.

2) AVAILABILITY AND FAULT-TOLERANCE

Each edge computing server store the secret shadows of all vectors that generated by (t, n) threshold secret sharing scheme on other edge computing servers. We must ensure that these edge computing servers can cooperate to complete the task of face recognition using the secret shadows they stored and can not get any privacy information only by these secret shadows. For fault-tolerance, we should make sure that any t of n edge computing servers can work together to carry on authentication.

3) EFFICIENCY

Because of Edge computing we introduced in our system has a certain amount of computing and storage ability, we can reduce the computation and communication cost by some preprocessing procedures on the edge computing server before the data upload to the cloud server. What's more, due to the introduction of edge computing, the amount of interaction with the server can be reduced accordingly.

E. BACKGROUND

1) FACE VERIFICATION

Face verification is the task of determine whether two face images belong to the same person. Face feature extraction is the first step of face verification. In early researches, many classical methods appeared, such as SIFTs, LBPs, and Gabor features. In recent years, CNNs-based methods have proven to be more efficient for face feature extraction. One of the most representative methods is DeepID proposed by Sun *et al.* [17]. The structure of network in DeepID is shown in Figure 2. The network contains four convolutional layers and four max-pooling layers. Through this network, a 160-dimensional feature vectors can be extracted from a face image. In this paper, method of DeepID is used to extract facial feature vector.

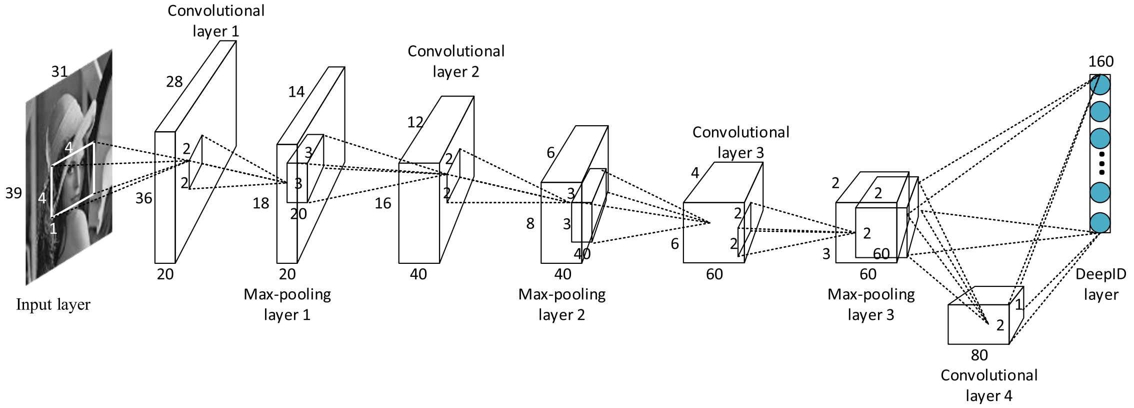


FIGURE 2. DeepID ConvNet structure.

We use the cosine similarity technique for face verification based on DeepID features. Cosine similarity is more efficient in metric learning problem than Euclidean distance. Let x and y are two vectors, cosine similarity can be defined as:

$$COS(x,y) = \frac{x^T y}{\|x\|\|y\|} \tag{1}$$

The calculation results of cosine similarity are confined within the range of -1 and 1. The larger the value of the result is, the higher the similarity between the two vectors is. A threshold should be chosen to determine whether two vectors are belong to same person or not.

2) SECRET SHARING

Secret Sharing is an effective method to protect important information from being stolen by means of distributed storage. With secret sharing, we can distribute secret shadows of a secret among a set of participants by giving each participant a secret shadow in such a way that only a particular subset can cooperate to recover the secret while any unqualified subset can not obtain any information about the secret. The (t, n) threshold secret sharing scheme in [18] and [19] is one of the most widely used algorithms for secret sharing. In this scheme, the dealer divides the secret into n secret shadows and shares them among n participants in such a way that at least t or more participants can reconstruct the secret, but any $(t-1)$ or fewer participants can obtain nothing about the secret. ''Fault-tolerance'' and ''availability'' are the most obvious advantages of secret sharing. For instance, if one participant is absent, other participants can also reconstruct the secret successfully. The specific algorithm of (t, n) threshold secret sharing is described as follows.

Secret Distribution: suppose S is a secret of the dealer, he can choose a random polynomial $f(x)$ of the degree $(t-1)$ with the only restriction that $f(0) = S$ and then choose n non-zero element $x_i(1 \leq i \leq n)$ that are different from each other. Next the dealer will compute the secret shadows $y_i = f(x_i)(1 \leq i \leq n)$ and (x_i, y_i) will be distributed to the participant P_i in which x_i is public among n participants but y_i as the sub-secret of the participant will not be public.

