PPOIM: Privacy-Preserving Shape Context Based Image Denoising and Matching with Efficient Outsourcing

Meng Zheng, Jun Zhou(✉), Zhenfu Cao, and Xiaolei Dong

Shanghai Key Lab for Trustworthy Computing, East China Normal University, Shanghai 200062, China
zmecdu2016@163.com, {jzhou,zfcao,dongxiaolei}@sei.ecnu.edu.cn

Abstract. With the emerging techniques of wireless communication and cloud computing, large volumes of multimedia data are outsourced from resource constrained users to the cloud with abundant resource for both delegated storage and computation. Unfortunately, there is a risk of users' image privacy leakage in the process of outsourcing to untrusted cloud. Most of the existing work achieved privacy-preserving image feature extraction and matching by using public key (fully) homomorphic encryption (FHE), but the heavy computational overhead and communication overhead cannot adapt to resource-constrained mobile devices. Other works disabled to realize image denoising in the encrypted domain or only focused on the scale-invariant feature transform (SIFT) descriptor that is inappropriate for position-sensitive feature extraction. To address these issues, in this paper, a privacy-preserving shape context based image denoising and matching protocol PPOIM with efficient outsourcing is proposed. Firstly, to improve the accuracy of image matching, a privacy-preserving image denoising scheme PPID is proposed without exploiting public key FHE. Then, based on PPID, a privacy-preserving image matching protocol PPOIM adopting shape context descriptor is devised, where two secure and efficient comparison and counting protocols in the encrypted domain are presented. All the original image privacy, query image privacy and image matching result privacy are well protected. Finally, formal security proof and extensive simulations on real-world data sets demonstrate the efficiency and practicability of our proposed PPOIM.

Keywords: Image matching · Privacy-preserving
Shape context descriptor · Secure outsourced computation

1 Introduction

With the development of big data and social network like Flickr or Facebook, huge amounts of personal users' multimedia data are delegated to the cloud from the resource-constrained mobile devices for both outsourced storage and

D. Naccache et al. (Eds.): ICICS 2018, LNCS 11149, pp. 215–231, 2018.
https://doi.org/10.1007/978-3-030-01950-1_13

outsourced computation with expensive complexity. Among types of image processing, image matching have played an increasingly important role in our everyday life. The widely adopted technique of content-based image match means that the cloud returns the boolean match result between images and the user's queried one with similar features such as color, shape and texture that are extracted by exploiting scale-invariant feature transform (SIFT) descriptor, shape context (SC) descriptor, etc. Taking medical image for example, the physicians can judge the aging degree of the elderly persons, by matching their medical image (i.e. X-ray film) with the pattern images signaling different levels of aging, adopting the extracted features such as the step length and the angle with which the elderly's limbs can be lifted.

Unfortunately, the cloud server either works under the semi-honest model or malicious model, where the cloud either strictly carries out the protocol specifications but intending to extract the private information from the interactions with users, or performs arbitrarily to destruct the protocol execution. Therefore, it would disclose the private health condition of the elderly persons by delegating the medical images in their plaintext to the cloud for feature extraction and matching. How to devise an efficient privacy preserving image feature extraction and matching protocol becomes a critical issue for convincing solutions.

Recently, a series of research has focused on the field of privacy-preserving image feature extraction and matching [1–3,6–8,12,13,16–19,21,23–25]. Hsu et al. [4] studied privacy-preserving outsourced feature extraction in the encrypted domain, by using Paillier's additive homomorphic encryption. Unfortunately, their protocol is either computationally-intensive or risks the privacy leakage of the original image. To address the issues, Hu et al. [5] devised a secure outsourcing computation of feature extraction over encrypted image data, by splitting the original image and designing privacy-preserving multiplication and comparison protocols executed by two non-colluded servers, by exploiting Brakerski et al.'s somewhat homomorphic encryption [15]. However, the level of fully homomorphism respectively proposed in [14] and [15] is restricted and the ciphertext expansion would increase every time a ciphertext multiplication is required. Thus the heavy computational and communication overhead in both [4] and [5] is intolerable by resource-constrained devices. J. Zhou et al. [12] proposed an efficient privacy-preserving image feature extraction protocol, however all the above [4,5,12] adopted SIFT descriptor, which is only appropriate for searching images with a transforming rotation, scaling, and translation, but cannot be applied to the scenario of image matching adopting the features as relative positions between pixels, as is suggested in the example for judging the aging level of the elderly. Belongie et al. [8] presented an approach to measure similarity between shapes for object recognition based on shape context based descriptor. However, the issue of image privacy-preserving was not considered. In [6], Wang et al. studied privacy-preserving shape-based feature extraction by exploiting the techniques of homomorphic encryption and the garbled circuit protocol, respectively. The high computational complexity can still not adapt to resource-constrained users.

On the other hand, image noise may be introduced under different conditions from intrinsic sensors or extrinsic environments, which are often difficult to avoid in practice and significantly affect the accuracy of image matching. Zheng et al. [9] proposed a privacy-preserving image denoising protocol from external cloud databases by using secure similarity search, Yao's garbled circuits and image denoising operations, to ensure that similar patches with high quality are precisely obtained after encrypted similarity search. Unfortunately, the denoising operations were completed in the plaintext domain without considering image privacy protection. To address the issues mentioned above, in this paper, a privacy-preserving shape context based image denoising and matching protocol PPOIM with efficient outsourcing is proposed. The main contributions are summarized as follows.

Firstly, a privacy-preserving image denoising protocol PPID is proposed in the encrypted domain, by devising a lightweight secure outsourced computation without public key fully homomorphic encryption (FHE).

Secondly, based on the proposed PPID, we present an efficient privacy-preserving image matching scheme PPOIM based on shape context descriptor. Especially, two efficient comparison and counting protocols in the encrypted domain are carefully designed. Both the original image privacy and the matching result privacy are well protected, and only the authorized user can successfully decipher the final matching result.

Finally, formal security proof and extensive evaluations demonstrate the efficiency and practicability of our PPOIM. Both the computational cost and communication cost are dramatically reduced, compared to the state-of-the-art using public key FHE.

The remainder of this paper is organized as follows. We present the network architecture and the security model in Sect. 2. Then the privacy-preserving image denoising protocol PPID and the privacy-preserving shape context based image matching protocol are proposed in Sect. 3. Formal security proof and performance evaluations are respectively presented in Sects. 4 and 5. Finally, we conclude our paper in Sect. 6.

2 Network Architecture and Security Model

2.1 Network Architecture

The network model of privacy-preserving shape context based image denoising and matching mainly comprises three entities: the data owner, the user and the cloud, which are demonstrated in Fig. 1. The main procedure of our proposed PPOIM are described as follows, (1) The data owner outsources an encrypted database of image patches to the cloud for generating high quality similar patches; (2) The user sponsors an image search token request to the data owner; (3) The data owner performs the search token authorization to the user if her/his image query is permitted; (4) The user uploads the encrypted

query image together with the search token to the cloud; (5) The cloud performs privacy-preserving image denoising and matching by adopting shape context based descriptor and calculating the matching cost in the encrypted domain; (6) The cloud returns all encrypted matching results to the user for decrypting, if the matching cost is smaller than the cost threshold set by the user, two images are considered to be matched each other.

