Soft Comput (2017) 21:4349–4359
DOI 10.1007/s00500-016-2066-5

METHODOLOGIES AND APPLICATION

Privacy-preserving outsourcing of image feature extraction in cloud computing

Ping Li[1] · **Tong Li**[2] · **Zheng-An Yao**[1] · **Chun-Ming Tang**[3] · **Jin Li**[4]

Published online: 12 February 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract Private image data, especially including the biometric data with an authentication property, has received more and more attention along with the development of researches on big data. Consequently, to protect the private image data while enabling outsourced image computations becomes a major concern. In this present paper, we study the privacy-preserving face recognition by using a method that is different from the method of fuzzy classification recognition, which is scale-invariant feature transform (SIFT) as our key technical tool. We first propose a scheme in which the client encrypts his private image data locally and outsources the corresponding results to a company. The latter performs most of the computations, but remains ignorant of the original data. To prevent some potential adversary from forging the data, we introduce a third party who can decrypt any given valid ciphertext. Based on these ideas, we adopt the BCP double-decryption cryptosystem for our scheme. Some analyses show that our proposed scheme is secure, efficient and scalable.

Keywords Cryptography · Privacy preserving · Face recognition · Fuzzy recognition · Homomorphic encryption

Communicated by V. Loia.

✉ Ping Li
liping26@mail2.sysu.edu.cn

Tong Li
litongziyi@mail.nankai.edu.cn

Zheng-An Yao
mcsyao@mail.sysu.edu.cn

Chun-Ming Tang
ctang@gzhu.edu.cn

Jin Li
lijin@gzhu.edu.cn

1 School of Mathematics and Computational Science, Sun Yat-sen University, 510275 Guangzhou, China

2 College of Computer and Control Engineering, Nankai University, 300071 Tianjin, China

3 School of Mathematics and Information Science, Guangzhou University, 510006 Guangzhou, China

4 School of Computational Science and Education Software, Guangzhou University, 510006 Guangzhou, China

1 Introduction

Biometric techniques have rapidly developed over the past decades to a reliable means of authentication, which are increasingly deployed in various application areas. For example, face recognition, electrocardiograms (ECG) signal, finger code and iris code can be used for biometric authentication. In particular, feature point detection and matching as one step of the biometric authentication is an essential research topic with wide current interest in many computational vision applications. Note that some specific locations may exist in images, such as mountain peaks, building corners or doorways. These kinds of localized features usually called *keypoint features* or *interest points* are recognized as important and invariant features that can be utilized for the design of content authentication, face recognition and robust watermarking. There are many image recognition classification recognition methods, template matching, statistical classification and fuzzy classification recognition. Specific methods, such as PCA, FLD, Harris, Mexican-Hat wavelet filtering and fuzzy K means clustering, have been widely used in various applications.

Recently, scale-invariant feature transform SIFT is a hot issue in computer vision to extract and describe local features in image. Low (2004) showed that there are four major com-

putational stages to generate a set of image features; here we denote two points:

Scale space extreme detection: Use a Difference-of-Gaussian (DoG) function to convolve with image at multiple scales, then the potential feature points are chosen as local extremes of the DoG images across scales, which are invariant with respect to scale and orientation. Each sample point is compared with its eight neighbors (called pixels) in the current scale and nine corresponding neighbors above and below. If this sample point is the extreme among all neighbors, then it is selected as a key point.

Keypoint descriptor: Based on the gradient directions of the local image, each key point location is assigned orientations. The descriptor is represented as a vector containing the values of all assigned orientation histogram entries.

However, because of the importance of the feature key point, if the attacker compares with feature key point in a benchmark image database, he can deduce the content of the image or even recover part of the image based on these feature key points. Meanwhile, the ubiquitous use of face biometrics raises important privacy concerns, such as the increasing deployment of surveillance cameras in public places: bank, airport, railway station and even where you work. This allows to trace people against our will.

Under this kind of situation, researches on the privacy-preserving sensitive data (not only image feature detection) over the encrypted domain become necessary, such as image search (Jégou and Zisserman 2014; Grauman and Fergus 2013), and anonymous authentication and keyword search (Wang et al. 2014, 2013; Tang and Liu 2015; Li et al. 2007).

Usually, to achieve privacy preserving, clients operate homomorphic encryption techniques, such as Paillier (1999), Damgård et al. (2007), Boneh et al. (2005) and Bresson et al. (2003) on the sensitive data before outsourcing. Here, researchers outsource the encrypted sensitive data to the cloud for its abundant computing resources and benefits, since the computational ability of mobile devices is constrained and limited.

Some works on privacy-preserving feature extraction over the encrypted domain were proposed in (Hsu et al. 2012; Qin et al. 2014; Zhou et al. 2015). In (Hsu et al. 2012), the authors realized privacy-preserving SIFT over the encrypted domain. However, they used Paillier cryptosystem, in which the decryption mechanism for the data owner is unpractical, since the computational complexity on the encrypted image data of the mobile devices is resource constrained. In (Zhou et al. 2015), the authors used full homomorphic encryption as the technique of privacy preserving and realized a CCA2 security for cloud-assisted health-care system. The security of their model is improved; however, the running time of the model is also increased accordingly.

In this paper, we consider the following scenario to realize a secure and efficient privacy-preserving outsourced image feature extraction scheme over the encrypted domain.

Note that for usual image recognition, the image always has a explicit, clear and positive pattern. However, for may practical problems, the image itself has a kind of fuzziness, such as license plate in intelligent transportation system and remote sensing images. In this case, we need to use fuzzy image processing. Anyway, this is another research theme, and we will not consider this situation here. Thus, we still assume that our image is clear and explicit. For the secret image data, we use BCP (Bresson et al. 2003) as our encryption scheme, which is an additively homomorphic scheme with double decryption mechanisms. (That is, such scheme has two independent, additively homomorphic decryption mechanisms). The master key is stored on a third party (T) who can decrypt any given ciphertext without the consent of a client. To protect the privacy of the client's data, we require to restrain the unlimited ability of the third party. Hence, in our scheme, the third party is only responsible for generating and issuing public parameter for the other clients. Keeping this in mind, our basic construction consists of three steps:

1. A client P has his own public and secret keys and sends his encrypted data and public key to a company S, respectively.
2. After receiving the encrypted input, the company S generates a DoG image (over the encrypted domain).
3. Client P and company S together run the Extreme protocol to obtain the feature key point.
4. Once the company S obtains the feature key point in the encrypted domain, he needs to compute the feature key point descriptor in the encrypted domain. Next, he executes the descriptor and matching.

The remainder of this paper is organized as follows. In Sect. 2, we introduce some preliminaries. In the next section, we construct our scheme. In Sect. 4, we formally discuss the privacy-preserving SIFT problem in the encrypted domain. In Sect. 5, we discuss multiparty privacy-preserving SIFT in the encryption domain. In Sect. 6, we analyze the security and complexity of our scheme. Finally, conclusions and future work are given in Sect. 7.

2 Preliminaries

2.1 Notations and security model

Throughout this paper, we use the following basic knowledge: let $\mathbb{Z}$, $\mathbb{R}$ and $\mathbb{N}$ be integer, real and natural number set, respectively. We denote some finite set by D. Then, $d \leftarrow \mathsf{D}$ is used to denote the fact that d is chosen randomly from

the uniform distribution over D. For some algorithm A, we say $y \leftarrow \mathsf{A}(x)$; this simply means that y is the output of the algorithm A with a fixed input $x \leftarrow \mathsf{D}$.

