Privacy Preserving Face Identification in the Cloud through Sparse Representation

Xin Jin[1,*], Yan Liu[1,2], Xiaodong Li[1,], Geng Zhao[1]
, Yingya Chen[1], and Kui Guo[1]

[1]Beijing Electronic Science and Technology Institute, Beijing 100070, China,
GOCPCCC Key Laboratory of Information Security, Beijing 100070, China
[2]Xidian University, Xi'an, 710071, China
{jinxin,lxd}@besti.edu.cn

Abstract. Nowadays, with tremendous visual media stored and even processed in the cloud, the privacy of visual media is also exposed to the cloud. In this paper we propose a private face identification method based on *sparse representation*. The identification is done in a secure way which protects both the privacy of the subjects and the confidentiality of the database. The face identification server in the cloud contains a list of registered faces. The surveillance client captures a face image and require the server to identify if the client face matches one of the suspects, but otherwise reveals no information to neither of the two parties. This is the first work that introduces sparse representation to the secure protocol of private face identification, which reduces the dimension of the face representation vector and avoid the patch based attack of a previous work. Besides, we introduce a secure Euclidean distance algorithm for the secure protocol. The experimental results reveal that the cloud server can return the identification results to the surveillance client without knowing anything about the client face image.

Keywords: Privacy Preserving, Face Identification, Private Computing, Sparse Representation, Cloud Computing

1 Introduction

Face recognition has played an important role in surveillance and security. Nowadays, cloud computing has changed the way of traditional face recognition system. The big data of face images or videos and powerful face recognition program have been stored and running in the cloud server, which supports large scale video surveillance applications such as face tracking, suspect searching.

However, tremendous surveillance cameras have distributed everywhere. The privacy of people in the surveillance videos from the public places is being violated. The suspect searching applications can be misused to track the wanted person of criminals. Once the face recognition system is linked to a universal database such as ID cards, civilians could be tracked as someone's wishes. On the other hand, the suspect list could be released to public, which may cause more crimes.

In this work, as show in Fig. 1, our scenario is set as that the surveillance client captures a face image and send to the cloud server which contains a list of suspects. The server compares the client face to each suspect in the list. The match or not match results are returned to the client. Using current face recognition methods, the contents of client face image are completely known to the server. Meanwhile, the client can guess out who are in the suspect list through several times of face identification.

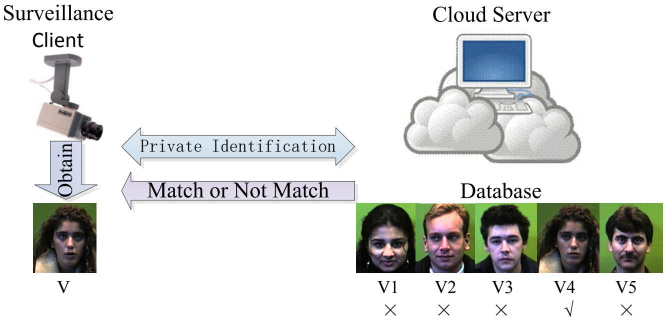

Fig. 1. The application scenario. The surveillance client capture a face image from public places such as airport, railway station. The face image is identified through our privacy preserving method with the suspect face data in the cloud server. After that, the client only learns the matching results. The cloud server learns nothing.

To protect the privacy of both the face captured by the client and the suspects in the cloud server, we propose a privacy preserving face identification method based on sparse representation and several cryptography tools. The client only knows the matching result. The cloud server knows nothing.

Recently, a system called Secure Computation of Face Identification (SCiFI) [1] was developed. This system use two cryptography tools (homomorphic encryption and oblivious transfer) to implement a privacy preserving computation of the Hamming distance between two binary vectors. Each of the face in both the client and the server is represented as a binary vector using a local image patch based method with the assistant of a third party face database.

The SCiFI system [1] has two main drawbacks: (1) the dimension of the face representation vector is large because of the binary representation, which reduce the running efficiency of the system, (2) the local patch based representation could be attracted by reconstructing a fragmented face [2]. Recently, Luong et al. [2] has proposed a method of reconstructing a fragmented face to attack the SCiFI system and reconstruct faces from the secure identification protocol.