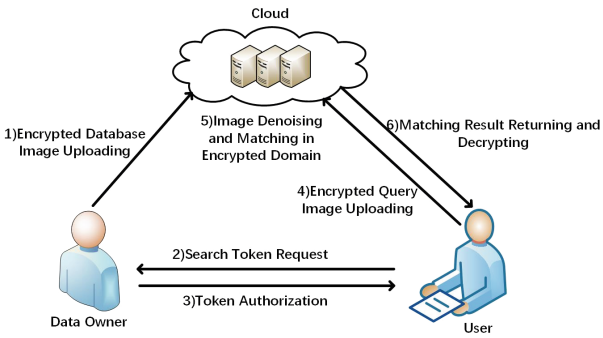

Fig. 1. Network architecture of privacy-preserving image denoising and matching

2.2 Security Model

We formally define the image privacy and the matching result privacy for our proposed PPOIM. The cloud is assumed to be honest-but-curious, which strictly executes the protocol specification but tries its best to extract the private information from the interactions among data owner, user and itself. Image privacy refers to that the data owner's database images cannot be accessed by the collusion between the cloud and malicious users and the user's query image cannot be disclosed to the collusion of the cloud and malicious owners. The matching result privacy means that whether the query image matches the database image can only be accessed by the authorized users. The formal security models of these three types of privacy are detailed in the full paper.

3 The Proposed PPOIM

In this section, a privacy-preserving shape context based image denoising and matching protocol PPOIM with efficient outsourcing is proposed, which is composed of three phases, namely the setup phase generating the required parameters, the privacy-preserving image denoising phase PPID, and the privacy-preserving image matching phase PPOIM where the final matching result can be decrypted by the authorized user.

3.1 Setup Phase

On input 1^λ where λ is the security parameter, the system runs a trapdoor permutation generator denoted as a probabilistically polynomial time (PPT) algorithm $\mathcal{G}(1^\lambda)$ and outputs a tuple of permutations (f, f^{-1}) on $\{0,1\}^{2\lambda}$ with a pair of corresponding keys (PK_f, SK_f). It also outputs two hash functions $H_0,\ H_1 : \{0,1\}^* \rightarrow \{0,1\}^{2\lambda}$ and a cluster of locality-sensitive hash (LSH) functions $h_i : \{0,1\}^* \rightarrow \{0,1\}^\lambda (i = 1, 2, \cdots, l)$. The public parameters are $PPR = (PK_f, H_0, H_1, h_i (i = 1, 2, \cdots, l))$ and the secret key is SK_f assigned to the user. Besides, suppose there is a secure symmetric encryption scheme $SE = (SE.Setup, SE.KGen, SE.E, SE.D)$ with a secret key $\mathbf{K} = (K_g, K_p)$ shared between the data owner and the user, and $F : \{0,1\}^\lambda \times \{0,1\}^* \rightarrow \{0,1\}^\lambda$ is a pseudorandom function (PRF).

3.2 The Proposed Privacy-Preserving Image Denoising Protocol PPID

In this subsection, an efficient privacy-preserving image denoising protocol PPID is presented, which is composed of four algorithms: **IndexGen** performed on the data owner side, encrypting patch databases with their corresponding secure indexes by exploiting locality-sensitive hashing (LSH) and symmetric encrytpion (SE), and uploading the encrypted database images to the cloud; **Request** executed on the user side, generating a secure query search token, and transmitting the search token and encrypted query patch to the cloud; **Search** run on the cloud side, ranking all candidate patches and filtering the false positive candidates for denoising operation; and **Denoising** carried out on the cloud side, recovering the clean encrypted patch.

(1) $\{[\mathbf{P}], \mathcal{D}\} \leftarrow \mathbf{IndexGen}(\mathbf{K}, PK_f, \mathbf{P})$. It takes as input the secret key $\mathbf{K} = (K_g, K_p)$, the public key PK_f for patch encryption and the patch set $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_N\}$, where N is the total number of database patches, and returns $\{[\mathbf{P}], \mathcal{D}\}$, where $[\mathbf{P}] = \{[\mathbf{p}_1], [\mathbf{p}_2], \cdots, [\mathbf{p}_N]\}$ and $\mathcal{D}$ refer to the ciphertexts of database images and a generic dictionary.

Let $p_{i,t} = (\rho_{i,t}, \theta_{i,t}) (i = 1, 2, \cdots, N; t = 1, 2, \cdots, n)$ be the polar coordinate of the t-th pixel in database patch $\mathbf{p}_i$, and $\mathbf{p}_i = \{p_{i,t}\}_{t=1}^n = \{(\rho_{i,t}, \theta_{i,t})\}_{t=1}^n$. The ciphertexts of database images $[\mathbf{P}] = \{[\mathbf{p}_i]\} (i = 1, 2, \cdots, N)$ are encrypted as follows. For brief description, we only detailed the process for encrypting $\rho_{i,t}$, and $\theta_{i,t}$ can be encrypted in the same way. The image data owner randomly chooses three big primes p, q, h of $|p| = |q| = |h| = \lambda$ which are kept secret, and computes the publicized $N^{"} = pq$, $N^{'} = pqh$. The message space of $\rho_{i,t}$ is on $\mathbb{Z}_{N^{"}}$ as a hidden subgroup of $\mathbb{Z}_{N'}$. Then, the owner computes $\rho_{i,t,p} \equiv \rho_{i,t} \bmod p$, $\rho_{i,t,q} \equiv \rho_{i,t} \bmod q$. She/he also randomly selects $K_{i,t} \in_R \mathbb{Z}_h$, and computes the additive blinding factor $U_{i,t}^{add} = K_{i,t} N^{"} \in_R \mathbb{Z}_{N'}$ and the multiplicative blinding factor $U_{i,t}^{mul} = K_{i,t} N^{"} + 1 \in_R \mathbb{Z}_{N'}$ $(i = 1, 2, \cdots, N; t = 1, 2, \cdots, n)$ such that the final image matching results in our proposed PPOIM can be correctly obtained in the

decryption phase **ImgDec** by calling the algorithm **PPOIM.Dec**$(\cdot)$ where all the additive and multiplicative blinding factors $U_{i,t}^{add}$, $U_{i,t}^{mul}$ can be cancelled out after modular $N^{"}$. Since we have $1 \equiv q^{-1}q \ mod \ p$, $1 \equiv p^{-1}p \ mod \ q$, the data owner calculates the ciphertexts as follows,

$$\begin{aligned} &C_{1,1} = f_{PK_f}(p \parallel h), \\ &C_{2,\rho_{i,t}} = q^{-1}q\rho_{i,t,p}^{p} + p^{-1}p\rho_{i,t,q}^{q} + U_{i,t}^{add} \ mod \ N', \\ &C_{3,\rho_{i,t}} = (q^{-1}q\rho_{i,t,p}^{p} + p^{-1}p\rho_{i,t,q}^{q})U_{i,t}^{mul} \ mod \ N'. \end{aligned} \tag{1}$$