Secret Reconstruction: any t or more participants can cooperate to complete the secret reconstruction. Assume that $(x_i, y_i)(1 \leq i \leq n)$ are the secret shadows of k participants and they want to cooperate to complete the secret reconstruction. Secret reconstruction is based on the algorithm of Lagrange interpolation polynomial, the calculation method is as follows:

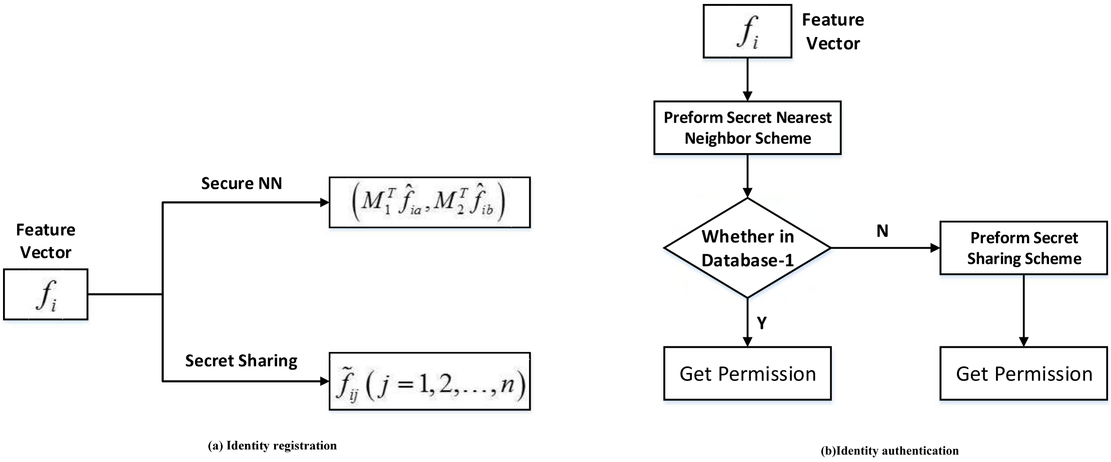
$$f(x) = \sum_{i=1}^{k} y_i \prod_{j=1, j\neq i}^{k} \frac{x - x_j}{x_i - x_j} \tag{2}$$

The free coefficient of this interpolation polynomial is the secret that we want to reconstruction, that is to say $f(0) = S$.

In this paper, we use secret sharing homomorphism technology [20], [21] which is a trick in secret sharing. For example, let A, B be two secrets that are shared among n participants, and any t participants can construct A or B by using Shamir's (t, n) threshold secret sharing scheme. However, what they should do to reconstruct $A+B$ without revealing the information about A and B? It is not hard to see that $A + B$ is linear operations, so if each of the t participants calculates the sum of the two secret shadows he holds, each of these sums itself is a secret shadow of the sum of two secrets $A + B$. It is also to say that if $f_1(x)$ and $f_2(x)$ encode A and B, then $g(x) = f_1(x) + f_2(x)$ encodes $A + B$. Therefore, to calculate the sum of two secrets $A + B$, each participate P_i who holds the secret shadows $f_1(x_i)$ and $f_2(x_i)$ should compute $g(x_i) = f_1(x_i) + f_2(x_i)$ and these t participants can reconstruct $A + B$ by using the algorithm of Lagrange interpolation polynomial.

IV. PROPOSED METHOD

In this section, a face privacy protection and authentication system is proposed. There are two stages in our system and the overview of proposed method is shown in Figure 3. The Figure 3.(a) describes the process of identity registration, on getting the feature vector f_i of the user who request for identity registration, the edge computing server will firstly perform the secure nearest neighbour algorithm to encrypt the feature vector f_i into $(M_1^T \hat{f}_{ia}, M_2^T \hat{f}_{ib})$ and store the encrypted feature vector in its database-1. The purpose of this operation is that the users can perform authentication on the edge


Feature Vector
f_i
Secure NN
$\left(M_1^T \hat{f}_{ia}, M_2^T \hat{f}_{ib}\right)$
Secret Sharing
$\tilde{f}_{ij}\,(j=1,2,\ldots,n)$
f_i
Feature Vector
Preform Secret Nearest Neighbor Scheme
Whether in Database-1
N
Y
Preform Secret Sharing Scheme
Get Permission
Get Permission

(a) Identity registration

(b)Identity authentication

FIGURE 3. Overview of proposed method.

computing server where they registered in without any other interactions with cloud server. Then n secret shadows $\tilde{f}_{ij}(1 \leq j \leq n)$ are generated by f_i with the method of secret sharing homomorphism and are sent to n edge computing servers through cloud server in a privacy-preserving manner, this operation makes sure that users can perform authentication on the edge computing except the one they registered in. Meantime, the cloud server can get the encrypted permission vector L from the permission allocation authority. The Figure 3.(b) presents the process of identity authentication. Firstly, we search in the database-1 of the edge computing server to find whether the feature vector f_q of the user who request for identity authentication can match a feature vector in database-1, if it is, we can get the permission of the user immediately, otherwise, this edge computing server need to cooperate with any other $(t-1)$ edge computing servers and cloud server to get the permission of this user by the scheme of secret sharing homomorphism. The detailed processes and methods are described as follows.

A. IDENTITY REGISTRATION

All users in this system should register first on the edge computing server which he belongs to before using authentication. In the stage of identity registration, a face image is picked by the camera and then will be sent to the edge computing server. Afterwards, a 160-dimensional feature vector $f_i = (f_{i,1}, f_{i,2}, \ldots, f_{i,160})^T$ is extracted from this image based on the face recognition network. At the same time, what sent to this edge computing server is still a encrypted permission vector L_i of n bits distributed by the permission allocation authority. The n elements of the vector L_i represent the permission of this user about those n edge computing servers.

1) PROTECT FACE PRIVACY WITH SECURE NEAREST NEIGHBOR ALGORITHM

Face feature vector can reveal some information of face image content. Therefore, the plaintext of the face feature vector can not be stored without protecting, effective encryption method must be employed to protect the plaintext of face feature vector from being stolen by adversary. In this paper, the secure nearest neighbor algorithm is used as our encryption method that can not only protect the privacy of face feature vector, but also guarantee the ciphertext of the feature vectors can be used for face recognition. The most obvious advantage of secure nearest neighbor algorithm is that it doesn't need high computation complexities or communication burden compared with the homomorphic encryption algorithm. A schematic of this algorithm is depicted in Figure 4. On getting the face image, the edge computing server will firstly extract a feature vector from it, then perform the secure nearest neighbor algorithm to encrypt the feature and store the ciphertext in its database-1. In the meantime, the permission allocation authority will encrypt the permission vector of this user in a asymmetric cryptographic algorithm (only the cloud server can decrypt it) and then send it to the edge computing server in a privacy-preserving manner.