We employ sparse representation to reduce the dimension of the face representation vector of [1] and avoid the patch based attack. As the classical sparse representation method does, a dictionary is learned from the list of suspects in the server. However, one can recover faces in the server easily using the learned dictionary. Thus we also add a third party face database and learn the dictionary form it. Then we use the sparse parameters as the representation vector of a face. After that, a privacy preserving computing method of Euclidean distance between two sparse parameter vectors is proposed. We extent the privacy preserving hamming distance of [1] to privacy preserving Euclidean distance. Experimental results reveal that our method can achieve comparable correction rate of identification to the local patch based method and need less computing time. Besides, we break out the restrict of binary representation of face when using the cryptographic tools of homomorphic encryption and obvious transfer. This will make converting modern face recognition algorithms to privacy preserving methods become possible in future work, which will make the face recognition and other computer vision algorithms running in the cloud more secure and less privacy leaking.

2 Background

In this section we briefly introduce two cryptography primitives(Homomorphic Encryption and Oblivious Transfer) and some basic parts of SCiFI system [1], just as what have been discussed in [2].

2.1 Cryptography Primitives

Homomorphic Encryption. The Paillier cryptosystem [3] [4], named after and invented by Pascal Paillier in 1999, is a probabilistic asymmetric algorithm for public key cryptography. The problem of computing n-th residue classes is believed to be computationally difficult. The decisional composite residuosity assumption is the intractability hypothesis upon which this cryptosystem is based. The scheme is an additive homomorphic cryptosystem; this means that, given only the public-key and the encryption of m_1 and m_2, one can compute the encryption of $m_1 + m_2$. The two properties of Paillier system are described as:

$$\begin{aligned} \mathrm{E}(m_1) &\equiv g^{m_1} \cdot x_1^N (mod\ N^2) \\ \mathrm{E}(m_2) &\equiv g^{m_2} \cdot x_2^N (mod\ N^2) \\ \mathrm{E}(m_1) \cdot \mathrm{E}(m_2) &\equiv g^{m_1} \cdot x_1^N \cdot g^{m_2} \cdot x_2^N mod\ N^2 \\ &\equiv g^{m_1+m_2} (x_1 \cdot x_2)^N mod\ N^2 \\ &\equiv \mathrm{E}(m_1 + m_2) \end{aligned} \tag{1}$$

$$\mathrm{E}(km_1) = \mathrm{E}(m_1)^k \tag{2}$$

where m_1 and m_2 are 2 plan texts. $E(\cdot)$ is the encryption function. $N = p \cdot q$, p and q are two large prime numbers. $N \in Z$. The plan text $m \in Z_N$. x is a random number. $x \in Z_N^*$. $\mathrm{gcd}(\mathrm{L}(g^e \ mod \ N^2), N) = 1$. e is the encryption key. Z, Z_N^*, e and $L(\cdot)$ are defined as:

$$\begin{aligned} Z_N &= \{x | x \in Z, 0 \leq x < N\}, \\ Z_N^* &= \{x | x \in Z, 0 \leq x < N, \mathrm{gcd}(x, N) = 1\}, \\ e &= \mathrm{lcm}(p-1, q-1), \\ S &= \{x < N^2 | x = 1 \ mod \ N\}, \\ \forall x \in S, \mathrm{L}(x) &= \frac{x-1}{N} \end{aligned} \tag{3}$$

where lcm means Least Common Multiple. gcd means Greatest Common Divisor.

Obvious Transfer. In cryptography, an oblivious transfer protocol (often abbreviated OT) is a type of protocol in which a sender transfers one of potentially many pieces of information to a receiver, but remains oblivious as to what piece (if any) has been transferred. The first form of oblivious transfer was introduced in 1981 by Michael O. Rabin. In this form, the sender sends a message to the receiver with probability 1/2, while the sender remains oblivious as to whether or not the receiver received the message. Rabin's oblivious transfer scheme is based on the RSA cryptosystem. A more useful form of oblivious transfer called 1-2 oblivious transfer or "1 out of 2 oblivious transfer," was developed later by Shimon Even, Oded Goldreich, and Abraham Lempel , in order to build protocols for secure multiparty computation. It is generalized to "1 out of n oblivious transfer" where the user gets exactly one database element without the server getting to know which element was queried, and without the user knowing anything about the other elements that were not retrieved. The latter notion of oblivious transfer is a strengthening of private information retrieval, in which the database is not kept private [5].

