manuscript No.
(will be inserted by the editor)

Privacy-Preserving Face Recognition with Outsourced Computation

Can Xiang · Chunming Tang

Received: date / Accepted: date

Abstract Face recognition is one of the most important biometrics pattern recognitions, which has been widely applied in a variety of enterprise, civilian and law enforcement. The privacy of biometrics data raises important concerns, in particular if computations over biometric data is performed at untrusted servers. In previous work of privacy-preserving face recognition, in order to protect individuals' privacy, face recognition is performed over encrypted face images. However, these results increase the computation cost of the client and the face database owners, which may enable face recognition cannot be efficiently executed. Consequently, it would be desirable to reduce computation over sensitive biometric data in such environments. Currently, no secure techniques for outsourcing face biometric recognition is readily available. In this paper, we propose a privacy-preserving face recognition protocol with outsourced computation for the first time, which efficiently protects individuals' privacy. Our protocol substantially improves the previous works in terms of the online computation cost by outsourcing large computation task to a cloud server who has large computing power. In particular, the overall online computation cost of the client and the database owner in our protocol is at most 1/2 of the corresponding protocol in the state of the art algorithms. In addition, the client requires the decryption operations with only $O(1)$ independent of M, where M is the size of the face database. Furthermore, the client can verify the correction of the recognition result.

Keywords Face recognition · Outsourced computation · Privacy-preserving

C. Xiang
College of Mathematics and Information Science,
Guangzhou University, Guangzhou 510006, China
E-mail: xiangcan1987@sina.com

C. Tang
College of Mathematics and Information Science,
Guangzhou University, Guangzhou 510006, China
Key Laboratory of Mathematics and Interdisciplinary Sciences of Guangdong Higher Education Institutes,
Guangzhou University, Guangzhou 510006, China
State Key Laboratory of Information Security,
Beijing 100093, China
E-mail: ctang@gzhu.edu.cn

1 Introduction

Biometric techniques have advanced over the past years to a reliable means of authentication, which have been deployed in various application domains. Many governments have already rolled out electronic passports [1] and IDs [2] that contain biometric information (e.g.,face image, fingerprints, and iris scan) of their legitimate holders. Unlike other types of data used for authentication purposes (passwords, key material, secure tokens, etc.), biometric data cannot be revoked and replaced with a new value, hence it calls for strict protection of such biometric data. In particular, face recognition systems have become more popular due to its unobtrusiveness and ease of use. Thus, face recognition systems have been widely applied in a variety of enterprise, civilian and law enforcement, such as surveillance of public places, access and border control at airports, facebook in social networking platforms, etc.

The widespread use of face recognition systems, however, it will bring privacy risks because biometric information can be collected and misused to profile and track against their will. These issues raise the desire to construct privacy-preserving face recognition systems. In recent years, many methods for protecting biometric data were proposed, such as methods on fuzzy vault

[3-5], secure sketches and fuzzy extractors [6-11], shielding functions [12-14], cancelable or revocable biometrics [15], and so on. These methods stored a function of each biometric rather than the biometrics data themselves, but it did not lead to compromise of the biometric data in the case of server compromise. For face recognition systems, in order to protect individuals' privacy, face recognition was performed over encrypted face images in previous works [16-17]. However, these results increased the computation cost of clients and database owners of the face images, which enable face recognition cannot be efficiently executed. Currently, no existing tools or techniques are readily available to carry out the huge computation task of the database owner. Thus, the problem of secure biometric face identification (or matching) with the aid of untrusted servers is the focus of this work.

In this paper, we concentrate on efficient privacy-preserving face recognition systems. The typical scenario here is a application which consists of three parties, i.e., a client, a database owner of face images and a cloud server. Both the client and database owner have limited (or weak) computing power, but the cloud server has the ability to process magnanimity data and perform parallel computation. The client provides a specific face image and needs to know the image whether is contained in the database of face images with the following requirements: 1)the client believes the database owner correctly performs the matching algorithm for the face recognition but without revealing any useful information to the database owner about the requested image as well as about the outcome of the matching algorithm; 2) the database owner requires privacy of its database beyond the matching results to the client; 3) the database owner needs help from the cloud server which cannot reveal any useful information about real face images and can greatly reduce the database owner's computation cost by using the ability of processing intelligent data and performing parallel computation.

1.1 Related Work

Some authors have proposed different complementary techniques for making surveillance cameras more privacy friendly, e.g. [18-20]. However, they do not consider face recognition. For privacy-preserving face recognition, Erkin et al.[16] proposed for the first time a strongly privacy-enhanced face recognition system. They used the standard and popular Eigenface [21-22] recognition algorithm. The system performs operations on encrypted images by means of homomorphic encryption schemes, more concretely, Pailler [23-24] as well as a cryptographic protocol for comparing two Pailler encrypted values based on DGK (Damgard, Geisler and Kroigard) cryptosystem [25-27]. They demonstrate that privacy-preserving face recognition is possible in principle and give required choices of parameter sizes to achieve a good classification rate. However, the proposed protocol requires $O(logM)$ rounds of online communication as well as computationally expensive operations on homomorphically encrypted data to recognize a face in the database of M faces. Due to these restrictions, the proposed protocol cannot be deployed in practical large-scale applications. After that, Sadeghi et al. [17] given two privacy-preserving face recognition protocols which substantially improved over previous work [16]. One is based on homomorphic encryption (see, e.g., [23-24]) and Yao et al.'s Garbled Circuit (GC)[28-29], the other is based on GC only. Althouth the protocols allowed to shift most of the computation and communication into a pre-computation phase, the computation cost of the client and database owner was not reduced. This means that efficiently implementing privacy-preserving face recognition is difficult for the client and the database owner with weak computing power. We improves Sadeghi et al. 's protocol based on homomorphic encryption and Garbled Circuit in this paper. In the rest of the paper, the protocol in [17] is based on homomorphic encryption and Garbled Circuit unless stated otherwise.