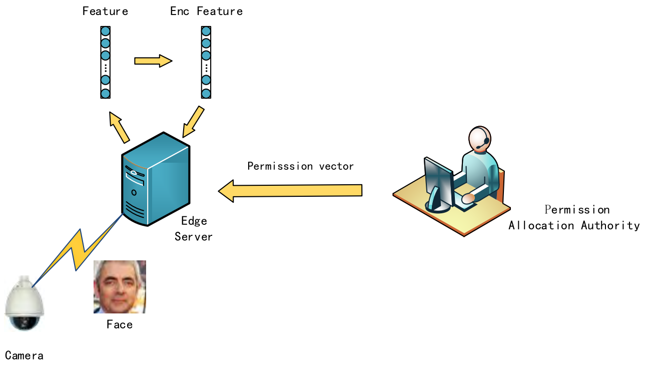


FIGURE 4. The overview of secure nearest neighbor algorithm.

The key of encryption algorithm contains a bit string S of 160 bits and two 160 × 160 invertible matrices M_1, M_2 that are shared among all edge computing servers and specific process of the algorithm is as follows.

1) Modify the feature vector $f_i = (f_{i,1}, f_{i,2}, \ldots, f_{i,160})^T$ of the user who request for identity registration into

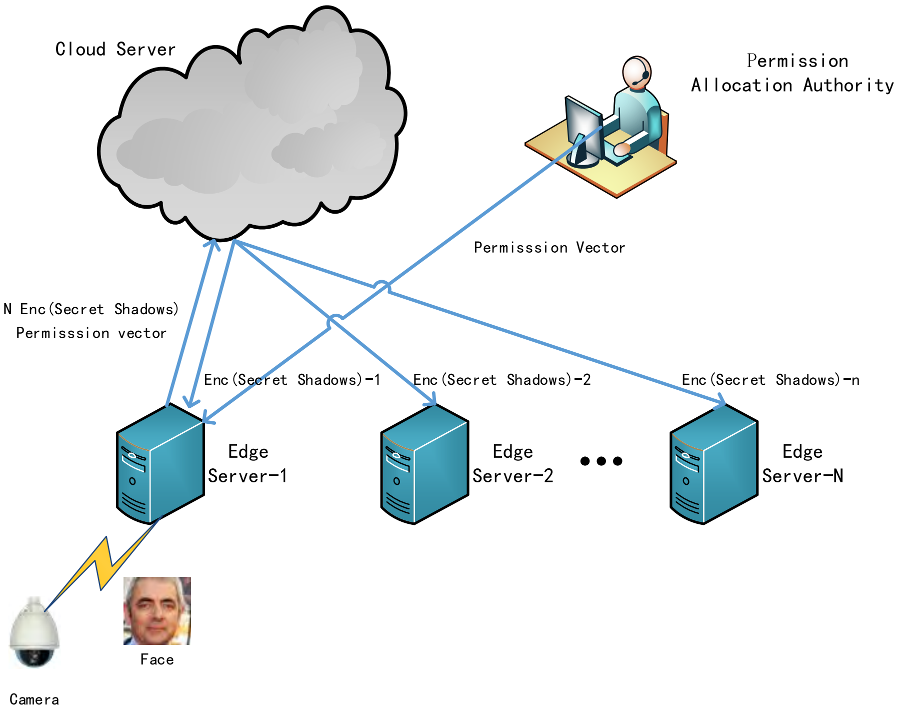


FIGURE 5. The overview of protect face privacy with secret sharing homomorphism algorithm.

$\hat{f}_i = (\frac{f_{i,1}}{\|f_i\|}, \frac{f_{i,2}}{\|f_i\|}, \ldots, \frac{f_{i,160}}{\|f_i\|})^T$ in which $\|f_i\|$ is the Euclidean norm of f_i.

2) Generate a pare of random vectors $(\hat{f}_{ia}, \hat{f}_{ib})$ with $\hat{f}_i$ and S. Notice that, for $j = 1$ to 160, if $S[j] = 0$, both $\hat{f}_{ia}[j]$ and $\hat{f}_{ib}[j]$ are set equal to $\hat{f}_i[j]$, if $S[j] = 1, \hat{f}_{ia}[j]$ is set to be a random value and then make $\hat{f}_{ib}[j]$ equal to $\hat{f}_i[j] - \hat{f}_{ia}[j]$.
3) Compute the result of $(M_1^T \hat{f}_{ia}, M_2^T \hat{f}_{ib})$ as our encrypted feature vector.

After being encrypted with the secure nearest neighbor algorithm, the ciphertext of the face feature vector will be stored in the database-1 of this edge computing server. The database-1 of each computing server is used for storing the face information of the users who registered on it. The purpose of doing this is to use the benefits of edge computing. Notice that, if a user request for authentication on an edge computing server, we can firstly search whether his face feature vector can be matched in the database-1 of this edge computing server, if so, we can get his permission immediately without any other interactions with the cloud server and other edge computing servers.