Assume that two-dimensional functions of the form $f(x, y)$ denote a discrete image. From a physical point of view, the value (or amplitude) of f at spatial position (x, y) is a positive scalar quantity determined by the source of the image, which means that the image $f(x, y)$ can be defined as:

$$f(x,y) = \begin{pmatrix} f(0,0) & \dots & f(0,N-1) \\ f(1,0) & \dots & f(1,N-1) \\ \vdots & \ddots & \vdots \\ f(M-1,0) & \dots & f(M-1,N-1) \end{pmatrix} .. \quad (1)$$

Here, the pair $(x, y) = (i, j)$ does not stand for any actual values of physical coordinates when the image was sampled, but we mention that the $j-$th samples along the $i-$th row. The value $f(i, j)$ at coordinates (i, j) is called a pixel. For the sake of simplicity, the image $f(x, y)$ will be denoted as

$$f(x,y) = (f(x = i, y = j))_{M\times N} = (f(i,j))_{M\times N}. \quad (2)$$

2.2 Cryptographic techniques

2.2.1 Additively homomorphic encryption

We say a public key cryptosystem $E = (KeyGen, Enc_{PK}, Dec_{SK})$ is additively homomorphic, if there exists an operation on ciphertexts $Enc_{PK}(x)$ and $Enc_{PK}(y)$ such that the result of that operation corresponds to a new ciphertext whose decryption yields the sum of the plaintext $x + y$, i.e.,

$$Dec_{SK}(Enc_{PK}(x) + Enc_{PK}(y)) = x + y.$$

For this property, the multiplication of $Enc_{PK}(x)$ with a constant λ can be computed as follows,

$$Dec_{SK}(Enc_{PK}(x)^{\lambda}) = Dec_{SK}(\overbrace{Enc_{PK}(x) \cdot \dots \cdot Enc_{PK}(x)}^{\lambda \ terms}) = \overbrace{x + x \dots + x}^{\lambda \ terms} = x \cdot \lambda.$$

As instantiation, we use the BCP cryptosystem by Bresson, Catalano and Pointcheval (2003), which is an additively homomorphic scheme with two independent decryption algorithms. The master decryption algorithm has a master secret key and the client decryption algorithm has a client key. Once a certain master key is known, the master decryption algorithm can decrypt a given ciphertext. We require the security of such homomorphic scheme with double decryption algorithm to be IND-CPA (also semantic security), which means that the adversary should not be able to distinguish the encryption of two arbitrary messages.

2.2.2 BCP encryption scheme

The BCP encryption scheme consists of three stages as follows:

$(\mathsf{pp}, \mathsf{mk}) \longleftarrow \mathsf{Setup}(\kappa)$: Assume that p, q, p', q' are distinct odd primes with $p = 2p' + 1$ and $q = 2q' + 1$. Given a safe parameter κ, let $n = pq$ with bit length κ. For the group $\mathbb{Z}^*_{n^2}$, let $\mathbb{G}$ be a cyclic group of quadratic residues modulo n^2, and we have ord($\mathbb{G}$)=$pp'qq'$. Actually, there are exactly n elements of order n in $\mathbb{Z}^*_{n^2}$, and all of them in the form $1 + kn(k \in [0, n - 1])$; the maximal order of element in $\mathbb{G}$ is $pp'qq'$. Choose a random element $\alpha \in \mathbb{Z}^*_{n^2}$ and set $g = \alpha^2 \bmod n^2$. Then the outputs of this algorithm contain *public parameters* $\mathsf{pp} = (n, k, g)$ and the *master secret key* $\mathsf{mk} = (p', q')$.

$(\mathsf{pk}, \mathsf{sk}) \longleftarrow \mathsf{KeyGen}(\mathsf{pp})$: choose a random $a \in [1, \text{ord}(\mathbb{G})]$ and set $h = g^a \bmod n^2$, and the algorithm's outputs $\mathsf{pk} = h$ and $\mathsf{sk} = a$ as the client of public and secret key, respectively.

$(A, B) \longleftarrow \mathsf{Enc}_{(\mathsf{pp},\mathsf{pk})}(m)$: given a message $m \in \mathbb{Z}_n$, choose a random $r \in \mathbb{Z}_{n^2}$, the output the ciphertext

$$A = g^r \bmod n^2, \qquad B = h^r(1 + mn) \bmod n^2. \quad (3)$$

The client decryption algorithm $\mathsf{Dec}_{(\mathsf{pp},\mathsf{sk})}$ can be described as

$m \longleftarrow \mathsf{Dec}_{(\mathsf{pp},\mathsf{sk})}(A, B)$: given a ciphertext (A, B) and a secret key $\mathsf{sk} = a$, compute the m as

$$m = \frac{\frac{B}{A^a} - 1 \bmod n^2}{n}. \quad (4)$$

The master decryption algorithm $\mathsf{mDec}_{(\mathsf{pp},\mathsf{mk})}$ is as follows:

$m \longleftarrow \mathsf{mDec}_{(\mathsf{pp},\mathsf{pk},\mathsf{mk})}(A, B)$: given a ciphertext (A, B), a client's public key $\mathsf{pk} = h$ and the master secret key mk, we can compute the client's secret key $\mathsf{sk} = a$ which corresponds to $\mathsf{pk} = h$ as follows:

$$a \bmod n = \frac{h^{p'q'} - 1 \text{mod} n^2}{n} \cdot k^{-1} \bmod n. \quad (5)$$

Since we use public key $\mathsf{pk} = h$ and a random $r \in \mathbb{Z}_{n^2}$ to encrypt plaintext $m \in \mathbb{Z}_n$, for the decryption algorithm, it is necessary to compute

$$r \bmod n = \frac{A^{p'q'} - 1 \bmod n^2}{n} \cdot k^{-1} \bmod n. \quad (6)$$

According to $a \bmod n$ and $r \bmod n$, we can compute $\tau = ar \bmod n$. So the algorithm outputs

$$m = \frac{(\frac{B}{g^{\tau}})^{p'q'} - 1 \bmod n^2}{n} \cdot \pi \bmod n, \tag{7}$$

where k^{-1} denotes the inverse of $k \bmod n$ and $\pi = (p'q')^{-1} \bmod n$. We use $[\cdot]$ to denote the encryption of message m, that is

$$\begin{aligned}[m] &= \mathsf{Enc}_{(\mathsf{pp},\mathsf{pk})}(m, r) \\ &= (g^r \bmod n^2, h^r(1+mn) \bmod n^2) \\ &= (A, B).\end{aligned} \tag{8}$$

ADD denotes additive gates securely, based on "Blinding" techniques in our construction. That is, given $[m] = (A, B)$ and $[m'] = (A', B')$, $\mathsf{ADD}([m], [m']) = (A \cdot A' \bmod n^2, B \cdot B' \bmod n^2) = [m + m']$. For an integer $a \in \mathbb{Z}$, $(A, B)^a = (A^a \bmod n^2, B^a \bmod n^2)$.