The related problem of privacy-preserving face detection [30] allows a client to detect faces on his image using a private classifier held by servers without revealing the face or the classifier to the other party. In order to preserve privacy, faces can be de-identified so that face recognition software cannot reliably recognize de-identified faces, even though many facial details are preserved as described in [31].

Beside privacy - preserving face recognition, there were a few attempts to make other biometric modalities privacy preserving, such as fingerprints and iris codes [32-34]. However, these works consider a different setting, where the biometric measurement is matched against a hashed template stored on a server. The server that performs the matching gets to know both the biometric and the detection result (the aim is only to secure storage of templates). Blanton and Aliasgari [35] proposed a secure outsourced computation scheme of iris matching. To the best of our knowledge, there is no prior solution to carry out the huge computation task of the database owner in the secure privacy-preserving face recognition system. In order to reduce the computation cost, we present a new protocol for privacy-preserving face recognition which can outsource lager

computation task to a third party (e.g., cloud servers) who has a huge computing power.

1.2 Contribution

We propose an efficient and secure privacy-preserving face recognition protocol with outsourced computation. Our protocol is based on the Eigenfaces recognition algorithm [21-22] and a hybrid Encryption based on FHE [36]. We do not use Garbled Circuits. Our protocol substantially improves over previous work [16-17] as it has only one round between the client and the database owner. Furthermore, the protocol can efficiently outsource most of the computation to an untrusted cloud server. The remaining computation cost of the client and the database owner is small. Beyond the encryption operations, the online computation cost of the client and the database owner in our protocol is at most 1/2 of the corresponding protocol in the state of the art algorithms, this is especially important for the client and the database onwer with weak computing power.

1.3 Organization

The rest of the paper is organized as follows. We summarize our model and security requirements, parameters setting and cryptographic tools used in our constructions in Section 2. A summary of the face recognition algorithm using Eigenfaces is reviewed in Section 3. Section 4 details our secure privacy-preserving face recognition protocol with outsourced computation. Security and efficiency analysis of our protocol are given in section 5. And section 6 concludes the paper.

2 Preliminaries

2.1 Model and security requirements

In this paper, three parties are involved in our schemes, that is, a client, a database owner of face images and an untrusted cloud server. Both the client and database owner have limited (or weak) computing power, but the cloud server has the ability to process magnanimity data and perform parallel computation. The client provides a specific face image and needs to know the image whether is contained in the database of face images with the following requirements: 1)the client trusts the database owner to correctly perform the matching algorithm for the face recognition but without revealing any useful information to the database owner about the requested image as well as about the outcome of the matching algorithm; 2) the database owner requires privacy of its database beyond the outcome of the matching algorithm to the client; 3) the database owner needs help from the cloud server which cannot learn any useful information about real face images and can greatly reduce the database owner's computation cost by using the ability of processing intelligent data and performing parallel computation.

We work in the semi-honest model where the client and the database owner are assumed to be honest-but-curious but the cloud server is untrusted.

Similar to [16,17], we summarize the notations and the parameters used in this paper in Table 1.

Table 1 Summarize of notations and parameters

Parameter	Description
M	number of faces in database
N	size of a face in pixels
K	number of Eigenfaces
Γ	face
Ψ	average face
$u_1, u_2, \cdots, u_K$	Eigenfaces
$\bar{\Omega}$	projected face for Γ
$\Omega_1, \cdots, \Omega_M$	projected faces in database
$D_1, \cdots, D_M$	squared distances between projected images
l'	the bit length of values $D_1, \cdots, D_M$
τ	threshold value

2.2 Cryptographic tools

Hybrid encryption based on FHE. We use a semantically secure hybrid encryption (HS) based on FHE scheme in [36], which is a combination of an ordinary (non-FHE) encryption scheme and a FHE scheme. In [36], the authors given a detailed hybrid encryption scheme using a symmetric encryption scheme and a FHE scheme. A public-key encryption schemes (e.g., RSA, Paillier) can as well be used as an ordinary encryption scheme. In this paper, we use a semantically secure hybrid encryption (HS) which is a combination of a public key encryption scheme and a FHE scheme. Let M be a plaintext space, (FHE.KeyGen,FHE.Enc, FHE.Dec,FHE.Eval) be a FHE scheme, and (PE.KeyGen, PE.Enc, PE.Dec) be a public-key encryption scheme (PE). A hybrid encryption scheme HS=(HS.KeyGen, HS.Enc_1,HS.Enc_2, HS.Dec, HS.Eval) consists of five PPT algorithms (PPT is shorthand for probabilistic polynomial time), which are described as follows.

- HS.KeyGen(λ) $\rightarrow$ $(pk, dk, pk', sk, \kappa)$. Takes as input a security parameter λ, runs FHE.KeyGen to obtain a public encryption key pk and a secret decryption key dk, runs PE.KeyGen to obtain public

encryption key pk' and a secret decryption key sk. Then, encrypts sk under the public key pk to obtain $\kappa \leftarrow \text{FHE.Enc}(pk, sk)$, outputs $(pk, dk, pk', sk, \kappa)$.
- $\text{HS.Enc}_1(pk', x) \rightarrow c_x$. Runs PE.Enc to encrypt message $x \in M$ under the public key pk', outputs ciphertext c.
- $\text{HS.Enc}_2(pk, \kappa, c) \rightarrow c_x$. On input (pk, κ, c), outputs a new ciphertext c_x which is equal to $\text{FHE.Enc}(pk, x)$.
- $\text{HS.Dec}(dk, c_x) \rightarrow x$. Same as FHE.Dec. Takes as input dk and c_x, and decrypts the ciphertext c_x to a plaintext $x \in M$ under the secret key dk.
- $\text{HS.Eval}(pk, C, c_1, c_2, \cdots, c_n) \rightarrow c_y$. Same as FHE.Eval. Given the public key pk, a circuit C and a set of n ciphertexts $c_1, c_2, \cdots, c_n$ deterministically compute and outputs a new ciphertext c_y.