where $\parallel$ means the concatenation operation, and q^{-1}, p^{-1} respectively denote the inverses of q and p in $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$. Finally, the data owner computes $C_{ram,\rho}^{add} = H_0(p \parallel h \parallel \bigcup_{i=1,t=1}^{N,n} C_{2,\rho_{i,t}})$, $C_{ram,\rho}^{mul} = H_0(p \parallel h \parallel \bigcup_{i=1,t=1}^{N,n} C_{3,\rho_{i,t}})$, and denotes $[\rho_{i,t}] = (C_{2,\rho_{i,t}}, C_{3,\rho_{i,t}})$. Note that $[\theta_{i,t}] = (C_{2,\theta_{i,t}}, C_{3,\theta_{i,t}})$ can be computed in the same way. We have $[\mathbf{p}_{i,t}] = ([\rho_{i,t}], [\theta_{i,t}])$ and the ciphertexts of database images $[\mathbf{P}] = (\{[\mathbf{p}_{i,t}](i = 1, 2, \cdots, N; t = 1, 2, \cdots, n)\}, C_{ram,\rho}^{add}, C_{ram,\rho}^{mul}, C_{ram,\theta}^{add}, C_{ram,\theta}^{mul})$. We denote the encryption algorithm to generate ciphertexts of database images $[\mathbf{P}]$ as **PPOIM.Enc**$(\cdot)$ which would also be exploited in the following phases of our proposed PPID. Then, the data owner initializes a dictionary $\mathcal{D}$ and the LSH value set $\mathbf{G}$ as two empty sets. For each patch $\mathbf{p}_i$ in patch set $\mathbf{P}$, the data owner computes LSH values with l LSH functions $h_1(\cdot), h_2(\cdot), \cdots, h_l(\cdot)$,

$$\mathbf{g}_i = (h_1(\mathbf{p}_i) \parallel 1, \cdots, h_l(\mathbf{p}_i) \parallel l), \tag{2}$$

where vector $\mathbf{g}_\mathrm{i}$ is the i-th element in $\mathbf{G}$, $\mathrm{g}_{i,j} = h_j(\mathbf{p}_i) \parallel j (j = 1, 2, \cdots, l)$ is the j-th element in vector $\mathbf{g}_\mathrm{i}$. Then, for each $\mathrm{g}_{i,j}$ in $\mathbf{g}_\mathrm{i} \in \mathbf{G}$, the owner generates

$$K_{1,i,j} = F(K_\mathrm{g}, 1 \parallel \mathrm{g}_{i,j}), K_{2,i,j} = F(K_\mathrm{g}, 2 \parallel \mathrm{g}_{i,j}). \tag{3}$$

The data owner initializes a counter $ctr = 0$. For each $\mathrm{g}_{i,j}$, if there exists any $\mathrm{g}_{k,j} = \mathrm{g}_{i,j} (k \in \{1, 2, \cdots, N\})$, then it considers $\mathbf{p}_k$ is associated with $\mathrm{g}_{i,j}$ and $ctr \leftarrow ctr + 1$. The data owner computes tag $u_{i,j}$ by applying pseudorandom function F and encrypts the corresponding patch sub-identifier $id_{k,j}$ using the symmetric encryption scheme SE as follows,

$$u_{i,j} = F(K_{1,i,j}, ctr), v_{i,j} = SE.E(K_{2,i,j}, id_{k,j}), \tag{4}$$

where $id_k = id_{k,1} \parallel id_{k,2} \parallel \cdots \parallel id_{k,l}$ is the unique identifier of a database patch $\mathbf{p}_k$ and $id_{k,j} (j = 1, 2, \cdots, l)$ is the sub-identifier of $h_j(\mathbf{p}_k) \parallel j$ in $\mathbf{g}_k$. Then, the tag-ciphertext pair $(u_{i,j}, v_{i,j})$ is inserted to a generic dictionary $\mathcal{D}$. Finally, the data owner sends $([\mathbf{P}], \mathcal{D})$ to the cloud server.

(2) $\{\mathrm{Q}, [\mathbf{q}], [t^{"}], [T]\} \leftarrow$ **Request**$(\mathbf{K}, PK_f, \mathbf{q}, t^{"}, T)$. When a user wants to request the database, she/he firstly need to obtain the token authorization from the data owner by receiving $C_{1,1} = f_{PK_f}(p \parallel h)$. Then, she/he decrypts $p \parallel h = f_{SK_f}^{-1}(C_{1,1})$ by using secret key SK_f and computes $q = N^{'}(ph)^{-1}$. After that, the user generates the ciphertext $[\mathbf{q}]$ and a secure search token Q for the query patch $\mathbf{q}$ as follows. The user firstly hashes $\mathbf{q}$ into a vector of l LSH values

$$\mathbf{g} = \{h_1(\mathbf{q}) \parallel 1, \cdots, h_l(\mathbf{q}) \parallel l\}, \tag{5}$$

where $g_j = h_j(\mathbf{q}) \parallel j$ $(j = 1, 2, \cdots, l)$ is the j-th element of the $\mathbf{g}$. For each LSH value g_j, a sub-token $Q_j = (K_{1,j}, K_{2,j})$ is generated via

$$K_{1,j} \leftarrow F(K_g, 1 \parallel g_j), K_{2,j} \leftarrow F(K_g, 2 \parallel g_j). \tag{6}$$

The resulting secure search token $Q = \{Q_1, Q_2, \cdots, Q_l\}$. On the other hand, the user randomly selects $K_t \in_R \mathbb{Z}_h$, and computes $U_t^{add} = K_t N^{"}$, $U_t^{mul} = K_t N^{"} + 1 \in_R \mathbb{Z}_{N'} (t = 1, 2, \cdots, n-1)$ such that the final matching result would be successfully decrypted after modulo $N^{"}$. Then the user encrypts patches $q_t \in \mathbf{q} (t = 1, 2, \cdots, n)$ with $\mathbf{PPOIM.Enc}(\cdot)$ to generate the ciphertexts $[q_t] = ([\rho_t], [\theta_t])$. Thus, $[\rho_t] = (C_{2,\rho_t}, C_{3,\rho_t})$, $[\theta_t] = (C_{2,\theta_t}, C_{3,\theta_t})$, $[\mathbf{q}] = \{[\rho_t], [\theta_t], C'^{,add}_{ram,\rho}, C'^{,mul}_{ram,\rho}, C'^{,add}_{ram,\theta}, C'^{,mul}_{ram,\theta}\}$. In addition, the user chooses two thresholds $t^{"}, T$ respectively for obtaining the candidate patches for denoising and for matching cost comparison to derive the final image matching result, encrypts them into $[t^{"}], [T]$ by exploiting algorithm $\mathbf{PPOIM.Enc}(\cdot)$. Finally, the user sends $(Q, [\mathbf{q}], [t^{"}], [T])$ to the cloud.

(3) $\{S^*, H\} \leftarrow \mathbf{Search}(Q, [\mathbf{q}], [t^{"}], [\mathbf{P}], \mathcal{D})$. For each sub-token Q_j in Q, the cloud re-computes the pseudorandom tag $u_j = F(K_{1,j}, ctr)$, where ctr is a self-incremental counter and initialized as 0. Let f_{id_i} be an occurrence counter initialized as 0. The cloud searches the generic dictionary $\mathcal{D}$ according to the pseudorandom tag u_j to locate the associated $v_{i,j} (j \in \{1, 2, \cdots, l\})$. If $u_j = u_{i,j} (j \in \{1, 2, \cdots, l\})$, it decrypts the corresponding patch identifier $id_{k,j} = SE.D(K_{2,j}, v_{i,j})$ via $K_{2,j}$, and increases $f_{id_i} \leftarrow f_{id_i} + 1$. Then, the cloud ranks the candidates $\mathbf{p}_i$ based on the occurrence counter f_{id_i}, and derives an initial set S^* of candidate patches.