2.2.3 Secure multiplications

The blinding can carry out the secure multiplications between two encrypted data. Assume that Bob has ciphertext $[x]$ and $[y]$, which were encrypted by Alice's public key. If Bob wants to obtain $[xy]$, he needs an interactive protocol with Alice. Firstly, he picks up randomly values r_x, r_y as homomorphic blinding factors of $[x]$, $[y]$, respectively. More precisely,

$$[z_x] = [x + r_x] = [x][r_x],$$

and

$$[z_y] = [y + r_y] = [y][r_y],$$

then Bob sends $[z_x]$ and $[z_y]$ to Alice. Since Alice has a secret key, she can decrypt $[z_x]$, $[z_y]$, compute and re-encrypt the multiplier $z_x z_y$ and then sends the final ciphertext to Bob. After receiving the value $[z_x z_y]$, Bob is ready to finish the following computation:

$$\begin{aligned}[xy][xy] &= [z_x z_y - (xr_x + yr_y + r_x r_y)] \\ &= [z_x z_y][x]^{-r_y}[y]^{-r_x}[r_x r_y]^{-1}.\end{aligned} \tag{9}$$

Here, the random values r_x, r_y are required to be uniformly distributed over $\mathbb{Z}_n$, and the blinding values $x + r_x$ and $y + r_y$ are not allowed to leak any information to Alice. In particular, if $x = y$, Alice computes $[x^2] = [z_x^2][x]^{-2r_x}[r_x^2]^{-1}$.

2.2.4 Extreme protocol

One of the most famous protocols for securely comparing two private data between two computationally bounded parties is based on Yao's garble circuit (Yao 1982, 1986; Kolesnikov et al. 2009). However, in (Damgård et al. 2007, 2009), the authors used homomorphic encryption and SMC by comparison.

Inspired by their ideas, we also use SMC to construct our Extreme protocol.

Initially, Company S has access to both $[a]$ and $[b]$, and computes $[x] = [2^l + a - b] = [2^l][a][b]^{-1}$. Since $0 \le a, b \le 2^l$, $l \in \mathbb{Z}$, x is a positive $(l+1)$-bit value. Assuming x_l is the most important bit of x, there is

$$x_l = 0 \Longleftrightarrow a < b.$$

- If S has $[x \bmod 2^l]$, then

  $$[x^l] = ([x][x \bmod 2^l]^{-1})^{2^{-1}}.$$

  Once S has $[x_l] = [a < b]$, using $m = (a < b)(a-b) + b$, the encryption of the minimum m is easily obtained.
- If S does not know $[x \bmod 2^l]$, then he needs to compute the $[x \bmod 2^l]$. To obtain this value, S takes a protocol with cline P.
  S chooses uniformly random value r as an additively blind factor of x and computes

  $$[y] = [x][r] = [x + r], \quad r \bmod 2^l,$$

  then $[y]$ is also a uniformly random value, company S can safely send it to client P, who will learn non-useful information after decryption. Client P then computes $d \bmod 2^l$, and returns $[y \bmod 2^l]$ back to S.

Now, S removes the initially add random value r as follows:

$$[x'] = [y \bmod 2^l][r \bmod 2^l]^{-1},$$

since $x' \bmod 2^l = (y \bmod 2^l - r \bmod 2^l) \bmod 2^l$. We remark that if $y \bmod 2^l > r \bmod 2^l$, x' is a result value; and if $y \bmod 2^l < r \bmod 2^l$, then an underflow has occurred. So, consider a parameter $s = (y \bmod 2^l < r \bmod 2^l)$, that is $s \in \{0, 1\}$, if S has $[s]$, then

$$[x \bmod 2^l] = [x'][s]^{2^l} = [x' + s^{2^l}].$$

As long as S knows $[s]$, the $[x \bmod 2^l]$ is obtained. This question is transformed into the comparison of two private inputs: $y' = y \bmod 2^l$ (held by S) and $r' = r \bmod 2^l$ (held by P). This problem is known as Yao's millionaire problem (Yao 1982). Hence, we use Yao's millionaire problem as a subprotocol, where the details can be described in (Tuyls et al. 2005; Blake and Kolesnikov 2006; Naor et al. 1999).

2.3 Security model

Our scheme is composed of three major entities: a company S (or server or cloud), a third party (T) and the clients. We assume T is trusted by all the clients, its only task is to generate a public system parameter of the encryption scheme which will be issued to the clients, and owning a master key which can decrypt any given ciphertext. Meanwhile, we require that T will not interact with the company S other than the clients and will not participate in any computation. The existence of S is reasonable; it can play the role of supervision, when some malicious entity interrupts the system, forges and makes bogus sensitive data intentionally. Government agents, medical organization or Public Security Bureau (PSB) can be the T in real life; it is important and necessary for our socirty.

From a theoretical point of view, executing protocols in semi-honest model (Goldreich 2004) is necessary, which means that S, T and the client P can be honest and follow the protocols, but try to gather or discover information about the (intermediate) results of the computation just by viewing or looking at the protocol's transcripts as much as possible. In addition, we assume that there is no collusion among any of the clients.

3 Our construction

In our scheme, the clients provide an encrypted image to the company for some service (i.e., face recognition), and the semi-honest company S is required to provide the corresponding service. We stress that T does not collude with cloud S and the client P.

An original SIFT key point is a pixel which possesses a local extremum in the scale space defined by difference-of-Gaussian (DoG) filters. In the scale space, if some one maliciously generates another extremum near the true one, there can be several equal extrema (eight neighbors of the true one) in a detection region get away with key point detection. On the other hand, to ensure the security of the data of the client, we consider a secret information transformation process.

Equipped with these preliminary points, we encrypt the secret data before operating the SIFT algorithm; this makes the output feature of the algorithm not dominant.

3.1 Enrollment

Third party T runs the algorithm Setup of the BCP encryption scheme, sets up the system parameters and sends public parameters $\mathsf{pp} = (n, k, g)$ to the client P. The master secret key $\mathsf{mk} = (p', q')$ is stored in the third party T.

3.2 Data upload

After receiving the public parameters $\mathsf{pp} = (n, k, g)$, client P generates its own public and secret keys: $\mathsf{pk} = h$ and $\mathsf{sk} = a$ under the received public parameters $\mathsf{pp} = (n, k, g)$ by the algorithm KeyGen of the cryptosystem. Using Enc, client P encrypts its private message, and then uploads encrypted data and pk to the company S.

3.3 Interaction between the company and client

After receiving the encrypted data from the client, company S wants to compute a privacy-preserving SIFT algorithm, including DoG transform, key point extraction, feature descriptor extraction and descriptor matching. When the client P gets the result of the encrypt outputs, he can decrypt and obtain the plaintext result.

It should be pointed out that in these protocols, we have used the secure multiparty computation (SMC) based on garbled circuit to solve the Extrema protocol, which has been shown to be more efficient than only homographic comparison in Sadeghi et al. 2010.

4 Privacy-preserving SIFT in the encrypted domain

Operating on plaintext space can be performed on correspondingly ciphertext space without disclosing the plaintext, satisfying that this operation is homomorphic encryption. Thus, privacy-preserving SIFT can be realized.