Similar to FHE, A HS scheme should also satisfy four properties, which is encryption correctness, evaluation correctness, succinctness and semantic security.

As instantiation we use the Paillier public-key encryption scheme [23-24] which has plaintext space Z_N and ciphertext space Z_{N^2}, where N is a T-bit RSA modulus, while we use the FHE scheme over the integers [37]. In [16], the privacy-preserving face recognition protocol uses the homomorphic cryptosystem of Damgard, Geisler and Kroigard (DGK) other than the Paillier public-key encryption scheme. The DGK homomorphic encryption scheme can reduce the ciphertext space to Z_N^*. In [17], the privacy-preserving face recognition protocol additionally uses Yao's Garbled Circuit other than the Paillier public-key encryption scheme. Both protocols in [16,17] do not use FHE.

In this paper, in an additively homomorphic Paillier encryption scheme, given encryptions $[a]_{PE}$ and $[b]_{PE}$, an encryption $[a+b]_{PE}$ can be computed by $[a+b]_{PE} = [a]_{PE} \cdot [b]_{PE}$, where all operations are performed in the algebra of the message or ciphertext space. Furthermore, in a FHE scheme, given encryptions $[a]_{FHE}$ and $[b]_{FHE}$, an encryption $[a+b]_{FHE}$ can be computed by $[a+b]_{FHE} = [a]_{FHE} + [b]_{FHE}$ and $[ab]_{FHE}$ can be computed by $[ab]_{FHE} = [a]_{FHE}[b]_{FHE}$.

3 Face recognition algorithm using Eigenfaces

In the following we briefly summarize the recognition process of the Eigenfaces algorithmm[16-17,21-22]. The algorithm obtains as input the query face image Γ represented as a pixel image with N pixels. Additionally, the algorithm obtains the parameters determined in the enrollment phase as inputs: the average face Ψ which is the mean of all training images, the Eigenfaces $u_1, \cdots, u_K$ which span the K-dimensional face space, the projected faces $\Omega_1, \cdots, \Omega_M$ being the projections of the M faces in the database into the face space, and the threshold value τ. The output r of the recognition algorithm is the index of that face in the database which is closest to the query face Γ or the special symbol $\perp$ if no match was found, i.e., all faces have a larger distance than the threshold τ. Specifically, the recognition algorithm consists of three phases, which are described as follows.

1. **Projection:** The average face Ψ is subtracted from the face Γ and the result is projected into the K-dimensional face space using the Eigenfaces $u_1, \cdots, u_K$. The result is the projected K-dimensional face $\bar{\Omega}$.
2. **Distance:** The square of the Euclidean distance D_i between the projected K-dimensional face $\bar{\Omega}$ and all projected K-dimensional faces in the database $\Omega_i (i = 1, 2, \cdots, M)$, is computed.
3. **Minimum:** The minimum distance D_{min} is selected. If D_{min} is smaller than the threshold τ, the index of the minimum value, i.e., the identifier i_{min} of the match found, is returned to the client as result $r = i_{min}$. Otherwise, the image was not found and the special symbol $r = \perp$ is returned.

4 Privacy-preserving face recognition with outsourced computation

In this section, we present a privacy - preserving face recognition protocol with outsourced computation. The protocol operates on encrypted images. three parties are involved in our schemes, that is, a client C, a database owner DB of face images and an untrusted cloud server CS. We work in the semi-honest attacker model. Informally, this assumes that the client C and the database owner DB follow the protocol but try to learn additional information from them. In addition, any outsourcer can verify the correctness of the untrusted cloud server output. It is also assumed that the parties communicate over an authenticated channel (this can be achieved by standard mechanisms and is thus outside the scope of this paper). We assume that a database owner has already set up the face recognition system by running the enrollment process (in the clear) on all available training images $\{\theta_1, \cdots, \theta_M\}$ to obtain the basis $u_1, \cdots, u_K$ of the face space and feature vectors $\Omega_i (i = 1, 2, \cdots, M)$ of faces to be recognized.

Furthermore, we assume that all coordinates of the eigenfaces and feature vectors are represented as integers. Each feature vector in the database is further accompanied by a string id_i that contains the identity of the person the feature vector belongs to; we assume that the identity is encoded as a non-zero element of the message space of the chosen encryption scheme.

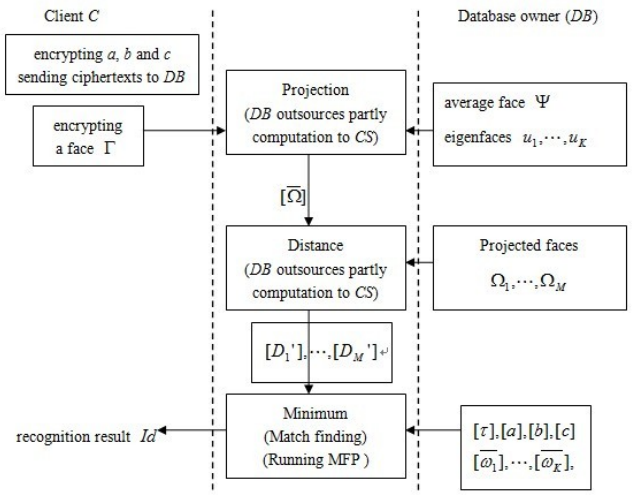

Fig. 1 Outline of our protocol

Fig. 1 shows an outline of our protocol, which is described as follows.