2.2 SCiFI Overview

The SCiFI system proposes a face representation method to represent a face image as a n dimensions binary vector $\mathbf{w} = [w_0, w_2, ..., w_{n-1}]$ using a public face database. The cloud server contains a list of M face binary vectors $\{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_M\}$ and thresholds $\{t_1, t_2, ...t_M\}$. The output of the protocol is R:

$$R = \begin{cases} \text{match, if } \mathrm{H}(\mathbf{w}, \mathbf{w}_i) < t_i \\ \text{not match, if otherwise} \end{cases} \tag{4}$$

, where $\mathrm{H}(\cdot)$ is the Hamming distance of two binary vectors.

The client uses the Paillier cryptosystem [3] to share the public key with the server and keeps the private key to itself. Through an oblivious transfer protocol, the client learns only if the Hamming distance between any pair of their vectors exceeds a threshold. The cloud server learns nothing. See [1] for implementation details. [2]

3 Privacy Preserving Face Identification

Algorithm 1 Private Face Identification

```
Input:
    The client's input is a face vector $\mathbf{s} = (s_0, s_1, ..., s_{l-1})$. In our application $l = 200$.
    The server's input is a list of $Q$ face vectors $\{s^1, s^2, ..., s^Q\}$. The server has additional
    inputs $\{t_1, t_2, ..., t_Q\}$ for each $s^i$. The two parties both know an upper bound $d_{max}$,
    in our application we set
    $d_{max} \leq 1 \times 10^6$.
Output:
    The client learns the indices $i$ for which $ED(s, s^i) \leq t_i$. The server learns nothing.
    The client uses Paillier to encrypt and send face vector $s = (s_0, s_1, ..., s_{l-1})$ item
    by item and the square of each item $(s)^2 = ((s_0)^2, (s_1)^2, ..., (s_{l-1})^2)$. The cloud
    server receives the encryption results of each item $(E_{pk}(s_0), E_{pk}(s_1), ..., E_{pk}(s_{l-1}))$
    and $(E_{pk}((s_0)^2), E_{pk}((s_1)^2), ..., E_{pk}((s_{l-1})^2))$. For each face in the list of suspects
    of the server, the following steps are repeated.
    For $i$th face in the server and for $j$th parameter in the face vector, the cloud server
    computes $E_{pk}(v_j)$, where $v_j = (s_j - s_j^i)^2$.

        $$\begin{aligned} E_{pk}((s_j - s_j^i)^2) &= E_{pk}((s_j)^2 - 2s_js_j^i + (s_j^i)^2) \\ &= E_{pk}((s_j)^2) \cdot E_{pk}(s_j)^{-2s_j^i} \cdot E_{pk}((s_j^i)^2) \end{aligned} \quad (5)$$

    The $s_j^i$ is known to the server.
    According to the properties of homomorphic encryption, the cloud server can
    compute the encrypted $d_E = (ED(s, s^i))^2$ by $E_{pk}(d_E) = \sum_{j=0}^{l-1} E_{pk}(v_j)$, $d_E \in$
    $[0, d_{max}]$. Then the server chooses a random number $r_i$ for each face vector, and
    computes $E_{pk}((ED(s, s^i))^2 + r_i)$. This number is sent to the client.
    The client receives the $E_{pk}((ED(s, s^i))^2 + r_i)$ and decrypts it.
    The two parties use $OT_1^{\frac{d_{max}}{4000}}$ protocol to judge if $(d_E)^i < t_i$ securely. The result
    $R_i$ computed in the client is:

        $$R_i = \begin{cases} 1 \text{ if } ((d_E)^i + r_i) \ mod \ (d_max + r_i) \leq t_i + r_i \\ 0 \text{ if otherwise} \end{cases} \quad (6)$$

    return $R_i$.
```

An overview of our method is shown in Fig. 2. In this section, we first introduce the modified spares represent based face identification. Then the private face identification protocol using our modified sparse representation is described.