However, LSH is an approximation algorithm that trades accuracy for efficiency, which usually locates a large number of candidates with false positives introduced. Thus, to filter the false positive candidates, the cloud computes distance between candidate $\mathbf{p}_i$ in S^* and query image $\mathbf{q}$. For each encrypted candidates patch $[\mathbf{p}_i] = \{[\mathbf{p}_{i,t}]\} = \{([\rho_{i,t}], [\theta_{i,t}])\} = \{((C_{2,\rho_{i,t}}, C_{3,\rho_{i,t}}), (C_{2,\theta_{i,t}}, C_{3,\theta_{i,t}}))\} (t = 1, 2, \cdots, n)$ and the encrypted query patch $[\mathbf{q}] = \{[\mathbf{q}_t]\} = \{([\rho_t], [\theta_t])\} = \{((C_{2,\rho_t}, C_{3,\rho_t}), (C_{2,\theta_t}, C_{3,\theta_t}))\} (t = 1, 2, \cdots, n)$, the cloud computes the squared distance between $[\mathbf{p}_i]$ and $[\mathbf{q}]$ in the encrypted domain to securely refine the ranking for each candidate in S^*.

$$d^2([\mathbf{p}_i], [\mathbf{q}]) = \sum_{t=1}^{n} (C^2_{3,\rho_{i,t}} + C^2_{3,\rho_t}) - 2 \sum_{t=1}^{n} [C_{3,\rho_{i,t}} C_{3,\rho_t} cos(C_{2,\theta_{i,t}} - C_{2,\theta_t})], \tag{7}$$

where the cosine function is approximated by aggregating the first t' items in its power series expansion as $cos\ x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots (-1)^{t'} \frac{x^{2t'}}{(2t')!}$ (i.e. In performance evaluation, we would study the impact of different t' on the accuracy of image matching result and the efficiency of our proposed PPOIM.)

Then the cloud compares the squared distance $d^2([\mathbf{p}_i], [\mathbf{q}])$ with threshold $[t^{"}]$ in the encrypted domain. Let the binary representations of $t^{"}$ and $d^2(\mathbf{p}_i, \mathbf{q})$

be $t^{"} = \overline{m_{n-1}m_{n-2}\cdots m_0}$ and $d^2(\mathbf{p}_i, \mathbf{q}) = \overline{m'_{n-1}m'_{n-2}\cdots m'_0}$. Owing to the fully homomorphic property of algorithm **PPOIM.Enc**$(\cdot)$, we denote

$$\begin{aligned}&[t^{"}] = \mathbf{PPOIM.Enc}(m_{n-1})\mathbf{PPOIM.Enc}(2^{n-1}) + \mathbf{PPOIM.Enc}(m_{n-2})\\ &\mathbf{PPOIM.Enc}(2^{n-2}) + \cdots + \mathbf{PPOIM.Enc}(m_0)\mathbf{PPOIM.Enc}(1),\\ &d^2([\mathbf{p}_i], [\mathbf{q}]) = \mathbf{PPOIM.Enc}(m'_{n-1})\mathbf{PPOIM.Enc}(2^{n-1}) + \mathbf{PPOIM.Enc}\\ &(m'_{n-2})\mathbf{PPOIM.Enc}(2^{n-2}) + \cdots + \mathbf{PPOIM.Enc}(m_0{}')\mathbf{PPOIM.Enc}(1).\end{aligned} \tag{8}$$

Then by exploiting the method of successive division with **PPOIM.Enc**(2) that can also be executed and uploaded by the user in the previous **Request** algorithm, the cloud can derive the binary encryption of $[t^{"}] = \overline{\mathbf{PPOIM.Enc}(m_{n-1})\mathbf{PPOIM.Enc}(m_{n-2})\cdots\mathbf{PPOIM.Enc}(m_0)} = \overline{m^e_{n-1}m^e_{n-2}\cdots m^e_0}$ and $d^2([\mathbf{p}_i], [\mathbf{q}]) = \overline{\mathbf{PPOIM.Enc}(m'_{n-1})\mathbf{PPOIM.Enc}(m'_{n-2})\cdots\mathbf{PPOIM.Enc}(m'_0)} = \overline{m'^e_{n-1}m'^e_{n-2}\cdots m'^e_0}$. For binary representations, we have the following observation for $i = 0, 1, \cdots, n-1$,

$$\begin{aligned}&m_i > m'_i \ if\ and\ only\ if\ m_i m'_i + m_i = 1,\\ &m_i = m'_i \ if\ and\ only\ if\ m_i + m'_i + 1 = 1,\\ &m_i < m'_i \ if\ and\ only\ if\ m_i m'_i + m_i + 1 = 1.\end{aligned} \tag{9}$$

Therefore, according to the property of full homomorphism of **PPOIM.Enc**$(\cdot)$, the cloud can evaluate Eq. (9) in the encrypted domain. To compare $t^{"}$ and $d^2(\mathbf{p}_i, \mathbf{q})$, the binary chop method is adopted. Specifically for $l = \lceil \frac{n}{2} \rceil$,we have

$$\underbrace{m_{n-1}\cdots m_l}_{hbs(t^{"})}\underbrace{m_{l-1}\cdots m_0}_{lbs(t^{"})} > \underbrace{m'_{n-1}\cdots m'_l}_{hbs(d^2(\mathbf{p}_i,\mathbf{q}))}\underbrace{m'_{l-1}\cdots m'_0}_{lbs(d^2(\mathbf{p}_i,\mathbf{q}))} \tag{10}$$

if and only if $(hbs(t^{"}) > hbs(d^2(\mathbf{p}_i, \mathbf{q}))) \vee (hbs(t^{"}) = hbs(d^2(\mathbf{p}_i, \mathbf{q}))) \wedge (lbs(t^{"}) > lbs(d^2(\mathbf{p}_i, \mathbf{q})))$, where $hbs(x)$, $lbs(x)$ respectively refer to the higher binary sequence and the lower binary sequence of x. To recursively performing the comparison until deriving the final output, it is also required to define the following three variations $h_{i,j}$,$e_{i,j}$ and $l_{i,j}$, respectively referring to the boolean logic values for the conditions $\overline{m_{i+j-1}\cdots m_i} > \overline{m'_{i+j-1}\cdots m'_i}$, $\overline{m_{i+j-1}\cdots m_i} = \overline{m'_{i+j-1}\cdots m'_i}$, $\overline{m_{i+j-1}\cdots m_i} \geq \overline{m'_{i+j-1}\cdots m'_i}$. It is obviously observed that $h_{0,n}, e_{0,n}, l_{0,n}$ will be the final result. For each time, by selecting $l = \lceil \frac{j}{2} \rceil$ and combining Eqs. (9) and (10), we have

$$\begin{aligned}&(1) If\ j = 1, h_{i,j} = m_i m'_i + m_i\ , Else\ h_{i,j} = h_{i+l,j-1} + e_{i+l,j-l}t_{i,l};\\ &(2) If\ j = 1, e_{i,j} = m_i + m'_i\ , Else\ e_{i,j} = e_{i+l,j-1}e_{i,j};\\ &(3) If\ j = 1, l_{i,j} = m_i m'_i + m_i + 1\ , Else\ l_{i,j} = t_{i+l,j-1} + e_{i+l,j-l}l_{i,l}.\end{aligned} \tag{11}$$

By comparing the threshold for denoising $[t^{"}]$ with each $d^2([\mathbf{p}_i], [\mathbf{q}])$ corresponding to each candidate $\mathbf{p}_i$ in S^*, all the encrypted comparing results $H = \{[h_{0,n}]_i\}(i = 1, 2, \cdots, N)$ can be computed according to Eq. (11).