2) PROTECT FACE PRIVACY WITH SECRET SHARING HOMOMORPHISM ALGORITHM

Based on the process of the previous section, we can only achieve the goal that the users can perform authentication on the edge computing server which they register on. However, we must guarantee that the users can do authentication on all of edge computing servers. Therefore, the method based on the (t, n) threshold secret sharing is proposed in this section to solve this problem. A schematic of this method is depicted in Figure 5. The edge computing server which performs identity registration will firstly extract a feature vector from the face image collected by the camera. Then n secret shadows are generated by it with the method of secret sharing homomorphism algorithm. Finally, this edge computing server sends these secret shadows to the other $n-1$ edge computing servers through the cloud server in a privacy-preserving manner and make each edge computing server get one secret shadow which will stored in its database-2. The specific process of the algorithm is as follows.

1) Generate a 160-dimensional symbol vector R_i according to the face feature vector $f_i = (f_{i,1}, f_{i,2}, \ldots, f_{i,160})^T$ which we get in section 4.A. Notice that, for $j = 1$ to 160, if $f_{i,j} \geq 0$, the value of $R_i[j]$ is set equal to 1; if $f_{i,j} < 0$, the value of $R_i[j]$ is set equal to -1.
2) Modify the feature vector $f_i = (f_{i,1}, f_{i,2}, \ldots, f_{i,160})^T$ into $\tilde{f}_i = (\tilde{f}_{i,1}, \tilde{f}_{i,2}, \ldots, \tilde{f}_{i,160})$ in which $\tilde{f}_{i,j} = \log \left| \frac{f_{i,j}}{\|f_i\|} \right|$ and $\|f_i\|$ is the Euclidean norm of f_i.
3) Divide the vector $\tilde{f}_i$ into n different secret shadow vectors $\tilde{f}_{i1}, \tilde{f}_{i2}, \ldots, \tilde{f}_{in}$ and for $k = 1$ to 160 $\tilde{f}_{i1}[k], \tilde{f}_{i2}[k], \ldots, \tilde{f}_{in}[k]$ are the secret shadows of $\tilde{f}_i[k]$. Firstly, for $k = 1$ to 160 randomly choose 160 polynomial $f_k(x)$ of degree $t-1$ in the finite field $GP(p)$, in which p is a large prime with $p > n$ and only set the free coefficient of $f_k(x)$ equal to $\tilde{f}_i[k]$. n distinct integers $x_1, x_2, \ldots, x_n$ should be selected in the finite field $GP(p)$ for n edge computing servers in advance and public among all the edge computing servers. Then, compute $\tilde{f}_{ij}[k] = f_k(x_j)(1 \leq j \leq n)$ for the j-th edge computing server E_j.

Finally, we can get n vectors $\tilde{f}_{i1}, \tilde{f}_{i2}, \ldots, \tilde{f}_{in}$ as the n secret shadows of $\tilde{f}_i$.

4) Encrypt these n secret shadows with the algorithm of AES and then upload them to the server along with the encrypted permission vector L_i and the symbol vector R_i. Next, after storing what it received, server will send these n secret shadows to the edge computing server they belong to. Finally, for $j = 1$ to n, j-th edge computing server get the ciphertext of $\tilde{f}_{ij}$, then decrypt it and store it in his database-2.

The above is the whole process of our identity registration and then we can ensure that all user can perform authentication on any edge computing server. The specific process of identity authentication will be described in detail next.

B. IDENTITY AUTHENTICATION

As described in section 3.A, the process of identity authentication in our system includes two stages. If a user request for identity authentication on an edge computing server, his face image will be picked and sent to this edge computing server. Then, a 160-dimensional feature vector $f_q = (f_{q,1}, f_{q,2}, \ldots, f_{q,160})^T$ is extracted from the image based on the face recognition network.

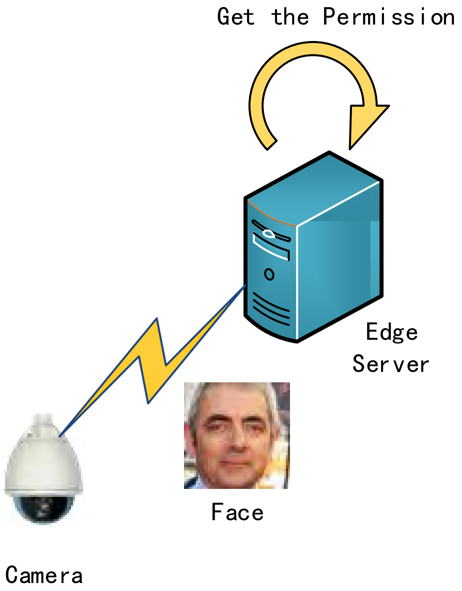


FIGURE 6. The overview of the step one in identity authentication.

1) STAGE ONE

A schematic of the process of stage one is depicted in Figure 6. In the first stage of the identity authentication, this edge computing server searches in its database-1 using the method of secure nearest neighbor to find whether the user is registered on it, if so, we can get his permission about this edge computing server immediately. The specific calculation process is as follows.

1) Modify the feature vector $f_q = (f_{q,1}, f_{q,2}, \ldots, f_{q,160})^T$ of the user who requests for identity registration into $\hat{f}_q = (\frac{f_{q,1}}{\|f_q\|}, \frac{f_{q,2}}{\|f_q\|}, \ldots, \frac{f_{q,160}}{\|f_q\|})^T$ in which $\|f_q\|$ is the Euclidean norm of f_q.