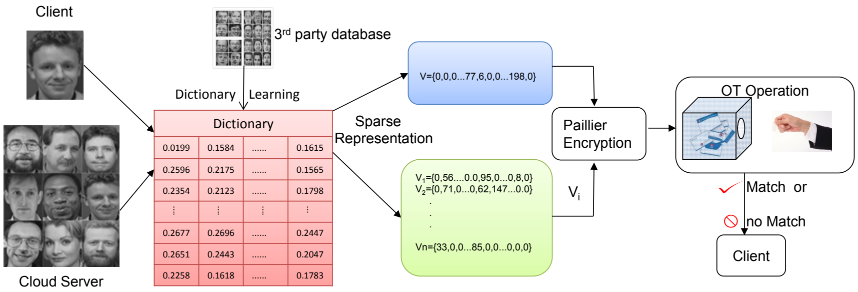

Fig. 2. The overview of our method. A third party face database is used to learn a dictionary. The face captured by the client and the faces in the list of the cloud server are represented sparse parameter vector. The Euclidean distance of the client face vector and each of the face vector in the server is computed in a privacy preserving way. The matching result is only known by the client. The cloud server learns nothing.

3.1 Modified Sparse Representation based Face Identification

The local image patch based face representation of SCiFI [1] is attacked by [2]. Besides, the dimension of binary face vector is 3000 (we have 100 face images in the $3rd$ database), which reduces the computing efficiency using cryptographic tools. Thus we introduce sparse representation into the secure protocol of SCiFI to reduce the dimension of face vector through sparse parameters and avoid fragment attack proposed by [2].

However, introducing the sparse representation to the secure protocol is a non-trivial work. If the dictionary is learned from the list of faces in the cloud server, the contents of the suspect list will be leaked to public. Besides, solving the linear system in the secure protocol based on Paillier system and oblivious transfer is too complicated and time consuming. Thus, we modify the classical sparse representation based face identification [6] from two aspects. (1) We add a third party face database to learn the dictionary. (2) We directly use the sparse parameter vector as the representation of a face image. The Euclidean distance is considered as the similarity criteria between two faces. Thus, we can compute the square of Euclidean distance in the secure protocol, which is less time consuming than solving a linear system.

We denote the face vector of our sparse parameter as $\mathbf{s} = \{s_1, s_2, ...s_l\}$. The square of Euclidean distance between face vector $\mathbf{s}^1$ and $\mathbf{s}^2$ is $ED(\mathbf{s}^1, \mathbf{s}^2) = \sum_{i=1}^{l}(s_i^1 - s_i^2)^2$. If the square of Euclidean distance is below a threshold, we consider that $\mathbf{s}^1$ and $\mathbf{s}^2$ belong to the same face. Learning individual thresholds is a hard task because these thresholds depend on variations in different images of the same face.

We learn the individual threshold for each face in the cloud server by the set difference for each person that will discriminate him/her from an ensemble of people. The threshold for the ith face is based on the smallest set difference between him and the rest of people in the ensemble [1].

3.2 Private Face Identification Protocol

We modify the secure protocol of SCiFI [1]. In our experiment, the dimension of a face vector $\mathbf{s} = \{s_1, s_2, ...s_l\}$ is $l = 200$. the max square of Euclidean distance of two face vector is $d_{max} = 1 \times 10^6$. We divide it equally to 250 parts for approximation. Each part has 4000 elements Thus when we implement the obvious transfer, only 250 pairs of public and private keys are needed. We use a Paillier encryption function, $E_{pk}(\cdot)$. pk is a public key that both parties know. The client knows the corresponding private key and decrypt messages. The complete privacy preserving face identification protocol is described in Algorithm 1.

4 Experimental Results

We test our method and compare the performance with SCiFI [1] in the faces94 dataset [7]. The server has totally 100 facial images of 20 persons. Each person has 5 different facial images. The third face database also contains 100 facial image of 20 different persons to the server. The $3rd$ face database contains 100 facial images randomly selected from the faces94 dataset. The average top one matching rate of our method is 91.55% which is a little less than 95.5% of the local image patch based method of [1]. However, we use only 200 dimension vector comparing with 3000 dimension vector of [1]. Furthermore, the fragment reconstruction based attack method by [2] cannot attack our method any more, because we do not use any fragment of facial images.