4.1 Scale space in encrypted domain

The first step of the SIFT algorithm is to construct scale space, while the element of this space is an image $f(x, y)$ convolved with Gaussian function $G(x, y, \sigma)$:

$$\begin{aligned} L(x, y, \sigma) &= G(x, y, \sigma) * f(x, y) \\ &= \sum_{u,v} G(u, v, \sigma) f(x - u, y - v), \end{aligned} \tag{10}$$

where $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$, σ notes variance and the scale size. If the size is bigger, the Guassian-blurred image is coarser, with more contrast and finer.

To detect stable key point locations efficiently in scale space, we need to take difference-of-Gaussians function convolved with the image, and such difference-of-Gaussians compose a new space, defined as DoG space. The element of DoG space can be computed like this:

$$D\_f(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * f(x, y) = L(x, y, k\sigma) - L(x, y, \sigma), \quad (11)$$

which means the difference of nearby scales convolved with an image $f(x, y)$.

Here, for practical implementation, the BCP cryptosystem scheme can only operate in integer domain, so we take an integer DoG function which is assigned different scales σ_i and σ_j:

$$\begin{aligned} L(x, y, \sigma) &= \lceil sG(x, y, \sigma) \rfloor \\ G_D(u, v, \sigma_{ij}) &= \lceil s(G(x, y, \sigma_i) - G(x, y, \sigma_j)) \rfloor, \\ &\quad \forall\, u \text{ and } v. \end{aligned} \quad (12)$$

Since the Gaussian function $G(\cdot)$ is smaller than 1, s as a scaling factor is used to enlarge $G(\cdot)$ such that it is an integer.

Client P provides an encrypted face image

$$\begin{aligned} \mathsf{Enc}_{(\mathsf{pp},\mathsf{pk})}(f(x, y), r_{x,y}) &= [f(x, y)] \\ &= ([f(i, j)])_{M\times N} \\ &= ((A_{ij}, B_{ij}))_{M\times N} \end{aligned} \quad (13)$$

to the company S for recognition. After receiving an encrypted image $[f(x, y)]$, S convolves it with the DoG function, so the encrypted image in the DoG space can be expressed as

$$\begin{aligned} & D\_[f(x, y)] \\ &= \prod_{u,v} ((A_{i-u,j-u}, B_{i-v,j-v})^{G_D(u,v,\sigma)})_{M\times N} \\ &= \prod_{u,v} \mathsf{Enc}_{(\mathsf{pp},\mathsf{pk})}(f(x-u, y-v), r_{x-u,y-v})^{G_D(u,v,\sigma)} \\ &= \mathsf{Enc}_{(\mathsf{pp},\mathsf{pk})}\left(\sum_{u,v} G_D(u, v, \sigma) f(x-u, y-v), r_{x,y}\right) \\ &= \mathsf{Enc}_{(\mathsf{pp},\mathsf{pk})}(G_D(x, y) * f(x, y), r_{x,y}) \\ &= \mathsf{Enc}_{(\mathsf{pp},\mathsf{pk})}(L(x, y, k\sigma) - L(x, y, \sigma)), \end{aligned} \quad (14)$$

where $r_{x,y}$ is chosen uniformly random and depends upon the location of a pixel on the DoG image.

According to the homomorphic property, the above equation can be recast as:

$$\begin{aligned} & D\_[f(x, y, \sigma)] \\ &= \prod_{u,v} \mathsf{Enc}_{(\mathsf{pp},\mathsf{pk})}(f(x-u, y-v), r_{x-u,y-v})^{G_D(u,v,\sigma)} \\ &= \prod_{u,v} (g^{r_{x-u,y-v}} \bmod n^2, h^{r_{x-u,y-v}}(1 + f(x-u, y-v)n) \\ &\quad \bmod n^2)^{G_D(u,v,\sigma)} \\ &= (g^{r_\sigma} \bmod n^2, h^{r_\sigma}(1 + G_D(x, y, \sigma) * f(x, y)n) \\ &\quad \bmod n^2) \\ &= \mathsf{Enc}_{(\mathsf{pp},\mathsf{pk})}(D\_f(x, y, \sigma), r_\sigma) \\ &= [D\_f(x, y, \sigma)], \end{aligned} \quad (15)$$

where $r_\sigma = \sum_{u,v} r_{x-u,y-v} G_D(u, v, \sigma)$ is used to encrypt a pixel $G_D(x, y, \sigma) * f(x, y)$.

Above, Eqs. (14)–(15) indicate that the client P sends the encrypted image $[f(x, y)]$ to the company S for operating difference-of-Gaussian function in ciphertext space identical to encrypting the difference-of-Gaussian image $D\_f(x, y, \sigma)$ under r_σ with the scale parameter σ.

Next, for brevity, we use $(\bar{A}_{ij}, \bar{B}_{ij})$ to indicate the DoG image pixel at position (i, j), i.e.,

$$D\_[f(x, y, \sigma)] = ((\bar{A}_{ij}, \bar{B}_{ij}))_{M\times N}, \quad (16)$$

where $\bar{A}_{ij} = \prod_{u,v} A_{i-u,j-u}^{G_D(u,v,\sigma)} \bmod n^2$, and $\bar{B}_{ij} = \prod_{u,v} B_{i-v,j-v}^{G_D(u,v,\sigma)} \bmod n^2$. Figure 1 shows that the original image $'girl'$a and $'Girl'$b and its encrypted image c and d, respectively.

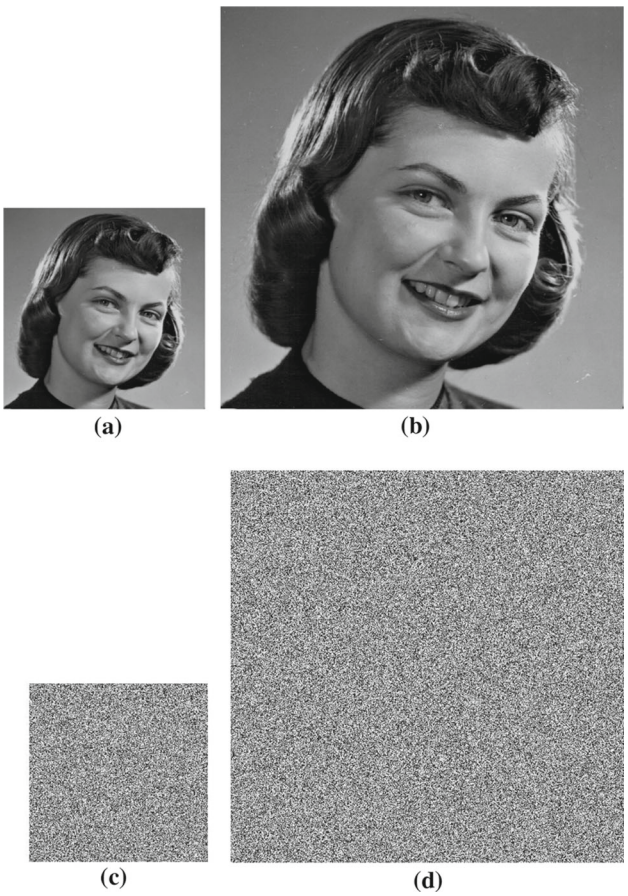

Fig. 1 **a** 256-bit image; **b** 512-bit image; **c**, **d** is BCP encrypted image of **a** and **b**, respectively

4.2 Local extrema detection via comparison of encrypted data

Once the company S obtains the DoG images, it wants to detect the local extrema of DoG images as a candidate key point. That is, each sample point in the DoG images is compared to 8 neighbors in current scale as well as 19 neighbors in adjacent scales, and only a pixel value as an extrema of all neighbors is selected as a candidate key point. Homomorphic encryption scheme guarantees that comparison between encrypted data is equivalent to obtaining the local extrema value in the original space (plaintext space). Due to the computational difficulty, company S needs to interact with client P to detect the extrema value using the Extreme protocol.