2) Based on the $\hat{f}_q$ and S, generate a pare of random vectors $(\hat{f}_{qa}, \hat{f}_{qb})$ and for $j = 1$ to 160, if $S[j] = 0, \hat{f}_{qa}[j]$ is set to a random value and then $\hat{f}_{qb}[j]$ is set equal to $\hat{f}_q[j] - \hat{f}_{qa}[j]$, if $S[j] = 1$, both $\hat{f}_{qa}[j]$ and $\hat{f}_{qb}[j]$ are set equal to $\hat{f}_q[j]$.

3) Compute the value of $(M_1^{-1}\hat{f}_{qa}, M_2^{-1}\hat{f}_{qb})$ and then the cosine similarity between the f_q and the feature vectors stored in the database-1 can be calculated, take the f_q and f_i for instance, the calculation process is as follows:

$$\begin{aligned} COS(f_q, f_i) &= (M_1^{-1}\hat{f}_{qa})^T M_1^T \hat{f}_{ia} + (M_2^{-1}\hat{f}_{qb})^T M_2^T \hat{f}_{ib} \\ &= (\hat{f}_{qa})^T \hat{f}_{ia} + (\hat{f}_{qb})^T \hat{f}_{ib} \\ &= (\hat{f}_q)^T \hat{f}_i \\ &= \frac{f_q^T f_i}{\|f_q\| \|f_i\|} \end{aligned} \tag{3}$$

4) Compare the values calculated above with the threshold t of our face verification, if $COS(f_q, f_i) \geq t$, then we say f_q and f_i are two features of one person. Therefore, the permission of him can be got without any other operations.

Identity authentication can be completed if the user's feature vector can be found in the database-1 of an edge computing server. Otherwise, Stage two should be performed to find whether the user is register on the other edge computing servers and get the permission.

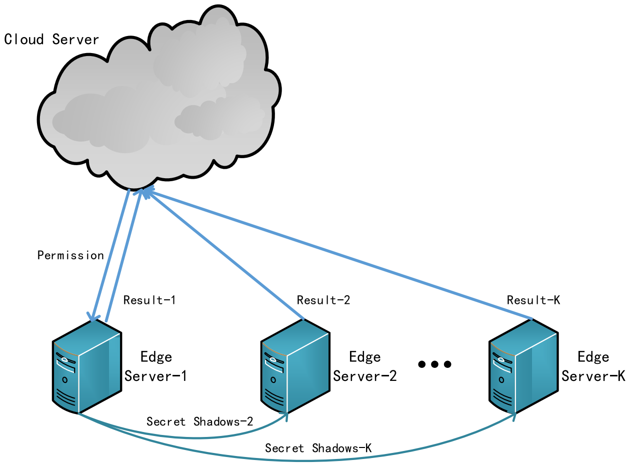


FIGURE 7. The overview of the step two in identity authentication.

2) STAGE TWO

A schematic of the process of stage two is depicted in Figure 7. The edge computing server firstly divides the face feature vector into t secret shadows with the (t, n) threshold secret sharing algorithm and then cooperate with any t edge computing servers and the cloud server to get the user's permission about this edge computing server. The specific process of stage two is as follows.

1) Generate a 160-dimensional symbol vector Q according to the plus-minus sign of the elements in the face feature vector $f_q = (f_{q,1}, f_{q,2}, \ldots, f_{q,160})^T$. If $f_{q,j} \geq 0$, set the value of $Q[j]$ equal to 1 and if $f_{q,j} < 0$, set the value of $Q[j]$ equal to -1.
2) Modify the feature vector $f_q = (f_{q,1}, f_{q,2}, \ldots, f_{q,160})^T$ into $\tilde{f}_q = (\tilde{f}_{q,1}, \tilde{f}_{q,2}, \ldots, \tilde{f}_{q,160})$ as: for $j = 1$ to 160, $\tilde{f}_{q,j} = \log\left|\frac{f_{q,j}}{\|f_q\|}\right|$ and $\|f_q\|$ is the Euclidean norm of f_q.
3) Randomly select any t edge computing servers to cooperate to complete face recognition. For the convenience of description, we choose first t computing servers as our examples. Firstly, divide the vector $\tilde{f}_q$ into t shadow vectors $\tilde{f}_{q1}, \tilde{f}_{q2}, \ldots, \tilde{f}_{qt}$ with the method of Shamir's (t, n) threshold secret sharing scheme. For $k = 1$ to 160 $\tilde{f}_{q1}[k], \tilde{f}_{q2}[k], \ldots, \tilde{f}_{qt}[k]$ are the t secret shadows of $\tilde{f}_i[k]$. Here, the $x_j (1 \leq j \leq t)$ of these t edge computing servers keep same as those in section 4.A.2. Then, the polynomial $g_k(x)$ of degree $t - 1$ is also chosen randomly in the finite field $GP(p)$ and the free coefficient of $g_k(x)$ is set equal to $\tilde{f}_q[k]$. Next, compute $\tilde{f}_{qj}[k] = g_k(x_j)(1 \leq j \leq n)$ for the j-th edge computing server E_j. Finally, t vectors $\tilde{f}_{q1}, \tilde{f}_{q2}, \ldots, \tilde{f}_{qt}$ can be got as t secret shadows of $\tilde{f}_q$ and the secret shadow $\tilde{f}_{qj}$ is sent to corresponding edge computing server E_j.
4) These t edge computing servers compute their vectors U_j^i by what they received. For $k = 1$ to 160, $U_j^i[k]$ can be computed by the edge computing servers $E_j(1 \leq j \leq t)$ as follows:

$$U_j^i[k] = (\tilde{f}_{qj}[k] + \tilde{f}_{ij}[k]) \prod_{l=1, l \neq j}^{t} \frac{-x_l}{x_j - x_l} (modp) \quad (4)$$