Projection. The input image Γ has to be projected onto the low dimensional face space spanned by the eigenfaces $u_1, \cdots, u_K$. the client C runs HS.Gen to obtain $(pk, dk, pk', sk, \kappa)$, where (pk, dk) is the public/secret key pair of the FHE and (pk', sk) is the public/secret key pair of the Paillier homomorphic encryption scheme. In addition, κ is the encryption of sk under FHE. Let κ be $[sk]_{FHE}$. The client C encrypts the face Γ as $[\Gamma]_{PE} = ([\Gamma_1]_{PE}, \cdots, [\Gamma_N]_{PE})$. Meanwhile, the client randomly chooses three elements a,b and c, and encrypts them as $[a]_{PE}$, $[b]_{PE}$ and $[c]_{PE}$.

The client sends $\{[\Gamma]_{PE}, [a]_E, [b]_{PE}, [c]_{PE}, [sk]_{FHE}\}$ along with (pk, pk') to the database owner DB. Using the homomorphic properties and outsourced computation, DB projects the encrypted face into the low-dimensional face space and obtains the encryption of the projected face $[\bar{\Omega}]_{PE} = ([\bar{\omega_1}]_{PE}, \cdots, [\bar{\omega_K}]_{PE})$ as follows.

1. For $i = 1, \cdots, K$, the database owner DB computes $p_i = -\sum_{j=1}^{N} u_{i,j}\Psi_j$, where Ψ_j is the component of the vector $\Psi = \frac{1}{M}\sum_{i=1}^{M}\theta_i$. This step can be completed in the pre-computation phase (offline phase).
2. The database owner DB encrypts p_i under the public key pk' to obtain $[p_i]_{PE} = [-\sum_{j=1}^{N} u_{i,j}\Psi_j]_{PE}$, This step is completed in the online phase.
3. DB computes $q_i = \prod_{j=1}^{N}[\Gamma_j]_{PE}^{u_{i,j}}$ by using outsourcing exponentiation algorithm (such as [38]) which can reduce the database owner DB 's computation cost.
4. For $i = 1, \cdots, K$, the database owner computes $[\bar{\omega_i}]_{PE} = [p_i]_{PE} \cdot q_i$.

Distance. After **Projection**, the database owner DB need to compute the Paillier encryption of the Euclidean distances between the projected face $\bar{\Omega}$ and all projected faces $\Omega_1, \cdots, \Omega_M$ in the database held by the database owner in [16,17]. In addition, DB also needs interaction with the client. In our protocol, DB does not need interaction with the client. Because the computation cost is very large for DB, who may no capacity to complete the computation task by himself. Thus, the database owner DB requires assistance from a third party (e.g., cloud server). In this paper, we use outsourced computation which can enable the database owner to outsource all or partly computation to the cloud server who has lager computing powering. For $i = 1, 2, \cdots, M$, the encryption of the square Euclidean distances $[D_i]_{PE} = [\|\Omega_i - \bar{\Omega}\|^2]_{PE} = [S_{1,i} + S_{2,i} + S_3]_{PE} = [S_{1,i}]_{PE} \cdot [S_{2,i}]_{PE} \cdot [S_3]_{PE}$, where $[S_{1,i}]_{PE} = [\sum_{j=1}^{K}\omega_{i,j}^2]_{PE}$, $[S_{2,i}]_{PE} = [\sum_{j=1}^{K}(-2\omega_{i,j}\bar{\omega_j})]_{PE} = \prod_{j=1}^{K}[\bar{\omega_j}]_{PE}^{-2\omega_{i,j}}$ and $[S_3]_{PE} = [\sum_{j=1}^{K}\bar{\omega_j}^2]_{PE}$. We notice that S_3 is a fixed value once the input mage Γ and the face database are fixed. Hence, DB only need to compute $[D_i']_{PE} = [S_{1,i} + S_{2,i}]_{PE} = [S_{1,i}]_{PE} \cdot [S_{2,i}]_{PE}$. These cannot effect the next step (See. Match finding). Specifically, $[D_i']_{PE}$ can be computed as follows.

1. To obtain $[S_{1,i}]_{PE}$, the database owner DB needs to complete two steps below.
   - DB computes $\sum_{j=1}^{K}\omega_{i,j}^2$ which can be pre - computed in the offline stage.
   - DB encrypts $\sum_{j=1}^{K}\omega_{i,j}^2$ under the public key pk' to obtain $[S_{1,i}]_{PE} = [\sum_{j=1}^{K}\omega_{i,j}^2]_{PE}$. This step is completed by the database owner DB in the online phase.
2. For computing $[S_{2,i}]_{PE}$, the database owner DB firstly outsources exponentiation $[\bar{\omega_j}]_{PE}^{-2\omega_{i,j}}$ to the cloud server CS by outsourcing exponentiation algorithm, and then multiply those exponentiation together to obtain $[S_{2,i}]_{PE}$.
3. For $i = 1, 2, \cdots, M$, DB computes $[D_i']_{PE} = [S_{1,i}]_{PE} \cdot [S_{2,i}]_{PE}$

Then, DB can finish this phase without interacting with the client.

Minimum (Match finding). In the last step of the recognition algorithm, the goal is to find the minimum value D from $\{D_i\}_{i=1}^{M}$ and its index Id_{min}. If the minimum value D is smaller than the threshold value τ known by the database owner, then a match is reported and an encryption of the identity Id_{min} which corresponds to the best matching feature vector is returned to the client. Because S_3 is a fixed value, we only need to find the minimum value D' from $\{D_i'\}_{i=1}^{M}$ and its index Id_{min}. If the minimum value D' is smaller than the value $\tau' = \tau - S_3$, then a match is reported and

an encryption of the identity Id_{min} which corresponds to the best matching feature vector is returned to the client.