4.3 Feature descriptor generation

In this step, the detected key point is assigned orientations based on local image gradient directions. So, the key point descriptor can be represented related to this orientation. First, we describe an SIFT feature descriptor over the plaintext domain.

An SIFT feature descriptor is made for the actual 16×16 region, which is further broken down into 16 4×4 blocks, around a center: feature point. For every detected feature point, the feature descriptor is accomplished at the corresponding scale. Then for these 4×4 blocks, we compute the gradient value and orientation for each position (x, y) as follows,

$$\begin{aligned} m(x, y) &= \sqrt{L_x^2 + L_y^2}, \\ \theta(x, y) &= \arctan \frac{L_y}{L_x}, \end{aligned} \tag{17}$$

where L_x and L_y can be difference approximations as $L(x+1, y, \sigma) - L(x-1, y, \sigma)$ and $L(x, y+1, \sigma) - L(x, y-1, \sigma)$, respectively. Once Eq. (17) is obtained, a number of restrictive directions can be defined by the histogram of weighted magnitudes.

Due to high computational complexity and impracticality in the encrypted domain, we use a modify descriptor to generate the weighted magnitudes located at four directions (0°, 45°, 90°, and 135°) in this paper, which constitutes a new four-dimensional vector. So, to calculate the feature descriptor in the encrypted domain, company S does the following steps: for each 4×4 block, let $V(k), k = 0, 1, 2, 3$ denote a four-dimensional feature descriptor and $\mathsf{Enc}(V(k), r_k) = [V(k)]$:

$$\begin{aligned} [V(0)] &= [V(0)][\mathrm{Diff}_{0^\circ}] \\ &= [V(0)][L(x+1, y, \sigma)][L(x-1, y, \sigma)]^{-1}, \\ &\quad \text{if } L(x+1, y, \sigma) \geq L(x-1, y, \sigma); \\ [V(1)] &= [V(1)][\mathrm{Diff}_{90^\circ}] \\ &= [V(1)][L(x, y+1, \sigma)][L(x, y-1, \sigma)]^{-1}, \\ &\quad \text{if } L(x, y+1, \sigma) \geq L(x, y-1, \sigma); \\ [V(2)] &= [V(2)][\mathrm{Diff}_{45^\circ}] \\ &= [V(2)][L(x-1, y-1, \sigma)][L(x+1, y+1, \sigma)]^{-1}, \\ &\quad \text{if } L(x-1, y-1, \sigma) \geq L(x+1, y+1, \sigma); \\ [V(3)] &= [V(3)][\mathrm{Diff}_{135^\circ}] \\ &= [V(3)][L(x+1, y-1, \sigma)][L(x-1, y+1, \sigma)]^{-1}, \\ &\quad \text{if } L(x+1, y-1, \sigma) \geq L(x-1, y+1, \sigma), \end{aligned} \tag{18}$$

where $[V(k)]$ are all initialized to be 1 and need Extreme protocol for comparison in the above equations. In Fig. 2, images e and f are the feature extractions of the original images a and b by SIFT, respectively. Images g and f indicate the feature extractions corresponding to the encrypted images c and d by piracy-preserving SIFT.

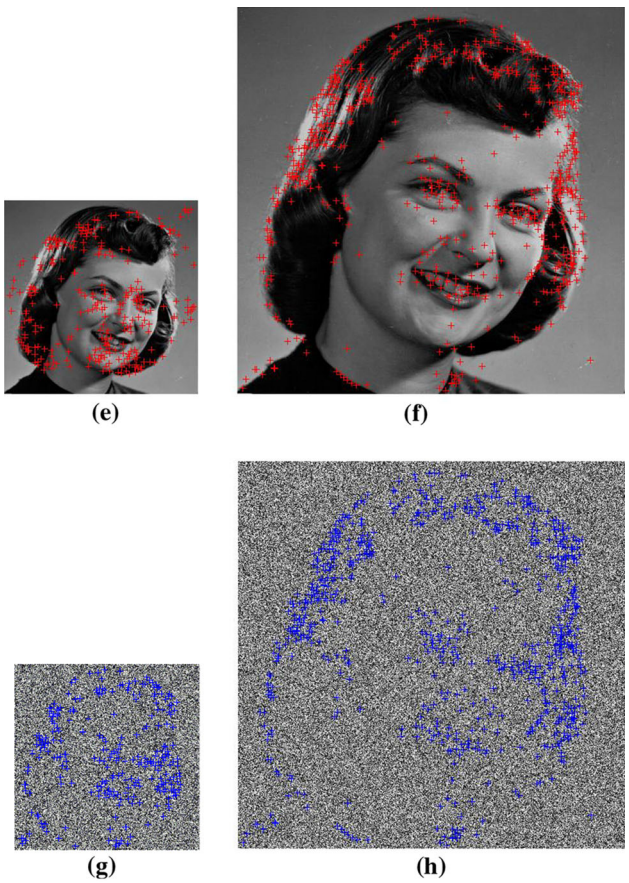

Fig. 2 **e**, **f** feature extraction in plaintext domain; **g**, **h** feature extraction in encrypted domain

4.4 Feature descriptor matching

As shown in the previous section, each descriptor has a split 64-dimensional vector V_d and is normalized to unit length. Once the company S obtains the descriptors for each encrypted image, it needs to compare the descriptors of two encrypted images. An original match of SIFT is accepted only if its distance is less than *disRatio* (a threshold value) times the distance to the second closest match. However, in the ciphertext space, we need to decide a matching strategy. The simplest way to find all corresponding feature descriptors is to compare all feature descriptors against all other feature descriptors in each pair of potentially matching images. Unfortunately, this is quadratic in the number of extracted feature descriptors, which means it is impractical for most applications.

For efficiency (Hsu et al. 2012), some researchers used similar metric for evaluation, which is to compute dot products between unit vectors:

$$Sim(V_i, V_j) = \sum_{k=0}^{63} V_i(k)V_j(k), \tag{19}$$

where V_i and V_j are descriptors. Indeed, in plaintext space, using similarity metric, Eq. (19) is really effective. However, it is not effective for match in ciphtertext space, since in the cipertext space Eq. (19) is expressed as

$$Sim([V_i], [V_j]) = \prod_{k=0}^{63} [V_i(k)]^{V_j(k)} \bmod n^2. \tag{20}$$

To calculate Eq. (20) in ciphertext space, the company S has to know the descriptor V_j first. This means that the privacy of the client P cannot be protected. Likewise, if S using Euclidean metric in the ciphertext space for similarity evaluation,

$$\begin{aligned} &Sim^2([V_i], [V_j]) \\ &= \left[\sum_{k=0}^{63} (V_i(k) - V_j(k))^2\right] \\ &= \prod_{k=0}^{63} [V_i(k)^2][V_i(k)V_j(k)]^{-2}[(V_j(k)^2] \bmod n^2, \end{aligned} \tag{21}$$

S only knows the encrypted descriptors $[V_i]$ and $[V_j]$; to achieve the desired goal, he needs to obtain $[V_i(k)^2]$, $[V_i(k)V_j(k)]^{-2}$ and $[(V_j(k)^2]$ via SMC with client P. However, this process costs large time.