5) These t edge computing servers upload U_j^i they computed above to the cloud server and what's more the symbol vector Q generated by the edge computing server that requests for identity authentication is also uploaded. Then, the cosine similarity between f_q and f_i can be computed by the cloud server computing as follows:

$$\begin{aligned} COS(f_q, f_i) &= \sum_{k=1}^{160} R_i[k]Q[k]2^{\sum_{j=1}^{n} U_j^i[k]} \\ &= \sum_{k=1}^{160} R_i[k]Q[k]2^{(\tilde{f}_{q,1}[k]+\tilde{f}_{i,1}[k])} \\ &= \sum_{k=1}^{160} R_i[k]Q[k]2^{\log\left|\frac{f_{q,k}f_{i,k}}{\|f_q\|\|f_i\|}\right|} \\ &= \sum_{k=1}^{160} \frac{f_{q,k}f_{i,k}}{\|f_q\|\|f_i\|} \\ &= \frac{f_q^T f_i}{\|f_q\|\|f_i\|} \end{aligned} \quad (5)$$

Similarly, the values of cosine similarity between f_q and all feature vector can be got. Then these values will be compared with the threshold t of our face verification to find whether this people is registered in our system. Finally, the permission can be got and returned to the edge computing server which require for identity authentication.

V. SECURITY ANALYSIS

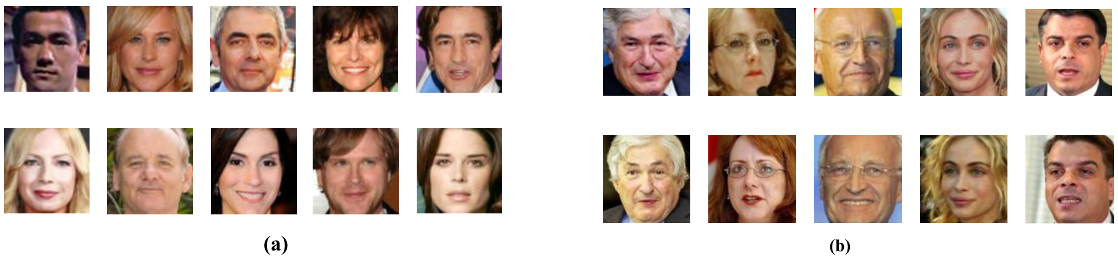
The security analysis of the proposed method is discussed in this section. The cloud server in our system is considered to be ''honest-but-curious'', which means that the cloud server will carry out the protocol what you specify, however, it will analyse and keep watching all over the sensitive information about face data. Therefore, the face data that uploaded to the cloud server need to be properly protected. The n edge computing servers we used in the system are defined as non-colluding and independent with each other. Thus, we must guarantee that the edge computing server can not obtain any information about the face data only by the secret shadow they receive from other edge computing servers.

A. CLOUD SERVER

As described in Section 3, for all the secret shadows generated in the stage of identity registration and authentication, they will to be encrypted by AES encryption algorithm before being uploaded to the cloud server. Thus, for the cloud server, the security of the secret shadows is dependent on the the security of AES encryption algorithm. In the stage two of identity authentication, the cloud server collects all the calculation results $U_j^i (1 \leq j \leq t)$ of t chosen edge computing servers but does not know the values of $x_j (1 \leq j \leq t)$. What's more, $U_j^i (1 \leq j \leq t)$ are computed by the secret shadows of $\tilde{f}_i$ and $\tilde{f}_q$ and the $\tilde{f}_q$ of every request are different from each other, although they are coming from the same person. Therefore, the cloud server has not enough information to compute to get the content of face date of the users and the security of face feature vectors in cloud server can be guaranteed.

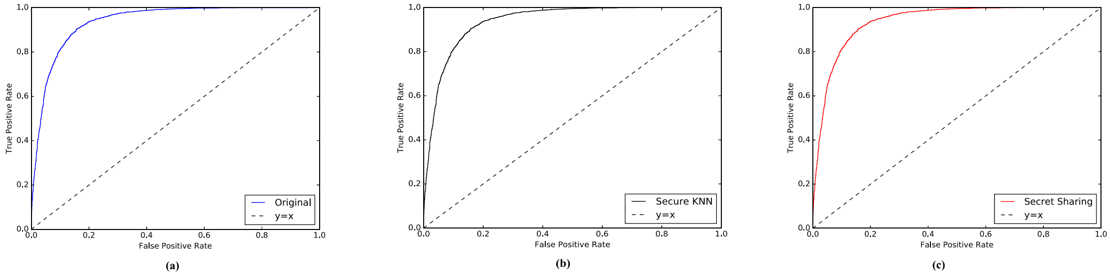
B. EDGE COMPUTING SERVER

In our system, the edge computing server is credible for the face data of the users who register on it. The secure nearest neighbor scheme is employed to protect the face feature vectors that are stored in database-1 from being stolen by the adversary. Although the edge computing server invaded by the adversary, it is proved to be secure against the Chipertext-only Attack (COA) model for our data protected by secure nearest neighbor algorithm in [22]. Moreover, the databae-2 store the secret shadows of the users who register on the other edge computing servers and the security of these data is based on the security of Shamir's scheme [18]. Each edge computing server only has a secret shadow of the face feature vectors registered on other edge computing servers, so it can not get any information about these vectors. Supposing that $t - 1$ or fewer edge computing servers are being invaded by the adversary at the same time, he has no way of recovering the polynomial of degree $t - 1$ described in Section 3, nor can


(a)

(b)

FIGURE 8. Example face images of the training and testing datasets. (a) CASIA-WebFace. (b) LFW.