(4) $[\hat{\mathbf{q}}] \leftarrow \textbf{Denoising}([\mathbf{q}], S^*, H)$ Collecting the encrypted database patch candidates in S^*, the cloud performs privacy-preserving image deniosing by exploiting the classical technique of non-local means (NLM) [10], [11], in which a weighted average computation in the encrypted domain is adopted. Given a noisy patch $[\mathbf{q}]$ and a set of ranked patches $S^* = \{[\mathbf{p}_1], [\mathbf{p}_2], \cdots, [\mathbf{p}_N]\}$, the clean patch $[\hat{\mathbf{q}}]$ is estimated as the weighted average of all ranked patches, the detailed process is described as follows.

To compute the normalizing factor $[Z]$, we define h as a filtering parameter depending on the standard deviation σ of the zero-mean Gaussian noise. Next, the cloud calculates $[Z] = \sum_{i=1}^{N} e'$, where

$$e' = e^{-d^2([\mathbf{p}_i],[\mathbf{q}])h^{-2}} = \sum_{i'=0}^{t'} (-1)^{i'} \frac{(d^2([\mathbf{p}_i],[\mathbf{q}])h^{-2})^{i'}}{i'!} \tag{12}$$

The index function e^x is approximated by aggregation the first $t^{'}$ items in its power series expansion as $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^{t'}}{t'!}$ (i.e. In performance evaluation, we would study the impact of different t' on the accuracy of image matching result and the efficiency of our proposed PPOIM) and the h^{-1} is the inverse of h. Next, the cloud calculates the weight $\omega([\mathbf{q}], [\mathbf{p}_i]) = [Z]^{-1}e'$, where $[Z]^{-1}$ is the inverse of $[Z]$. Finally, the clean patch $[\hat{\mathbf{q}}]$ is estimated as the weighted average of all encrypted ranked patches,

$$[\hat{\mathbf{q}}] = \sum_{i=1}^{N} \omega([\mathbf{q}], [\mathbf{p}_i])[\mathbf{p}_i][h_{0,n}]_i = \sum_{i=1}^{N} [Z]^{-1}e'[\mathbf{p}_i][h_{0,n}]_i. \tag{13}$$

If the full query image $\mathbf{I}_q$ is composed of several patches, then for each patch, the cloud adopts the same denoising method as is explained above to process patch $[\mathbf{q}]$.

3.3 The Proposed Privacy-Preserving Image Matching Protocol PPOIM

After denoising query image in the cloud, an estimate of the original query image $[\hat{\mathbf{I}}_q]$ composed of all $[\hat{\mathbf{q}}]$ can be produced. In this section, we firstly clarify the definition of Shape Context (SC) descriptor. Then, based on our proposed PPID in Sect. 3.2, a privacy-preserving SC-based image matching protocol PPOIM with efficient outsourcing is proposed, which consists of three algorithms **SCGen**, **ImgMatch** and **ImgDec**. We assume that as long as at least one shape in the database image matches the query image $[\hat{\mathbf{I}}_q]$, these two images matches successfully, regardless of the position and rotation angle of the shape in the database image. We also assume that database images and the query image are in the same polar coordinate system, which means that the query image shares the center point with database images. The cloud computes matching cost between the encrypted denoised query image $[\hat{\mathbf{I}}_q]$ and all database images $[\mathbf{I}_i](i = 1, 2, \cdots, N)$, then compares all matching cost with a threshold $[T]$.

Shape Context in Plaintext Domain. Belongie et al. [8] introduced the idea of shape context. In their work, a shape is represented by a set of points sampled from the contours, and shape context describes location information about all other boundary points relative to a specific boundary point in the shape. Here, we prefer to sample the shape with roughly uniform spacing. Each shape context is a coarse log-polar histogram of the coordinates of the remaining points measured using the reference point as the origin and the line joining the reference point and the center as the pole axis. Additionally, the center of mass of any shape is invariant to scaling, rotation or translation. Figure 2 shows the definition of Shape Context.

The shape 'A' in Fig. 2 is composed of a set of discrete points $A = \{a_i\}(i = 1, 2, \cdots, n)$ sampled from the contour. To compute a shape context of a_i in A, we create a new polar coordinate. Let the referenced point a_i be the new pole and the line joining a_i and the center o of the shape be the new pole axis $\overline{a_i o}$. The set of vectors originating from a_i to the remained $n-1$ points is generated. To compute the shape context, we firstly divide the full image space into 12 sectors by angle, then draw 5 concentric circles with a_i as center point and the power of 2 as radius. Thus, the full image can be divided into 60 bins. Next, we count the number of boundary points within each bin to form the shape context. All points falling in different bins forms different relative vectors, which becomes the shape context of the point a_i. Then we compute $T_{i,k}$ to indicate the set of points, namely vector $\overrightarrow{a_i a_j}$ in $bin(k)$, selecting a_i as the referenced point,

$$T_{i,k} = \{a_j | a_j \neq a_i, (\overrightarrow{oa_j} - \overrightarrow{oa_i}) \in bin(k)\}. \tag{14}$$

Let $h_i(k) = |T_{i,k}|$ represent the number of points in $T_{i,k}$, thus the shape context $h_i = \{h_i(k)\}(k = 1, 2, \cdots, 60)$.

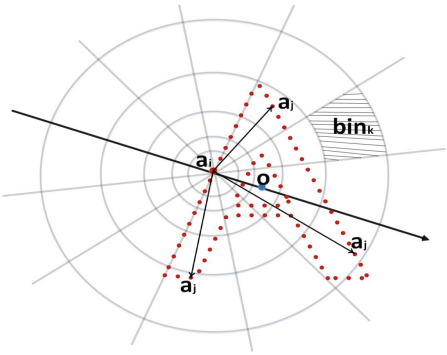

Fig. 2. The description of shape context

Privacy-Preserving Image Matching. In this subsection, a privacy-preserving image matching protocol based on shape context descriptor is proposed, which comprises the following three algorithms **SCGen**, **ImgMatch** and **ImgDec**. The details are presented as follows.

(1) $\{\{[h_x(k)]\},\{[h_t(k)]\}\} \leftarrow \mathbf{SCGen}([\mathbf{I}_i],[\hat{\mathbf{I}_q}])$ To generate the encrypted shape context for each sample point in $[\mathbf{I}_i]$ and $[\hat{\mathbf{I}_q}]$. Without loss of generality, we assume that the point $s_{q,t}$ is the pole in shape of image $[\hat{\mathbf{I}_q}]$ and the point $s_{i,x}$ is the pole in shape of image $[\mathbf{I}_i]$, and all $s_{q,t}$, $s_{i,x}(t,x=1,2,\cdots,n)$ are in the edge of shapes in each image, then we connect the pole $s_{q,t}$ with the center point o_q of shape in image $[\hat{\mathbf{I}_q}]$, the pole $s_{i,x}$ with the same point o_q respectively and divide the full image space into 60 bins, by adopting the method referred in **Shape Context in Plaintext Domain** part. Then the cloud counts how many points are located in $bin(k)\,(k=1,2,\cdots,60)$ in each shape respectively. Here, we mainly focus on generating the shape context of point $s_{i,x}$ in image $[\mathbf{I}_i]$. Each $bin(k)$ is fixed by two angles (θ_k,θ_{k_1}) and two polar radius (ρ_k,ρ_{k_1}), where $\theta_{k_1}>\theta_k$ and $\rho_{k_1}>\rho_k$. To determine whether an encrypted point $[s_{i,x'}]=([\rho_{x'}],[\theta_{x'}])$ is located in $bin(k)$, the cloud adopts the following modified privacy-preserving comparison operations presented in our proposed privacy-preserving image denoising protocol PPID. If the point $[s_{i,x'}]=([\rho_{x'}],[\theta_{x'}])$ is in $bin(k)$, it simultaneously satisfies the following four conditions:

(a)$\rho_{x'}>\rho_k$, returning a final result $[h_{0,n}^{x',k,1}]$; (b)$\rho_{x'}\leqslant\rho_{k_1}$, returning $[1-h_{0,n}^{x',k,2}]$;

(c)$\theta_{x'}>\theta_k$, returning a final result $[h_{0,n}^{x',k,3}]$; (d)$\theta_{x'}\leqslant\theta_{k_1}$, returning $[1-h_{0,n}^{x',k,4}]$.