To reduce the computational cost, we take the l_1 norm as our matching metric. In the plaintext space, the l_1 norm is defined as

$$Sim(V_i, V_j) = |V_i - V_j|_{l_1} = \sum_{k=0}^{63} |V_i(k) - V_j(k)|. \tag{22}$$

Then, in the ciphertext space, because of the homomorphic property, the above equation can de denoted as:

$$\begin{aligned} &Sim([V_i], [V_j]) \\ &= |[V_i] - [V_j]|_{l_1} \\ &= \sum_{k=0}^{63} |[V_i(k)] - [V_j(k)]| \\ &= \prod_{k\in\{t|V_i(t)>V_j(t);0\le t\le 63\}} [V_i(k)][V_j(k)]^{-1} \\ &\quad \times \prod_{k\in\{t|V_i(t)\leqslant V_j(t);0\le t\le 63\}} [V_i(k)][V_j(k)]^{-1} \bmod n^2. \end{aligned} \tag{23}$$

Equation (23) the computation of $|[V_i] - [V_j]|$ is dependent on our proposed Extreme protocol. Therefore, the matching can be realized.

5 Multiparty for privacy-preserving SIFT in the encrypted domain

In the previous section, we discussed privacy-preserving SIFT for one client P; however, there are more general cases where two or more people want to recognize encrypted images under different public keys. Usually, the SMC only deals with the data under the same public key; so, before using SMC, we need to transform these encrypted images (with different public keys) into the encrypted data with a single public key. If we use secret sharing scheme (Shamir 1979; Blakley et al. 1979; De Santis et al. 1994; Blundo et al. 1997, 1994, 1993) for multiparty case, all parties need to interact with each other and should be online since the decryption algorithm needs the secret key of the parties.

In this section, we will consider how to build a system with multiparty. In this case, our scheme needs to be modified as follows: the third party T sets up the BCP encryption scheme, generates the system parameters, holds the master key $\mathsf{mk} = (p', q')$ and distributes public parameters $\mathsf{pp} = (n, k, g)$ to the company S. Assume that T only interacts with the company S and the client $\mathsf{P}_i (1 \le i \le N)$ only interacts with S. After clients receive the public parameters from the company S, they can use their own key generation algorithm to generate their own public and secret keys, respectively, and to upload encryption of their secret data and public key to S. Then, S and T execute some cryptographic protocols, to transform all ciphertexts encrypted by

those clients' individual keys to other new kind of ciphertexts encrypted by the same public key. Using these new ciphertexts, S and T can operate the SMC protocol to compute any polynomial function. After the result (also encrypted under the same public key) is given, S needs to run a final SMC protocol with T so as to transform this result back into ciphertexts that are encrypted by the clients' individual public keys. During this process, the clients have no interaction with each other. We stress that every party in this model is semi-honest and the company S and T do not collude with each other.

6 Security and complexity analysis

6.1 Security analysis

Informally speaking, the Diffie–Hellman problem has two variants, a computational version and a decision version. For a computational Diffie–Hellman problem, given a multiplicative group $(\mathbb{G}, \cdot)$, an element $g \in \mathbb{G}$ having order $\mathrm{ord}(\mathbb{G})$, and two elements $a, b \in [1, \mathrm{ord}(\mathbb{G})]$, it is hard to find g^{ab}, when g^a and g^b, g and n are given.

6.1.1 Decision Diffie–Helleman (DDH) problem over $\mathbb{Z}^*_{n^2}$

We say DDH problem is hard if for all probabilistic, polynomial-time algorithms $\mathfrak{A}$, there exists a negligible function $\mathsf{negl}()$ such that for large enough l,

$$\begin{aligned}&|\mathrm{P}[\mathfrak{A}(\mathbb{G}, n, g^x \bmod n^2, g^y \bmod n^2, g^{xy} \bmod n^2) = 1]\\ &- \mathrm{P}[\mathfrak{A}(\mathbb{G}, n, g^x \bmod n^2, g^y \bmod n^2, g^z \bmod n^2) = 1]|\\ \leq &\mathsf{negl}(l),\end{aligned}$$

where $x,\ y,\ z \in \mathbb{Z}^*_{n^2}$ are randomly chosen. Note that when z is chosen at random from $\mathbb{Z}^*_{n^2}$, independent of anything else, the element $g^z \bmod n^2$ is uniformly distributed in $\mathbb{G}$.

This clearly shows that the ensemble of tuples of the type $(\mathbb{G}, n, g^x \bmod n^2, g^y \bmod n^2, g^{xy} \bmod n^2)$ is computationally indistinguishable from the ensemble of tuples of the type $(\mathbb{G}, n, g^x \bmod n^2, g^y \bmod n^2, g^z \bmod n^2)$.

According to (Bresson et al. 2003), the authors defined some concepts, as follows.

6.1.2 Lift Diffie-Hellman problem

The Lift Diffie–Hellman computational problem is said to be hard if, for any probabilistic polynomial time algorithm $\mathfrak{A}$, one negligible function $\mathsf{negl}()$ exists such that for large enough l,

$$\begin{aligned}\mathrm{P}[\mathfrak{A}(n, X, Y, Z \bmod n) = Z(\bmod n^2)|p, q \leftarrow \mathsf{SP}(l/2);\\ n = pq;\ g \leftarrow \mathbb{G};\ x, y, z \leftarrow [1, \mathrm{ord}(\mathbb{G})];\\ X = g^x \bmod n^2;\ Y = g^y \bmod n^2;\\ Z = g^{xy} \bmod n^2;\ ] = \mathsf{negl}(l),\end{aligned}$$

where SP denotes the sets of safe prime numbers of length l.

6.1.3 Partial discrete logarithm over $\mathbb{Z}^*_{n^2}$

For all probabilistic polynomial time algorithm $\mathfrak{A}$, a negligible function $\mathsf{negl}()$ exists such that for large enough l,

$$\begin{aligned}&\mathrm{P}[\mathfrak{A}(n, g, h) = a \bmod n)|p, q \leftarrow \mathsf{SP}(l/2); n = pq;\\ &g \leftarrow \mathbb{G};\ a \leftarrow [1, \mathrm{ord}(\mathbb{G})];\\ &h = g^a \bmod n^2;\ ] = \mathsf{negl}(l).\end{aligned}$$

The relationship between the Lift Diffie–Hellman problem and the Partial Discrete Logarithm problem can be described by the following theorem:

Theorem (see Bresson 2003, Theorem 10). If the Partial Discrete Logarithm problem is hard, then so is the Lift Diffie-Hellman problem.

Theorem (One-wayness) (see Bresson (2003), Theorem 9). The BCP encryption scheme is one way if and only if the Lift Diffie–Hellman problem is hard.