(a)

(b)

(c)

FIGURE 9. ROC comparison about three face verification schemes. (a) Original. (b) Secure nearest neighbor. (c) Secret sharing.

he acquire any things about the face feature vectors of users in this system.

VI. PERFORMANCE EVALUATION

In previous sections, we introduce the algorithms and basic framework of our face privacy protection and authentication system. In this section, some of the different experimental tests have been carried out to present the performance evaluation of our system. The entire secure schemes are implemented using python language on a Windows operation system with Intel Core(TM) 3.3GHz CPU and 8G RAM.

A. EFFECTIVENESS EVALUATION

In our experiments, caffe, a popular deep learning framework, is used to train our neural network. We trained our face recognition network model on CASIA-WebFace database and tested on LFW database. CASIA-WebFace database contains 10575 subjects and 494414 images while LFW database contains 5749 subjects and 13233 images, but only 1680 of the person have two or more distinct images, so LFW database is usually used to test accuracy of face recognition. What's more, images in databases of CASIA-WebFace and LFW are mutually exclusive. Parts of the face images from training and testing datasets are shown in Figure 8, Figure.8(a) is 8 different examples of CASIA-WebFace database and Figure.8(b) is 4 pairs faces of LFW database and the images of each column are from the same people. After training the face recognition network, 6000 face pairs from LFW database are chosen for testing our network model. The results are shown in Figure 9.

Figure 9 compares the ROC curves drawn by the face verification result computed with the cosine similarity of original feature vectors and the feature vectors protected by two schemes we proposed in section 4. It is shown that these three curves are exactly the same. Therefore, we can conclude that the result of face verification with the ciphertext of feature vectors that protected by our proposed method is the same as the result computed by the plaintext of feature vector. It can be said that the accuracy of face verification is not influenced by the encryption algorithm we proposed. Moreover, the threshold of cosine similarity in our face verification is 0.52 that is chosen by plenty of experiments. Based on this threshold, the face recognition network we train achieves 92.4591% test accuracy on 6000 face pairs from LFW database. Though the accuracy of our network model is not high enough compared to the state of art work, what is focused in this paper is whether our secure algorithm will affect the accuracy of face verification.

B. EFFICIENCY EVALUATION

In this subsection, various experiments have been done to evaluate the efficiencies of time consumption and storage consumption about the methods we proposed in this paper.

1) TIME CONSUMPTION

We compare the time consumption of face recognition achieved by the plaintext of face data with the ciphertext that encrypted by two privacy-preserving methods we proposed above. The time consumption on the method of secure nearest neighbor contains the time of feature vectors extraction, encryption and recognition. While for the method of secret sharing, in addition to the time consumption mentioned above, it also contain the time that spent on the interaction between the edge computing servers and cloud server. The results about time consumption of these three situations are recorded in Figure 10 according to considerable experiments.

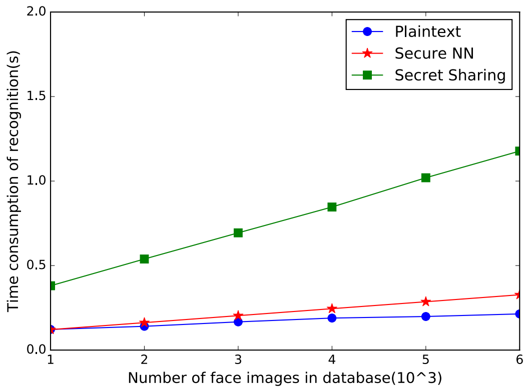


FIGURE 10. Time consumption of two methods.

TABLE 1. Storage consumption of 10000 face data.

Face data	Storage consumption(KB)
Face Image	85504.53
Face Feature Vector	7031.21

As illustrated in Figure 10, we can find that the time consumption of these three methods are almost linear to the number of face images in database. Notice that, the time consumption of face recognition with the face feature vectors protected by the secure nearest neighbour is almost equal to that with the plaintext of face feature vectors. Thus, performing secure nearest neighbour scheme on the edge computing server can protect the face data of the users registered on it from being stolen by the adversary and won't waste too much time at the same time. From Figure 10 we can also see that the time consumption of face recognition with the face feature vectors that protected by secret sharing scheme is under an acceptable range. Although the time cost of secret sharing scheme is a little longer, it can achieve cross-authentication among all the edge computing servers on the premise of protecting the privacy of all face data. What's more, it is most frequently for the users who request for identity authentication on the edge computing server that they registered on in the most of scenario. Thus, the identity authentication can be complete only on the edge computing server without the additional communication burdens in this scenario so that a lot of time could be saved.

2) STORAGE CONSUMPTION

According to the previous introductions, our system is based on edge computing that allows us to preprocess the face data on the edge of the Internet to avoid uploading the data to the Internet directly. In table 1, we present the comparison of storage consumption about face images and corresponding face feature vectors, the results are calculated by 10000 face images. We can conclude from table 1 that the storage consumption of the face feature vectors that extracted by edge computing server is nearly ten times smaller than original face images. Moreover, it can also reduce the communication burdens during the protocol execution.

VII. CONCLUSION

A privacy-preserving edge computing-based face verification system for user authentication is proposed in this paper. Face feature vectors are extracted by the method of convolutional neural network. Edge computing is introduced in our system to perform privacy-preserving face verification. All operations on edge computing server and cloud server are preformed on the encrypted feature vectors to prevent privacy leaks. A secure nearest neighbor scheme is proposed for the edge computing server to compute the cosine similarity over encrypted feature vectors. What's more, a secret sharing homomorphism technology is used in our system for a part of all edge computing servers to compute cosine similarity respectively and complete authentication along with the cloud service provider. The results of our experiments show that technology of secret sharing homomorphism can enhance the fault-tolerance of our identity authentication system and the introduction of edge computing can improve the authentication efficiency enormously. How to ameliorate the encryption algorithm to further reduce the time consumption of authentication is still an open problem and will continue to be studied in our future work.