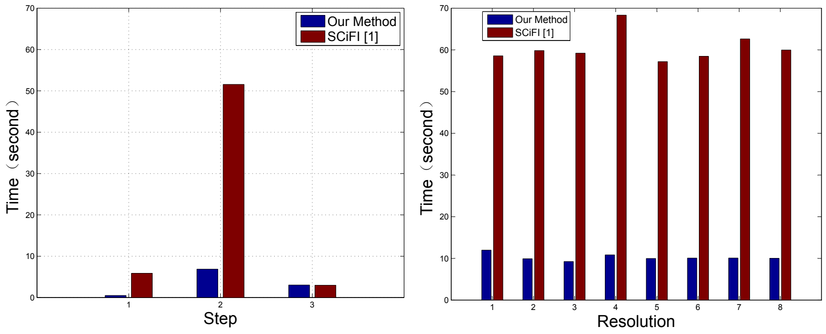

Fig. 3. The computing time of each main step. $step1$ for face vector generation, $step2$ for the Paillier encryption, $step3$ for the oblivious transfer. The computing times of face vector generation and the Paillier encryption of our method are less than those of SCiFI [1]. The resolutions tested are $138 * 168, 230 * 280, 276 * 336, 322 * 392, 38 * 448, 414 * 504, 460 * 560$.

In addition, the sparse representation make our face vector much shorter than that of SCiFI [1], which reduces the computing time using time-consuming cryptography algorithm. The Paillier encryption is called each item of the face vector.

We test the computing time of each main step. In *step*1 the sparse face vector is computed. In *step*2 the Paillier encryption is called. In *step*3 the oblivious transfer is executed. To avoid network delay, we set the client and server in the same PC (Windows 32, 2.92GHz Intel Core2 Duo CPU, 3GB RAM). As shown in Fig. 3, the computing times of face vector generation and the Paillier encryption of our method are less than those of SCiFI [1]. All the time in this experiment is the average time of 10 facial images.

The computing times in different resolutions are also compared with those of SCiFI [1]. The total computing times of each resolutions are shown in Fig. 3. Our method is obvious faster than the method of SCiFI [1].

5 Conclusion and Discussion

In this paper we propose a private face identification method based on sparse representation. The identification is done in a secure way which protects both the privacy of the subjects and the confidentiality of the database. This is the first work that introduces sparse representation to the secure protocol of private face identification, which reduces the dimension of the face representation vector and avoid the patch based attack of a previous work.

6 Acknowledgements

This work is partially supported by the National Natural Science Foundation of China (No.61402021, No.61402023, No.61170037), the Fundamental Research Funds for the Central Universities (No.2014XSYJ01, No.2015XSYJ25), and the Science and Technology Project of the State Archives Administrator. (No.2015-B-10).

References

1. Osadchy, M., Pinkas, B., Jarrous, A., et al. SCiFI - A system for Secure Face Identification IEEE Symposium on Security and Privacy (S&P), pp.239-254, IEEE (2010)
2. Luong, A., Gerbush, M., Waters, B., Grauman, K. Reconstructing a fragmented face from a cryptographic identification protocol. IEEE Workshop on Applications of Computer Vision (WACV), pp.238-245, IEEE (2013)
3. Paillier, P. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. EUROCRYPT, pp.223-238, Springer (1999)
4. Paillier Cryptosystem. `https://en.wikipedia.org/wiki/Paillier_cryptosystem`
5. Oblivious Transfer. `https://en.wikipedia.org/wiki/Oblivious_transfer`
6. Wright J., Ganesh, A., et al.Robust Face Recognition via Sparse Representation IEEE Trans Pattern Anal Mach Intell, Vol. 31, No. 2, pp.210 - 227, IEEE (2008)
7. Collection of Facial Images: Faces94. `http://cswww.essex.ac.uk/mv/allfaces/faces94.html`