Thus, the computation result $[h^{x'}]=[h_{0,n}^{x',k,1}]\cdot[1-h_{0,n}^{x',k,2}]\cdot[h_{0,n}^{x',k,3}]\cdot[1-h_{0,n}^{x',k,4}]$ means whether a point $s_{i,x'}$ is in $bin(k)$, and $[h_x(k)]=\sum_{x'=1}^{n}[h^{x'}]$ represents the encrypted number of points in $bin(k)$. Thus, all $\{[h_x(k)]\}(k=1,2,\cdots,60)$ constitutes the shape context of point $s_{i,x}$. Similarly, the shape context of point $s_{q,t}$ in image $[\hat{\mathbf{I}_q}]$ can be calculated as $\{[h_t(k)]\}(k=1,2,\cdots,60)$.

(2) $\{\{[h_{0,n}^{i}]\},C_3\} \leftarrow \mathbf{ImgMatch}(\{[h_x(k)]\},\{[h_t(k)]\},[T])$ After obtaining the shape context for each point, the cloud firstly finds the most matching point among all points $s_{i,x}(x=1,2,\cdots,n)$ in image $[\mathbf{I}_i]$ for each point $s_{q,t}(t=1,2,\cdots,n)$ in $[\hat{\mathbf{I}_q}]$. The cloud computes $[cost_{t,x}]$ denoted as the encrypted matching cost between point $s_{q,t}$ and $s_{i,x}$,

$$[cost_{t,x}]=\tfrac{1}{2}\sum_{k=1}^{60}\frac{([h_t(k)]-[h_x(k)])^2}{[h_t(k)]+[h_x(k)]}, \tag{15}$$

where $[h_t(k)]$ and $[h_x(k)]$ are shape contexts at points $s_{q,t}$ and $s_{i,x}$, respectively.

Given the set of cost $[cost_{t,x}]$ between point $s_{q,t}$ on the query image and all points $s_{i,x}$ on the database images, the cloud need find the minimum matching cost for $s_{q,t}$ in $[\hat{\mathbf{I}_q}]$ in encrypted domain. Thus, the cloud adopts a

modified privacy-preserving comparison operation presented in our proposed privacy-preserving image denoising protocol PPID as follows: A variant $f_{x,x'} = [1 - h_{0,n}^{x,x'}](x, x' = 1, 2 \cdots, n)$ is defined as the comparing result between $[cost_{t,x}]$ and $[cost_{t,x'}]$, where $h_{0,n}^{x,x'}$ is the tag showing the whether $cost_{t,x}$ is larger than $cost_{t,x'}$. To find the minimum matching cost with point $s_{q,t}$, the cloud computes $[cost_{t,x}]^{min} = \sum_{x=1}^{n}(\prod_{x'=1}^{n} f_{x,x'})[cost_{(t,x)}]$. Then the cloud can minimize the total encrypted minimum matching cost for each encrypted database image $[\mathbf{I}_i]$,

$$[cost_i] = \sum_{t=1}^{n}[cost_{t,x}]^{min}. \tag{16}$$

The cloud obtains N such encrypted matching cost $\{[cost_i]\}(i = 1, 2, \cdots, N)$ and compares them with the threshold $[T]$ by executing the same comparison algorithm mentioned in denoising part, generating the encrypted comparing results $\{[h_{0,n}^i]\}(i = 1, 2, \cdots, N)$. Finally, the cloud computes $C_3 = H_1(\bigcup_{i=1}^{N}[h_{0,n}^i] \,\|\, C_{ram,\rho}^{add} \,\|\, C_{ram,\rho}^{mul} \,\|\, C_{ram,\theta}^{add} \,\|\, C_{ram,\theta}^{mul} \,\|\, C_{ram,\rho}^{',add} \,\|\, C_{ram,\rho}^{',mul} \,\|\, C_{ram,\theta}^{',add} \,\|\, C_{ram,\theta}^{',mul})$ and returns it with $\{[h_{0,n}^i]\}(i = 1, 2, \cdots, N)$ to the user.

(3) $\{h_T^i\} \leftarrow$ **ImgDec**$(\{[h_{0,n}^i]\}, C_3, [\mathbf{P}], [\mathbf{q}], SK_f)$ After receiving the final encrypted comparison results $\{[h_{0,n}^i]\}(i = 1, 2, \cdots, N)$, the authorized user performs algorithm **PPOIM.Dec**$(\cdot)$ as follows. The user firstly decrypts $p \,\|\, h = f_{SK_f}^{-1}(C_{1,1})$ by using the secret key SK_f, and checks whether all of $C_{ram,\rho}^{add} = H_0(p \,\|\, h \,\|\, \bigcup_{t=1}^{n} C_{2,\rho_{i,t}})$, $C_{ram,\rho}^{mul} = H_0(p \,\|\, h \,\|\, \bigcup_{t=1}^{n} C_{3,\rho_{i,t}})$, $C_{ram,\theta}^{add} = H_0(p \,\|\, h \,\|\, \bigcup_{t=1}^{n} C_{2,\theta_{i,t}})$, $C_{ram,\theta}^{mul} = H_0(p \,\|\, h \,\|\, \bigcup_{t=1}^{n} C_{3,\theta_{i,t}})$, $C_{ram,\rho}^{',add} = H_0(p \,\|\, h \,\|\, \bigcup_{t=1}^{n} C_{2,\rho_t})$, $C_{ram,\rho}^{',mul} = H_0(p \,\|\, h \,\|\, \bigcup_{t=1}^{n} C_{3,\rho_t})$, $C_{ram,\theta}^{',add} = H_0(p \,\|\, h \,\|\, \bigcup_{t=1}^{n} C_{2,\theta_t})$, $C_{ram,\theta}^{',mul} = H_0(p \,\|\, h \,\|\, \bigcup_{t=1}^{n} C_{3,\theta_t})$, $C_3 = H_1(\bigcup_{i=1}^{N}[h_{0,n}^i] \,\|\, C_{ram,\rho}^{add} \,\|\, C_{ram,\rho}^{mul} \,\|\, C_{ram,\theta}^{add} \,\|\, C_{ram,\theta}^{mul} \,\|\, C_{ram,\rho}^{',add} \,\|\, C_{ram,\rho}^{',mul} \,\|\, C_{ram,\theta}^{',add} \,\|\, C_{ram,\theta}^{',mul})$ hold. If not, this algorithm outputs $\perp$; otherwise, the user continues to compute $q = N^{'}(ph)^{-1}$, $N^{"} = pq$ and

$$\begin{aligned} C_{T,p}^i &= ([h_{0,n}^i] \bmod N^{"}) \bmod p = H_{T,p}^i \bmod p, \\ C_{T,q}^i &= ([h_{0,n}^i] \bmod N^{"}) \bmod q = H_{T,q}^i \bmod q. \end{aligned} \tag{17}$$

Then the user can decipher the matching results $h_{0,n}^i(i = 1, 2, \cdots, N)$ by exploiting the Chinese Remainder Theorem (CRM) as follows,

$$h_{0,n}^i = h_p' q H_{T,p}^i + h_q' p H_{T,q}^i \bmod N^{"} \tag{18}$$

where h_p', h_q' respectively satisfies $h_p' q \equiv 1 \bmod p$, $h_q' p \equiv 1 \bmod q$ which can be efficiently computed since the greatest common divisor of p and q namely $gcd(p, q) = 1$. If the final result $h_{0,n}^i = 1(i = 1, 2, \cdots, N)$, the image $\mathbf{I}_i$ corresponding to this result matches $\mathbf{I}_q$; Otherwise, it means that the matching cost is larger than T, and $\mathbf{I}_i$ mismatches $\hat{\mathbf{I}_q}$.