Proof We use converse-negative proposition to prove the theorem. Necessarily, we assume that if one can efficiently solve the LDH problem, namely given $g, X = A = g^r \bmod n^2, Y = g^x \bmod n^2$ and $t = B \bmod n = pk^r(1 + mn) \bmod n^2 = pk^r \bmod n^2$, one can calculate $Z = pk^r \bmod n^2$, where $(A,\ B) = (g^r \bmod n^2,\ pk^r(1 + mn) \bmod n^2)$ is an encryption of message m. Once one calculates $Z = pk^r \bmod n^2$, then he can calculate the inverse of Z, namely Z^{-1}, finally he recovers the message: $m = \frac{Z^{-1}B-1}{n} \bmod n^2$.

Sufficiently, we assume that BCP is not one-way, namely the message m can be recovered from a correctly encrypted data $(A = X,\ B = Z(1 + mn) \bmod n^2)$, indeed, B can be expressed by the formula $B = Z(1+mn) = Z+Zmn = Z+(Z \bmod n)mn = Z + tmn \bmod n^2$, where $g, t = Z \bmod n$ are given. For a randomly chosen message m', we use $(A = X,\ B = t(1 + m'n) \bmod n^2)$ as its ciphertext, then $Z + tmn = t + tm'n \bmod n^2$, we can calculate $Z = t(1 - (m' - m)n) \bmod n^2$.

Now, we describe the definition of the semantic security.

Semantic Security, IND-CPA. Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme, and let $\mathfrak{A} = (\mathfrak{A}_1, \mathfrak{A}_2)$ be an adversary. We say that Π is secure in the sense of IND-CPA, if

$$\mathrm{Avd}^{ind-cpa}_{\mathfrak{A},\Pi}(k) = 2 \cdot \mathrm{P}[(pk, sk) \leftarrow (\mathsf{KetGen}(1^k)); (m_0, m_1, s) \leftarrow \mathfrak{A}_1(pk); b \leftarrow \{0, 1\}; y \leftarrow \mathsf{Enc}_{pk}(m_b); \mathfrak{A}_2(m_0, m_1, s, y) = b] - 1$$

is negligible for $k \in \mathbb{N}$. For the first adversary $\mathfrak{A}_1$, just like a message generator, given the public key pk, he can generate a message space and from that sample two messages m_0, m_1, then output a triple (m_0, m_1, s), where s is a secret state information, possibly including pk. We require that the length of the above two messages is the same, i.e., $|m_0| = |m_1|$. For the second adversary $\mathfrak{A}_2$, after receiving an encryption y of a random message m_b and the previously saved state s, he outputs a bit b. Thus, the semantic security shows that whatever is efficiently computable about the plaintext, given the ciphertext, is also efficiently computable without the ciphertext. □

Theorem (Semantic Security) (see Bresson (2003), Theorem 11) If Decisional Diffie–Hellman Assumption in $\mathbb{Z}^*_{n^2}$ holds, then the BCP scheme is semantically secure.

Proof First, we assume the BCP encryption scheme is not semantic security, namely there is a polynomial time adversary $\mathfrak{A}$ that can break semantic security, then the Decisional Diffie–Hellman assumption does not hold over $\mathbb{Z}^*_{n^2}$. However, DDH assumption relies on the hardness of computing discrete logarithm over $\mathbb{Z}^*_{n^2}$, so our assumption is a contraction.

Now, the details can be described as follows: given $pk = (n, g, h)$ where $h = g^a$, the adversary $\mathfrak{A}$ chooses distance message m_0, m_1 and sends it to the challenger. Then the challenger tosses a fair coin $b \in \{0, 1\}$, performs the encryption scheme, $c^* = \mathsf{Enc}_{(pk)}(m_b) = (A, B)$ where $A = g^\gamma \bmod n^2$ and $B = g^\beta(1 + m_b n) \bmod n^2$ and sends c^* to the adversary. Upon receipt of c^*, $\mathfrak{A}$ must answer either 0 or 1 as his working out of challenger's coin tossing.

Let $\mathsf{G} = (g, g^a, g^\gamma, g^\beta)$; if G is not a Diffie–Hellman quadruple, in a information theoretic sense, the adversary gains nothing about m_b from $\mathsf{Enc}_{(pk)}$ even he has a polynomially unbounded computing power. If G is a Diffie–Hellman quadruple, then the encryption $\mathsf{Enc}_{(pk)}$ is valid and will give the correct answer with non-negligible probabilistic advantage.

Assume that $\beta = a\gamma + r \bmod \mathrm{ord}(\mathbb{G})$, then

$$\begin{aligned} B &= g^\beta(1 + m_b n) \bmod n^2 \\ &= g^{a\gamma + r}(1 + m_b n) \bmod n^2 \\ &= g^{a\gamma} g^{r_1} g^{r_2 p' q'}(1 + m_b n) \bmod n^2 \\ &= g^{a\gamma + r_1}(1 + n)(1 + m_b n) \bmod n^2 \\ &= g^{a\gamma + r_1}(1 + (r_2 + m_b)n) \bmod n^2. \end{aligned}$$

Since $r \in [1, \mathrm{ord}(\mathbb{G})]$ and r is coprime with $\mathrm{ord}(\mathbb{G})$, we have r and $p'q'$ coprime and r can be written as $r = r_1 + r_2 p' q'$ where $r_1, r_2 \in \mathbb{Z}_n$. We assume that $g^{r_2 p' q'} = (1 + n) \bmod n^2$. In the above equation, r_2 hides m_b and $\mathfrak{A}$ cannot guess b.

Since our privacy-preserving SIFT scheme is based on BCP encryption scheme and SMC, our scheme is semantically secure. □

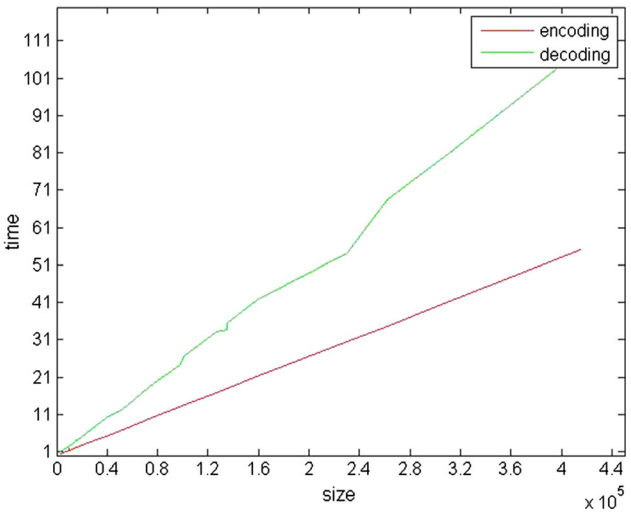

Fig. 3 Encryption and decryption time with different size images

6.2 Complexity analysis

For the experiments, we use *lenovo* computer with a 2.67GHz Intel Core i5 CPU. We choose $p = 59$, $q = 23$, then $n = pq = 1357$, $n^2 = 1841449$. When the image size of ('girl') is 512×512, $\alpha = 915122$, $sk = 78656$, then $g = \alpha^2 \bmod n^2 = 1464460$, $h = g^{sk} \bmod n^2 = 462141$. Its encrypted time is 50 s, and the operation *powermod* takes most of that time. The decryption time is 95 s. The original SIFT in the plaintext domain is 9 s. However, in the encrypted domain, the feature detection time is 55 s. From Fig. 3, we know that the encryption and decryption time is linear (or approximated) for different size images with different parameters.