As we need to return the identity of the best matching feature vector, we also have to keep track of the IDs during the minimum computation. This is done by working with pairs $([D_i']_{PE}, [Id_i]_{PE})$ of distances and their corresponding identities. To check if the minimum distance is smaller than τ', we can treat the value τ' as one additional distance that has the special identity 0. Together with the distances $D_1' \cdots, D_M'$, the client, the database owner, and the cloud server jointly carry out the protocol with verifiable outsourced computation to find minimum distance and the corresponding identity $([D']_{FHE}, [Id]_{FHE})$, where $D' \in \{\tau', D_1', \cdots, D_M'\}$ and $Id \in \{0, Id_1, \cdots, Id_M\}$. Thus, if a face image could be recognized the value Id contains the corresponding identity. If no match could be found Id is equal to 0. Some encrypted values are finally sent to the client as the result of the private face recognition protocol. Then, the client can obtain the recognition result Id by some computations and can verify the correctness of the result. To achieve this, the client C, the database owner DB, and the cloud server CS jointly run the following match finding protocol (MFP) with verifiable outsourced computation(VOC).

1. The database owner DB constructs a circuit $C_{circuit}$ with multi-input, which is shown in Fig.2. This can be completed by the database owner in the offline stage.
2. DB sends the circuit $C_{circuit}$, $[sk]_{FHE}$ and $\sigma = \{[\tau]_{PE}, [\bar{\omega_1}]_{PE}, \cdots, [\bar{\omega_K}]_{PE}, [D_1']_{PE}, \cdots, [D_M']_{PE}, [a]_{PE}, [b]_{PE}, [c]_{PE}\}$ to the cloud server CS.
3. For each element in the set σ, CS runs the algorithm $HS.Enc_2$ to get $\sigma' = \{[\tau]_{FHE}, [\bar{\omega_1}]_{FHE}, \cdots, [\bar{\omega_K}]_{FHE}, [D_1']_{FHE}, \cdots, [D_M']_{FHE}, [a]_{FHE}, [b]_{FHE}, [c]_{FHE}\}$.
4. CS computes $FHE.Eval(pk, C_{circuit}, \sigma')$ to obtain $[\Delta_1]_{FHE}$ and $[\Delta_2]_{FHE}$, then sends them to the client.
5. The client decrypts $[\Delta_1]_{FHE}$ and $[\Delta_2]_{FHE}$ under the secret key dk to obtain Δ_1 and Δ_2. If $\Delta_1 - a = c(\Delta_2 - b)$, then the client accepts the match result $Id = \Delta_2 - b$, otherwise rejects. If $Id = 0$, it shows that no match could be found in the database held by the database owner.

In our minimum (Match) finding protocol, the online computation and round complexity have been substantially improved for the client C and the database owner DO, we have given the comparison of three minimum protocols from two aspects, i.e., the online round complexity and the asymptotic computation complexity (ACC1), which is shown as in Table 2 with parameter $m \approx \frac{l'M}{T-\kappa'}$, where T is the asymmettric security parameter and κ' is the statistical correctness parameter in [17].

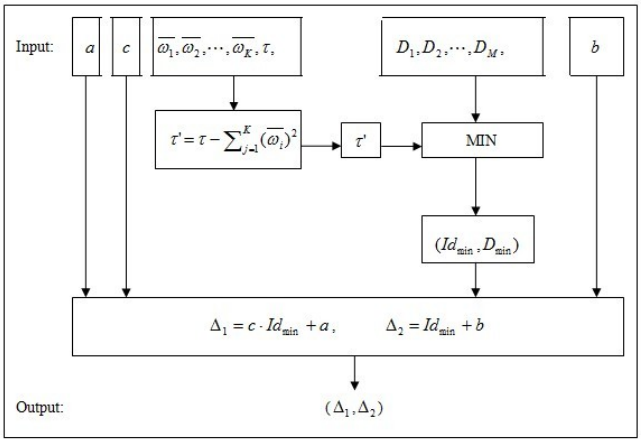

Fig. 2 Circuit $C_{circuit}$ for match finding with VOC

5 Security and efficiency analysis

The security of our protocol is based on the security of there schemes, i.e. the privacy-preserving face recognition scheme in [17], HS scheme in [36] and the verifiable outsourced computation scheme for simultaneous exponentiations in [38]. This three schemes are secure which has been proved in [17,36,38]. Thus, our protocol is secure.

In order to illustrative the efficiency of our protocol, we will give detailed analysis from three respects, i.e., the round, the communication and computation complexity.

5.1 Round complexity

The round complexity of our protocol is very low. Firstly, sending the encrypted face image takes one move. Secondly, for outsourced computation, it needs 5 moves between the database owner and the cloud server. In the last step of FMP, sending the result of FMP takes one move between the client and the cloud server. Hence the overall round cost is 7 moves. In addition, we note that it has only 3 moves if the database owner completes exponentiations by himself rather than the cloud server in both projection and calculation distance phases. Therefore, the round complexity of our protocol is $O(1)$. Furthermore, our protocol does not require the database owner interaction with the client for calculation distance, but the database owner needs interactions with the cloud server if partly computation task is outsourced

Table 2 Comparison of three minimum protocols.