It is noted that the algorithms **PPOIM.Enc**$(\cdot)$ and **PPOIM.Dec**$(\cdot)$ preserve the fully homomorphic property, by supporting the mixed operations (i.e. the addition and multiplication operations) on ciphertexts of polar coordinates of both the database images and the query image, namely $[\rho_{i,t}], [\theta_{i,t}], [\rho_t], [\theta_t] (i = 1, 2, \cdots, N; t = 1, 2, \cdots, n)$, that are required in our PPOIM. All the additive and multiplicative blinding factors $U_{i,t}^{add}, U_{i,t}^{mul}, U_t^{add}, U_t^{mul}$ can be cancelled out after modular $N^{"}$ in **PPOIM.Dec**$(\cdot)$ and the original image matching result would be successfully recovered. The correctness of our proposed PPOIM can be straightforwardly derived from the protocol descriptions presented above.

4 Security Proof

In this section, we give the formal security proof of our proposed PPOIM in the aspects of image privacy and matching result privacy.

Theorem 1: (**Image Privacy**) The database image privacy is unconditionally-secure (information theoretic secure) against the collusion between the cloud and malicious users, namely $H(\rho_{i,t}|[\rho_{i,t}]) = H(\rho_{i,t})$ and $H(\theta_{i,t}|[\theta_{i,t}]) = H(\theta_{i,t})$ where $H(\cdot)$, $H(\cdot|\cdot)$ respectively refer to the entropy function and the conditional entropy function. The unconditional security of query image privacy can be achieved in the same way.

In our PPOIM, the cloud and malicious users not holding secret key SK_f cannot invert the one-way trapdoor permutation f from $C_{1,1}$ generated by **PPO IM.Enc**$(\cdot)$ in Eq. (1) to derive p, q, which are adopted to encrypt each database image $\mathbf{p}_i = (\rho_{i,t}, \theta_{i,t})$. Moreover, the uniformly distributed randomnesses $U_{i,t}^{add}, U_{i,t}^{mul}$ are adopted to further blind $\mathbf{p}_i$ to guarantee the unconditional security of database image privacy. The proof details are referred to the full paper.

Theorem 2: (**Matching Result Privacy**) Let $\mathcal{A}$ be a malicious adversary defeating the matching result privacy of our proposed PPOIM with a non-negligible advantage defined as $\epsilon^{'}{}^{,n(\lambda)}$, where $n(\lambda)$ refers to the total number of queries made to the oracles and λ is the security parameter. There exists a simulator $\mathcal{B}$ who can use $\mathcal{A}$ to invert the one-way trapdoor permutation with the non-negligible probability $\epsilon \geq \epsilon^{'}{}^{,n(\lambda)} - \frac{n(\lambda)}{2^{\lambda-1}}$. In our proposed PPOIM, the matching result privacy is achieved since only the authorized user possessing the secret key SK_f can decrypt $p \,\|\, h = f_{SK_f}^{-1}(C_{1,1})$, compute $q = N^{'}(ph)^{-1}$, and recover the image matching result $h_{0,n}^i$ by Eqs. (17) and (18). in **PPOIM.Dec**$(\cdot)$. The proof details are referred to the full paper.

5 Performance Evaluation

In this section, we evaluate the performance of our proposed PPOIM in the aspects of computational overhead, communication overhead and image matching accuracy. We conduct the extensive evaluation to demonstrate the performance of our proposed PPOIM on the MPEG-7 shape silhouette database [22]

in the aspects of computational cost, communication cost on the data owner, the cloud and the user's ends, and the image matching accuracy. All our experiments are implemented by exploiting MIRACLE library [20] on a Windows 10 with Intel Core i5-7400 CPU 3.00GHz. The performance is analyzed by an efficiency comparison between our proposed PPOIM and the privacy-preserving shape context based image matching protocol exploiting public key FHE [8], [15]. Let the security parameter be $\lambda = 512$. In our proposed PPOIM, we respectively set $|p| = |q| = |h| = 512$, and the one-way trapdoor permutation implemented by RSA on $\mathbb{Z}_{N"}$ where $|N"| = 1024$-bit long. Figures 3, 4 and 5 studied the computational cost under the parameters: the number of database images N, the sampled points in each image n and the threshold t' for power series expansion. Figure 3 demonstrates that the computational cost on the data owner's end of our proposed PPOIM is dramatically lower than [8]. The reason is that [8] requires to execute public key FHE on each sampled point of all database images, namely $O(Nn)$ times in total; while in our PPOIM, the one-way trapdoor permutation, implemented by RSA and the computational cost of which is much less than public key FHE, is required to perform only once to encrypt batch of sampled points. Figure 4 demonstrates the computational cost on the cloud's end of our PPOIM is considerably less than [8], owing to the fact that Brakerski's public key FHE adopted in [8] requires to perform $O(N_m^2)$ multiplications for a ciphertext multiplication where N_m denotes the number of ciphertext components. Additionally, N_m would increase by one every time a ciphertext multiplication is needed for image denoising and matching in the encrypted domain. On the contrary, multiplication is required to perform only once every time a ciphertext multiplication is needed in our PPOIM. Figure 5 illustrates that the computational cost on the user's end is significantly lower than [8], since the decryption of Brakerski's public key FHE [15] requires the inner product composed of $O(N_m)$ multiplications; while in our PPOIM the multiplication complexity for decryption is $O(1)$.

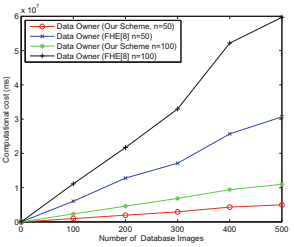

Fig. 3. Computational cost comparison on data owner's end

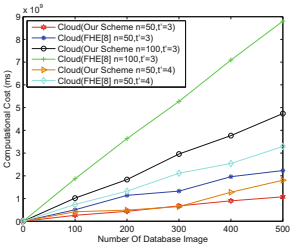

Fig. 4. Computational cost comparison on cloud's end

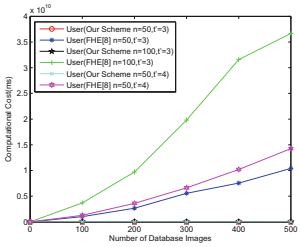

Fig. 5. Computational cost comparison on user's end

Figures 6, 7 and 8 show that the communication cost of our PPOIM are dramatically reduced no matter at the data owner, the cloud and the user's ends under the parameters N, n, t', and the number of LSH functions l, owing to