Thus, the client can pre-compute all randomization factors g^r and h^r during idle times. In this case, the feature descriptor matching an image takes less time.

7 Conclusion

In this present paper, we propose a double decryption-based privacy-preserving SIFT scheme over the encrypted domain. This scheme can be used to handle the privacy-preserving problem encountered in a cloud computing environment. We hope that the clients do no or little computation, while the company, server or cloud can finish almost all the tasks of

SIFT-based applications without learning anything to obtain the client's privacy. The experimental result shows that the proposed scheme is correct and appropriate for many types image database. As future works, we will continue to study on privacy-preserving image and improve the homomorphic comparison efficiency.

Acknowledgements The work of Zheng-An Yao was partially supported by the NSFC (Grant Nos. 11271381, 11431015) and China 973 Program (Grant No. 2011 CB808000). Chun-Ming Tang's work was supported by the NSFC (Grant No. 11271003), the National Research Foundation for Doctoral Program of Higher Education of China (Grant No. 20134410110003), and the Project of Department of Education of Guangdong Province (Grant No. 2013KJCX0146). Last, but not least, the authors are very grateful to the editor and the reviewers for their valuable comments.

Compliance with ethical standards

Conflict of interest There is no conflict of interest between the authors.

Informed consent This article does not contain any studies with human participants performed by any of the authors.

References

Blake IF, Kolesnikov V (2006) Conditional encrypted mapping and comparing encrypted numbers. In: Financial Cryptography and Data Security, Springer, pp 206–220

Blakley GR et al (1979) Safeguarding cryptographic keys. Proc Natl Comp Conf 48:313–317

Blundo C, Cresti A, De Santis A, and Vaccaro U (1994) Fully dynamic secret sharing schemes. In: Advances in Cryptology–CRYPTO'93, Springer, pp 110–125

Blundo C, De Santis A, De Simone R, Vaccaro U (1997) Tight bounds on the information rate of secret sharing schemes. Designs Codes Cryptograph 11(2):107–110

Blundo C, De Santis A, Vaccaro U (1993) Efficient sharing of many secrets. In: STACS 93, Springer, pp 692–703

Boneh D, Goh EJ, and Nissim K (2005). Evaluating 2-dnf formulas on ciphertexts. In Theory of cryptography, pages 325–341. Springer

Bresson E, Catalano D, Pointcheval D (2003) A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In: Advances in Cryptology-ASIACRYPT 2003, Springer, pp 37–54

Brown M, Lowe DG (2007) Automatic panoramic image stitching using invariant features. Int J Comp Vision 74(1):59–73

Damgård I, Geisler M, Krøigaard M (2007) Efficient and secure comparison for on-line auctions. In: Information security and privacy, Springer, pp 416–430

Damgård I, Geisler M, Krøigaard M (2009) A correction to 'efficient and secure comparison for on-line auctions'. Int J Appl Cryptograph 1(4):323–324

De Santis A, Desmedt Y, Frankel Y, Yung M (1994) How to share a function securely. In: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing, ACM, pp 522–533

Garay JA, Schoenmakers B, Villegas J (2007) Practical and secure solutions for integer comparison. In Public Key Cryptography-PKC 2007, Springer, pp 330–342

Goldreich O (2004).Foundations of cryptography: volume 2, basic applications. Cambridge University Press

Grauman K and Fergus R (2013). Learning binary hash codes for large-scale image search. In: Machine learning for computer vision, Springer, pp 49–87

Hsu CY, Lu CS, Pei SC (2012) Image feature extraction in encrypted domain with privacy-preserving sift. IEEE Trans Image Process 21(11):4593–4607

Jégou H, Zisserman A (2014) Triangulation embedding and democratic aggregation for image search. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, IEEE, pp 3310–3317

Ke Y, Sukthankar R, Huston L, Ke Y, Sukthankar R (2004) Efficient near-duplicate detection and sub-image retrieval. In: Proceedings of the 12th Annual ACM International Conference on Multimedia, vol 4, pp 869–876

Kolesnikov V, Sadeghi AR, Schneider T (2009) Improved garbled circuit building blocks and applications to auctions and computing minima. In: Cryptology and Network Security, Springer, pp 1–20

Li J, Kim K, Zhang F, Chen X (2007) Aggregate proxy signature and verifiably encrypted proxy signature. Provable Security, Springer, pp 208–217

Lindeberg T (1994) Scale-space theory: a basic tool for analyzing structures at different scales. J Appl Stat 21(1–2):225–270

Lindell Y, Pinkas B (2009) A proof of security of yao's protocol for two-party computation. J Cryptol 22(2):161–188

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comp Vision 60(2):91–110

Naor M, Pinkas B, Sumner R (1999) Privacy preserving auctions and mechanism design. In Proceedings of the 1st ACM conference on Electronic commerce, ACM, pp 129–139

Paillier P (1999) Public-key cryptosystems based on composite degree residuosity classes. In: Advances in cryptology-EUROCRYPT'99, Springer, pp 223–238

Peter A, Tews E, Katzenbeisser S (2013) Efficiently outsourcing multiparty computation under multiple keys. IEEE Trans Inform Foren Security 8(12):2046–2058

Qin Z, Yan J, Ren K, Chen CW, Wang C (2014) Towards efficient privacy-preserving image feature extraction in cloud computing. In: Proceedings of the ACM International Conference on Multimedia, ACM, pp 497–506

Sadeghi AR, Schneider T, Wehrenberg I (2010) Efficient privacy-preserving face recognition. In: Information, Security and Cryptology–ICISC 2009. Springer, Heidelberg, pp 229–244

Shamir A (1979) How to share a secret. Commun ACM 22(11):612–613

Tang Y, Liu L (2015) Privacy-preserving multi-keyword search in information networks. IEEE Trans Knowl Data Eng 27(9):2424–2437

Tuyls P, Akkermans AHM, Kevenaar TAM, Schrijen GJ, Bazen AM, Veldhuis RNJ (2005) Practical biometric authentication with template protection. In: Audio-and Video-Based Biometric Person Authentication, Springer, pp 436–446

Wang B, Yu S, Lou W, Hou YT (2014) Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud. In: Proceedings IEEE INFOCOM, 2014, pp 2112–2120

Wang J, Ma H, Tang Q, Li J, Zhu H, Ma S, Chen X (2013) Efficient verifiable fuzzy keyword search over encrypted data in cloud computing. Comp SciInform Syst 10(2):667–684

Yao ACC (1982) Protocols for secure computations. In: 23rd annual symposium on foundations of computer science (Chicago, Ill., 1982), IEEE, New York, pp 160–164

Yao ACC (1986) How to generate and exchange secrets. In: 27th Annual Symposium on Foundations of Computer Science, 1986, IEEE, pp 162–167

Zhou J, Cao Z, Dong X, Lin X (2015) PPDM: A privacy-preserving protocol for cloud-assisted e-healthcare systems. IEEE J Select Topics Signal Process 9(7):1332–1344