Protocol	[16]	[17]	Ours
Round Complexity [moves]	$6\lceil logM\rceil+1$	3	2
ACC1 (C online)	$2M\text{Dec}_{PE}+l'M\text{Dec}_{DGK}$	$m\text{Dec}_{PE}$+$3l'M$ Hash	2 Dec_{FHE}
ACC1 (DO online)	$2M\text{Exp}_{PE}+l'M\text{Exp}_{DGK}$	$m\text{Exp}_{PE}$	1 Enc_{PE}

to the cloud server, in which it can reduce the computation cost of the database owner.

5.2 Online communication complexity

For communication complexity, the communication complexity highly depends on the size of Paillier encryption, FHE, and outsourcing exponentiation algorithm. In the offline phase, the circuit $C_{circuit}$ with VOC can be transmitted to the cloud server. In the following, we only analyze the online communication complexity.

- $\boldsymbol{C \rightarrow DB}$. For the client sends some encrypted data to the database owner, the client requires transmission of 1 FHE encrypted value $[sk]_{FHE}$, (N+3) Paillier encrypted values $\{[\Gamma_1]_{PE},\cdots,[\Gamma_N]_{PE},[a]_{PE},[b]_{PE},[c]_{PE}\}$.
- $\boldsymbol{DB \rightleftharpoons CS}$. In the distance calculation phase, for $i=1,\cdots,K$, to obtain $[\bar{\omega_i}]_{PE}=[p_i]_{PE}\cdot q_i$, the database owner outsources the computation $q_i=\prod_{j=1}^{N}[\Gamma_j]_{PE}^{u_{i,j}}$ to CS by using outsourced computation algorithm for simultaneous exponentiation (Sexp) in [38] in the projection stage. Therefore, it needs $K\cdot\lceil N/2\rceil$ operations of Sexp, which means that the communication overhead is $K\cdot\lceil N/2\rceil\cdot(6logp+12Len_{[\Gamma_j]_{PE}})$, where p is the bit length of $u_{i,j}$ and $Len_{[\Gamma_j]_{PE}}$ is the bit length of $[\Gamma_j]_{PE}$. In the distance computation stage, it needs $K\cdot\lceil N/2\rceil$ operations of Sexp for computing $[S_{2,i}]_{PE}$, which means that the communication overhead is $K\cdot\lceil N/2\rceil\cdot(6logp'+12Len_{[\bar{\omega_j}]_{PE}})$ bits, where p' is the bit length of $-2\omega_{i,j}$ and $Len_{[\bar{\omega_j}]_{PE}}$ is the bit length of $[\bar{\omega_j}]_{PE}$.
- $\boldsymbol{DB \rightarrow CS}$. In the minimum (match) finding stage, the circuit $C_{circuit}$ can be transmitted in the offline stage. It requires transmission of 1 FHE encrypted value and $(K+2M+4)$ PE encrypted values in the online stage. Similar to [16,17], we only requires transmission of $(K+M+4)$ PE encrypted values if we omit the statistic for the transmission of $[id]_{PE}$.
- $\boldsymbol{CS \rightarrow C}$. In the minimum (Match) finding stage, it requires transmission of 2 FHE encrypted value $[\Delta_1]_{FHE}$ and $[\Delta_2]_{FHE}$.

Suppose that the size of FHE-ciphertexts is γ bits. Similar to [16,17], we omit the statistic for the transmission of $[id_i]_{PE}$. Let k be the number of the packed ciphertexts in [17]. We now compare our protocol with the previous works [16,17] as shown in Table 3. Unfortunately, the online communication cost of our protocol is larger than the precious works because we use outsourced computation algorithms which means that the outsourcer requires some interactions with the cloud server.

5.3 Online computation complexity

The overall online computation complexity of our protocol is substantially lower. We denote by Enc_{PE} an invocation of the Paillier homomorphic encryption algorithm, by Dec_{PE} an invocation of the Paillier homomorphic decryption algorithm, by Enc_{FHE} an invocation of the FHE algorithm, by Dec_{FHE} an invocation of the fully homomorphic decryption algorithm, by MM a modular multiplication, by MInv a modular inverse, by Exp a modular exponentiation, by Sexp_{VOC} an invocation of the verifiable outsourced computation algorithm for simultaneous exponentiation. We omit other operations such as modular additions. More precisely, the online computation cost of the client and the database owner is given as follows.

- In the projection phase, the client needs (N+3)Enc_{PE} and 1 Enc_{FHE}, and the database owner requires K Enc_{PE}, $K\cdot\lceil N/2\rceil$ Sexp_{VOC} and $(K\lceil N/2\rceil)$ MM.
- In the distance computation phase, the database owner needs M Enc_{PE}, $(M\cdot\lceil K/2\rceil)\text{Sexp}_{VOC}$ and $(M\cdot\lceil K/2\rceil)$MM.
- In the minimum distance (or match)finding phase, the client needs 2Dec_{FHE}. Because the database owner needs to encrypt $\{\tau,id_0,id_1,\cdots,id_M\}$ under PE, the database owner needs $(M+2)\text{Enc}_{PE}$ in the online phase.

Compared with the previous algorithms in [16-17], our protocol is superior in efficiency due to the reduction of the computation cost of the client and the database owner. Similar to [16,17], we omit the statistic for the computation cost of the encryption $\{id_i\}_{i=0}^{M}$ under PE. Table 4 presents the comparison of the online computation cost of the client and the database owner in the three algorithms. In particular, beyond the encryption operations, we note that the overall computation cost of the client and the database owner in our

Table 3 Comparison of round and asymptotic communication complexity(ACC).