REFERENCES

[1] H. Gunasinghe and E. Bertino, "PrivBioMTAuth: Privacy preserving biometrics-based and user centric protocol for user authentication from mobile phones," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 1042–1057, Apr. 2018.

[2] O. Ogbanufe and D. J. Kim, "Comparing fingerprint-based biometrics authentication versus traditional authentication methods for e-payment," *Decis. Support Syst.*, vol. 106, pp. 1–14, Feb. 2018.

[3] C.-T. Hsieh, C.-C. Han, C.-H. Lee, and K.-C. Fan, "Person authentication using nearest feature line embedding transformation and biased discriminant analysis," presented at the Int. Carnahan Conf. Secur. Technol. (ICCST), Madrid, Spain, Oct. 2017.

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

[5] N. Powers *et al.*, "The cloudlet accelerator: Bringing mobile-cloud face recognition into real-time," presented at the IEEE Globecom Workshops (GC Wkshps), San Diego, CA, USA, Dec. 2015.

[6] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[7] M. Blanton and P. Gasti, "Secure and efficient protocols for iris and fingerprint identification," presented at the Eur. Symp. Res. Comput. Secur., Leuven, Belgium, Sep. 2011.

[8] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," presented at the Int. Conf. Inf. Secur. Cryptol., Seoul, South Korea, Dec. 2009.

[9] J. Bringer, H. Chabanne, and A. Patey, "Practical identification with encrypted biometric data using oblivious ram," presented at the Int. Conf. Biometrics (ICB), Madrid, Spain, Jun. 2013.

[10] J. Bringer, H. Chabanne, and A. Patey, "Shade: Secure Hamming distance computation from oblivious transfer," presented at the Int. Conf. Financial Cryptogr. Data Secur., Okinawa, Japan, Apr. 2013.

[11] J. Bringer, H. Chabanne, M. Favre, A. Patey, T. Schneider, and M. Zohner, "GSHADE: Faster privacy-preserving distance computation and biometric identification," presented at the 2nd ACM Workshop Inf. Hiding Multimedia Secur., Salzburg, Austria, Jun. 2014.

[12] Y. Cai and C. Tang, "Securely outsourced face recognition under federated cloud environment," presented at the 15th Int. Symp. Parallel Distrib. Comput. (ISPDC), Fujian, China, Jul. 2016.

[13] Y. Ma, L. Wu, X. Gu, J. He, and Z. Yang, "A secure face-verification scheme based on homomorphic encryption and deep neural networks," *IEEE Access*, vol. 5, pp. 16532–16538, 2017.

[14] M. Haghighat, S. Zonouz, and M. Abdel-Mottaleb, "CloudID: Trustworthy cloud-based and cross-enterprise biometric identification," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7905–7916, Nov. 2015.

[15] Z. Xia, N. N. Xiong, A. V. Vasilakos, and X. Sun, "EPCBIR: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing," *Inf. Sci.*, vol. 387, pp. 195–204, May 2017.

[16] Y. Xu, J. Gong, L. Xiong, Z. Xu, J. Wang, and Y.-Q. Shi, "A privacy-preserving content-based image retrieval method in cloud environment," *J. Vis. Commun. Image Represent.*, vol. 43, pp. 164–172, Feb. 2017.

[17] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," presented at the IEEE Conf. Comput. Vis. Pattern Recognit., Columbus, OH, USA, Jun. 2014.

[18] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[19] G. R. Blakley, "Safeguarding cryptographic keys," presented at the Nat. Comput. Conf., New York, NY, USA, Jun. 1979.

[20] J. C. Benaloh, "Secret sharing homomorphisms: Keeping shares of a secret secret," presented at the Conf. Theory Appl. Cryptograph. Techn., 1986.

[21] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," presented at the Proc. 12th Annu. ACM Symp. Theory Comput., Chicago, IL, USA, May 1988.

[22] W. K. Wong, D. W.-L. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on encrypted databases," presented at the ACM SIGMOD Int. Conf. Manage. Data, New York, NY, USA, Jun./Jul. 2009.

XIANG WANG received the B.S. and M.E. degrees from Shandong University, in 2004 and 2007, respectively, and the Ph.D. degrees from Peking University, in 2011. He is currently an Associate Professor with the School of Telecommunications Engineering, Xidian University. He is a member of the ACM and China Computer Federation. His research interests focus on digital contents protection and visual cryptography.

HEYU XUE received the B.S. degrees in information security from Xidian University, Xi'an, China, in 2016, where he is currently pursuing the M.S. degree with the Department of Communication. His current research interests include image processing and privacy protection.

XUEFENG LIU received the B.S. and Ph.D. degrees in information security from Xidian University, China, in 2007 and 2013, respectively, where he is currently an Associate Professor. His research interests include the fields of cloud computing security and applied cryptography.

QINGQI PEI received the B.S., M.S., and Ph.D. degrees in computer science and cryptography from Xidian University, in 1998, 2005, and 2008, respectively. He is currently a Professor, and also a member of the State Key Laboratory of Integrated Services Networks. He is a professional member of the ACM, member of the IEEE, and a Senior Member of the Chinese Institute of Electronics and China Computer Federation. His research interests focus on digital contents protection and wireless networks and security.