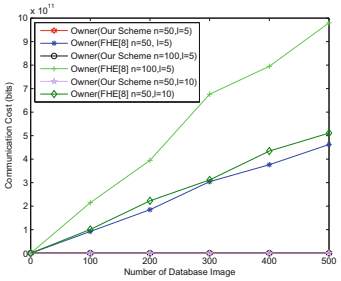

Fig. 6. Communication cost comparison on data owner's end

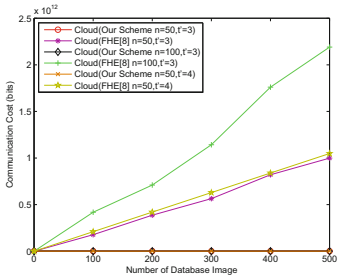

Fig. 7. Communication cost comparison on cloud's end

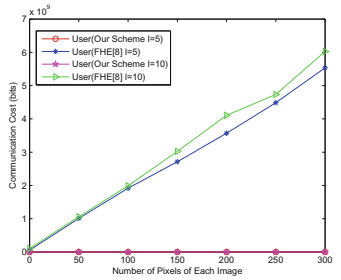

Fig. 8. Communication cost comparison on user's end

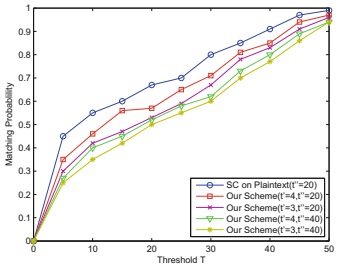

Fig. 9. Image matching accuracy comparison

the same fact that each ciphertext multiplication in [15] would incur an additional ciphertext component, leading to a high communication cost. Figure 9 demonstrates that the image matching accuracy of our PPOIM in the encrypted domain is only slightly lower than the corresponding protocol in plaintext without affecting its availability. It is observed that the matching probability increases as the threshold $t^{'}$ of power series expansion and the threshold of matching cost T increase, since the approximate integers adopted to evaluate the encrypted squared distance $d^2([\mathbf{p}_i], [\mathbf{q}])$ in Eq. (7) and the encrypted normalizing factor $[Z]$ in Eq. (12) would be more accurate, and more database images would match the queried one. The matching probability also increases as the threshold $t^{"}$ for obtaining the candidate patches for denoising decreases, since more precise patches are found to recover the original clean image more accurately in the encrypted domain.

6 Conclusion

In this paper, a privacy-preserving shape context based image denoising and matching protocol PPOIM with efficient outsourcing is proposed. Firstly, to improve the accuracy of image matching, a privacy-preserving image denoising

scheme PPID is proposed without exploiting public key FHE. Then, based on PPID, a privacy-preserving image matching adopting shape context descriptor is devised. Formal security proof and extensive simulations demonstrate the efficiency and practicability of our proposed PPOIM.

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China under Grant 61602180, 61632012, 61672239 and U1636216, and in part by Natural Science Foundation of Shanghai under Grant 16ZR1409200.

References

1. Weng, L., Amsaleg, L., Morton, A., Marchand-Maillet, S.: A privacy-preserving framework for large-scale content-based information retrieval. IEEE Trans. Inf. Forensics Secur. **10**(1), 152–167 (2015)
2. Ferreira, B., Rodrigues, J., Leitao, J., Domingos, H.: Practical privacy-preserving content-based retrieval in cloud image repositories. IEEE Trans. Cloud Comput. (2017). https://doi.org/10.1109/TCC.2017.2669999
3. Zhang, L., Jung, T., Feng, P., Liu, K., Li, X.-Y., Liu, Y.: PIC: enable large-scale privacy preserving content-based image search on cloud. In: Proceedings of IEEE ICPP, pp. 949–958, September 2015
4. Hsu, C.Y., Lu, C.S., Pei, S.C.: Image feature extraction in encrypted domain with privacy-preserving SIFT. IEEE Trans. Image Process. **21**(11), 4593–4607 (2012)
5. Hu, S., Wang, Q., Wang, J., Qin, Z., Ren, K.: Securing SIFT: privacy-preserving outsourcing computation of feature extractions over encrypted image data. IEEE Trans. Image Process. **25**(7), 3411–3425 (2016)
6. Wang, Q., Hu, S., Ren, K., Wang, J., Wang, Z., Du, M.: Catch me in the dark: effective privacy-preserving outsourcing of feature extractions over image data. In: Proceedings of INFOCOM, pp. 1170–1178 (2016)
7. Xia, Z., Wang, X., Zhang, L., Qin, Z.: A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing. IEEE Trans. Inf. Forensics Secur. **11**(11), 2594–2608 (2017)
8. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell. **24**(4), 509–522 (2002)
9. Zheng, Y., Cui, H., Wang, C., Zhou, J.: Privacy-preserving image denoising from external cloud databases. IEEE Trans. Inf. Forensics Secur. **12**(6), 1285–1298 (2017)
10. Chan, S.H., Zickler, T., Lu, Y.M.: Monte Carlo non-local means: random sampling for large-scale image filtering. IEEE Trans. Image Process. **23**(8), 3711–3725 (2014)
11. Buades, A., Coll, B., Morel, J.-M.: A non-local algorithm for image denoising. In: Proceedings of IEEE CVPR, pp. 60–65, June 2005
12. Zhou, J., Cao, Z., Dong, X., Lin, X.: PPDM: a privacy-preserving protocol for cloud-assisted e-healthcare systems. IEEE J. Sel. Top. Signal Process. **9**(7), 1332–1344 (2015)
13. Gentry, C., Halevi, S., Vaikuntanathan, V.: i-hop homomorphic encryption and rerandomizable Yao circuits. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 155–172. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_9

14. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_2
15. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_29
16. Liu, X., Choo, R., Deng, R., Lu, R., Weng, J.: Efficient and privacy-preserving outsourced calculation of rational numbers. IEEE Trans. Dependable Secur. Comput. **15**(1), 27–39 (2018)
17. Liu, X., Qin, B., Deng, R., Li, Y.: An efficient privacy-preserving outsourced computation over public data. IEEE Trans. Serv. Comput. **10**(5), 756–770 (2017)
18. Taeho, J., Mao, X., Li, X.: Privacy-preserving data aggregation without secure channel: Multivariate polynomial evaluation. In: Proceedings of IEEE INFOCOM, pp. 2634–2642 (2013)
19. Damgard, I., Geisler, M., Kroigard, M.: Homomorphic encryption and secure comparison. Int. J. Appl. Cryptogr. **1**(1), 22–31 (2008)
20. Multiprecision integer and rational arithmetic C/C++ library. http://www.shamus.ie/
21. Naehrig, M., Lauter, K., Vaikuntanathan, V.: Can homomorphic encryption be practical? In: Proceedings of CCSW, pp. 113–124 (2011)
22. Jeannin, S., Bober, M.: Description of core experiments for MPEG-7 motion/shape. RIM 2003. Hrvatska znanstvena bibliografija i MZOS-Svibor (2003)
23. Ghinita, G., Rughinis, R.: An efficient privacy-preserving system for monitoring mobile users: making searchable encryption practical. In: Proceedings of ACM CODASPY, pp. 321–332 (2014)
24. Tang, Q., Wang, J.: Privacy-preserving context-aware recommender systems: analysis and new solutions. In: Pernul, G., Ryan, P.Y.A., Weippl, E. (eds.) ESORICS 2015. LNCS, vol. 9327, pp. 101–119. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24177-7_6
25. Veugen, T.: Encrypted integer division and secure comparison. Int. J. Appl. Cryptogr. **3**(2), 166–180 (2014)