Protocol	[16]	[17]	Ours
Round Complexity	$O(logM)$	$O(1)$	$O(1)$
Moves	$6\lceil log(M+1)\rceil + 4$	6	3 (or 7)
ACC (online, [bits])	$2T(l'M + K + N + 1 + 8M)$	$2T(l'M + k + m + N + 1)$	$6K \cdot \lceil N/2\rceil \cdot (logp' + 8T + logp)$ $+2T(K + N + M + 7) + 4\gamma$

Table 4 Comparison of asymptotic computation complexity (online)

Protocol	[16]	[17]	Ours
C	$(N+1)\mathrm{Enc}_{\mathrm{PE}} + (K+2M)\mathrm{Dec}_{\mathrm{PE}}$ $+(l'M)\mathrm{Dec}_{\mathrm{DGK}} + K\mathrm{MM}$	$(N+1)\mathrm{Enc}_{\mathrm{PE}} + (k+m)\mathrm{Dec}_{\mathrm{PE}}$ $+(3l'M)\mathrm{Hash} + K\mathrm{MM}$	$(N+3)\mathrm{Enc}_{\mathrm{PE}} + 1\mathrm{Enc}_{\mathrm{FHE}}$ $+2\mathrm{Dec}_{\mathrm{FHE}}$
DO	$(K+M)\mathrm{Enc}_{\mathrm{PE}}$ $+((l'+K+2)M + (N+1)K)\mathrm{Exp}$ $+((N+M+2)K + 1 - M)\mathrm{MM}$	$(K+\mathbf{M})\mathrm{Enc}_{\mathrm{PE}}$ $+(KN + KM + k + 1 + m)\mathrm{Exp}$ $+(KN + M(K-1) + 1 + \lceil K/k\rceil)\mathrm{MM}$	$(M+K+1)\mathrm{Enc}_{\mathrm{PE}}$ $+(K\lceil N/2\rceil + M\lceil K/2\rceil)\mathrm{Sexp}_{\mathrm{VOC}}$ $+(K\lceil N/2\rceil + M\lceil K/2\rceil)\mathrm{MM}$
Sum	$(K+M+N+1)\mathrm{Enc}_{\mathrm{PE}}$ $+(K+2M)\mathrm{Dec}_{\mathrm{PE}} + (l'M)\mathrm{Dec}_{\mathrm{DGK}}$ $+((l'+K+2)M + (N+1)K)\mathrm{Exp}$ $+((N+M+3)K + 1 - M)\mathrm{MM}$	$(K+M+N+1)\mathrm{Enc}_{\mathrm{PE}}$ $+(k+m)\mathrm{Dec}_{\mathrm{PE}} + (3l'M)\mathrm{Hash}$ $+(KN + KM + k + 1 + m)\mathrm{Exp}$ $+(K(N+1+M) - M + 1 + \lceil K/k\rceil)\mathrm{MM}$	$(K+M+N+4)\mathrm{Enc}_{\mathrm{PE}} + 1\mathrm{Enc}_{\mathrm{FHE}}$ $+2\mathrm{Dec}_{\mathrm{FHE}}$ $+(K\lceil N/2\rceil + M\lceil K/2\rceil)\mathrm{Sexp}_{\mathrm{VOC}}$ $+(K\lceil N/2\rceil + M\lceil K/2\rceil)\mathrm{MM}$

protocol is at most 1/2 of the corresponding protocol in the state of the art algorithms [16,17].

6 Conclusion

In this paper, we present a privacy-preserving face recognition scheme with outsourced computation, which allows to match an encrypted image showing a face against a database of facial templates in such a way that the biometric itself and the detection result is hidden from the server that performs the matching. In particular, our protocol allows the database owner to securely outsource some computation task to an untrusted cloud server and detect the dishonest behavior of untrusted cloud server. Furthermore, the client can verify the correctness of the recognition result. Compared with the state-of-the-art algorithms [16-17], beyond the same operations for encryption, the overall online computation cost of the database owner and the client is greatly reduced. However, the online communication cost cannot reduce due to using outsourced computation. Thus, the key problem of our protocol is that how to further reduce the online communication cost and the client's computation cost on the future work.

Acknowledgements This work is supported in part by the National Natural Science Foundation of China under Grant No. 11271003, the National Research Foundation for the Doctoral Program of Higher Education of China under Grant No. 20134410110003, High Level Talents Project of Guangdong, Guangdong Provincial Natural Science Foundation under Grant No. S2012010009950, the Project of Department of Education of Guangdong Province under Grant No 2013KJCX0146, and the Natural Science Foundation of Bureau of Education of Guangzhou under Grant No. 2012A004.

References

1. Internfgational Civil Aviation Organization (ICAO). Machine Readable Travel Documents (MRTD), Doc 9303, Part 1, Fifth Edition, 2003.
2. Naumann I, and Hogben G. Privacy features of European aid card specifications. Network Security, European Network and Information Security Agency (ENISA), vol. 8, pp. 9-13, 2008.
3. Juels A, and Sudan M. A fuzzy vault scheme. In: International Symposium on Information Theory, 2002.
4. Chang E-C, and Li Q. Small secure sketch for point-set difference. Cryptology ePrint ArchiveReport 2005/145, 2005.
5. Chang E-C, and Li Q. Hiding secret points amidst chaff. In: Advances in Cryptology-EUROCRYPT, vol. LNCS 4004, pp. 59-72, 2006.
6. Juels A, and Wattenberg M. A fuzzy commitment scheme. In ACM Conference on Computer and Communications Security (CCS), pp: 28-36, 1999.
7. Dodis Y, Reyzin L, and Smith A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Advances in Cryptology- EUROCRYPT, vol. LNCS 3027, pp. 523-540, 2004.
8. Boyen X. Reusable cryptographic fuzzy extractors. In ACM Conference on Computer and Communications Security (CCS), pp. 82-91, 2004.
9. Dodis Y, and Smith A. Correcting errors without leaking partial information. In ACM Symposium on Theory of Computing (STOC), pp. 654-663, 2005.
10. Boyen X, Dodis Y, Katz J, Ostrovsky R, and Smith A. Secure remote authentication using biometric data. In Advances in Cryptology-EUROCRYPT, vol. LNCS 3493, pp. 147-163, 2005.
11. Dodis Y, Ostrovsky R, Reyzin L, and Smith A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM Journal of Computing, vol. 38, no. 1, pp. 97-139, 2008.
12. Linnartz J, and Tuyls P. New shielding functions to enhance privacy and prevent misuse of biometric templates. In International Conference on Audio and Video Based Biometric Person Authentication, June, 2003.
13. Verbitskiy E, Tuyls P, Denteneer D, and Linnartz J. Reliable biometric authentication with privacy protection. In

Benelux W.I.C. Symposium on Information Theory, pp. 125-131, 2003.
14. Tuyls P, Akkermans A, Kevenaar T, Schrijen G.-J, Bazen A, and Veldhuis G. Practical biometric authentication with template protection. In Audio- and Video-Based Biometric Person Authentication (AVBPA), vol. LNCS 3546, pp. 436-446, 2005.
15. Ratha NK, Connell JH, and Bolle RM. Enhancing security and privacy in biometrics-based authentication systems. IBM Systems Journal, vol. 40, no. 3, pp. 614-634, 2001.
16. Erkin Z, Franz M, Guajardo J, Katzenbeisser S, Lagendijk I, and Toft T. Privacy-preserving face recognition. In Privacy Enhancing Technologies (PET'09), vol. LNCS 5672, pp. 235-253, 2009.
17. Sadeghi AR, Schneider T, and Wehrenberg I. Efficient Privacy-Preserving Face Recognition. Information, Security and Cryptology-ICISC 2009, vol. LNCS 5984, pp. 229-244, 2010.
18. Senior A, Oankanti A, Hampapur A, et al. Enabling video privacy through computer vision. IEEE Security and Privacy Magazine, vol. 3, no. 3, pp. 50-57, 2005.
19. Dufaux F, and Ebrahimi T. Scrambling for video surveillance with privacy. In: Proceedings of IEEE Workshop on Privacy Research In Vision, New York, p. 160, June 2006.
20. Yu X, Chinomi K, Koshimizu T, Nitta N, Ito Y, and Babaguchi N. Privacy protecting visual processing for secure video surveillance. In: Proceedings of the International Conference on Image Processing (ICIP'08), October 2008, San Diego, Calif, USA, pp. 1672-1675, 2008.
21. Turk M, and Pentland A. Eigenfaces for recognition. Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991.
22. Turk MA, and Pentland AP. Face recognition using eigenfaces. In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR'91), pp. 586-591, 1991.
23. Paillier P. Public-key cryptosystems based on composite degree residuosity classes. In Advances in Cryptology-EUROCRYPT'99, vol. LNCS 1592, pp. 223-238, 1999.
24. Damgard IB, and Jurik MJ. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In Public-Key Cryptography (PKC'01), vol. LNCS 1992, pp. 119-136, 2001.
25. Damgard I, Geisler M, and Kroigard M. Efficient and secure comparison for online auctions. In Australasian Conference on Information Security and Privacy (ACISP'07), vol. LNCS 4586, pp. 416-430, 2007.
26. Damgard I, Geisler M, and Kroigard M. A correction to "efficient and secure comparison for on-line auctions". Cryptology ePrint Archive, Report 2008/321, http://eprint.iacr.org/2008/321, 2008.
27. Damgard I, Geisler M, and Kroigard M. Homomorphic encryption and secure comparison. Journal of Applied Cryptology, vol. 1, no. 1, pp. 22-31, 2008.
28. Yao AC. How to generate and exchange secrets. In The Computer Society of IEEE, 27th Annual Symp. on Foundations of Computer Science (FOCS), pp. 162-167, 1986.
29. Lindell Y, and Pinkas B. A proof of Yao's protocol for secure two-party computation. Cryptology ePrint Archive, Report 2004/175, 2004.
30. Avidan S, and Butman M. Efficient methods for privacy preserving face detection. In Advances in Neural Information Processing Systems (NIPS'06), pp. 57-64, 2006.
31. Newton EM, Sweeney L, and Malin B. Preserving privacy by de-identifying face images. IEEE Transactions on Knowledge and Data Engineering, vo. 17, no. 2, pp. 232-243, 2005.
32. Tuyls P, Akkermans AHM, Kevenaar TAM, Schrijen G.-J, Bazen AM, and Veldhuis RNJ. Practical biometric authentication with template protection. In: Inter. Conf. on Audio- and Video-Based Biometric Person Authentication (AVBPA 2005). vol. LNCS 3546, pp. 436-446, 2005.
33. Ratha N, Connell J, Bolle R, and Chikkerur S. Cancelable biometrics: A case study in fingerprints. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR), vol. IV, pp. 370-373, 2006.
34. Kevenaar T. Protection of Biometric Information. In: Security with Noisy Data, pp. 169-193, 2007.
35. Blanton M, and Aliasgari M. Secure Outsourced Computation of Iris Matching.
36. Li ZZ, and Lai TH. Deterministic Fully Homomorphic Encryptions for Privacy Preserving Cloud Computing, http://web.cse.ohio-state.edu/-lizh/.
37. Dijk MV, Gentry C, Halevi S, and Vaikuntanathan V. Fully homomorphic encryption over the integers, Advances in Cryptology, ASIACRYPT 2010, Vol. LNCS 6110, pp. 24-43, 2010.
38. Wang YJ, Wu QH, Wong DS, Qin B, Chow SSM, Liu Z, Tan X. Securely Outsourcing Exponentiations with Single Untrusted Program for Cloud Storage. ESORICS (1), pp. 326-343, 2014.