Outsource Photo Sharing and Searching for Mobile Devices With Privacy Protection

Lan Zhang∗, Taeho Jung †, Cihang Liu *, Xuan Ding *, Xiang-Yang Li†, Yinhao Liu*
∗ School of Software, Tsinghua University
† Department of Computer Science, Illinois Institute of Technology

Abstract—With the proliferation of mobile devices, cloud-based photo sharing and searching services are becoming common due to the mobile devices’ resource constrains. Meanwhile, there is also increasing concern about privacy in photos. In this work, we present a framework SouTu, which enables cloud servers to provide privacy-preserving photo sharing and search as a service to mobile device users. Privacy-seeking users can share their photos via our framework to allow only their authorized friends to browse and search their photos using resource-bounded mobile devices. This is achieved by our carefully designed architecture and novel outsourced privacy-preserving computation protocols, through which no information about the outsourced photos or even the search contents (including the results) would be revealed to the cloud servers. Our framework is compatible with most of the existing image search technologies, and it requires few changes to the existing cloud systems. The evaluation of our prototype system with 31,772 real-life images shows the communication and computation efficiency of our system.

I. INTRODUCTION

With the increasing population of smart personal devices (e.g., smartphone, tablet PC) as well as the emergence of wearable devices (e.g., Google Glass), huge amounts of photos are produced everyday. The data volume of photos is growing exponentially due to the high-resolution on-board cameras, and this makes the photo management and sharing challenging to mobile devices. Facing such challenge, users often choose to outsource the burdensome image storage and searching to cloud servers such as Amazon Cloud Drive, Dropbox and some image-oriented cloud (Cloudinary). Various social networking systems (Flickr, Facebook, Google Plus etc.) also provide photo sharing services for personal uses.

However, privacy becomes a critical issue when photos are outsourced to third parties. For example, the image recognition technique introduced by Facebook was very controversial in 2011, because the objects in users’ photos such as faces and cars can be automatically recognized [1], [2]. And tracking and sucking become easier with various image search engines (e.g., Google Image Search, Yahoo! Image Search). This controversy finally made Facebook to switch off its face recognition service in 2012. But it has brought back the functionality recently due to the need for image search. To handle such privacy issues, Google has decided to forbid the face recognition functionality on Google Glasses. Many users are still struggling to disable face recognition in Facebook. Simple privacy policy enforced by some companies cannot guarantee the privacy protection, especially when the search can be automatically conducted by a machine. Considering the rich sensitive information (e.g., people, location and event) embedded in the photos, privacy in images is in urgent need of protection.

To some extent, the above privacy concerns come from the fear that our photos might be illegally searched by a malicious hacker, and this is probably one of the primary reasons why users want to get rid of face recognition. However, the object recognition techniques could bring powerful ability to image search, and we believe simply disabling automatic recognition is not the best solution to the privacy problem because it also eliminates the potential utilities lying in the image search functionality. Ideally, privacy-sensitive users should have an option to use the secure version of photo sharing and searching system with little extra overhead, in which image search with object recognition is allowed for authorized users but the privacy leakages due to automatic recognition are prevented.

To achieve this vision, in this paper, we first design a framework SouTu, allowing mobile device users to enjoy photo sharing with fine-grained privacy protection policies, which can be provided by any cloud to attract privacy-seeking users. Via our framework, an owner can share his photos in a cloud without unintended access to his photos, and an authorized querier can send photo queries to conduct image search on others’ photos. To allow resource-bounded mobile devices to search a huge volume of photos, SouTu outsources the heaviest computation and storage tasks to the cloud, and only the private parts of photos are selectively protected to further reduce the computation and communication overhead. Despite such outsourcing, SouTu does not reveal the private image contents or the query contents (including its result) to the cloud. In the later section, to more aggressively enhance the performance, we further introduce the optimized SouTu where the computation overhead is reduced by half with only a little loss of accuracy.

SouTu can be considered as a step towards easily deployable frameworks for privacy-preserving photo sharing and searching services among mobile device users, taking advantage of the availability of cloud servers who possess powerful computation and storage abilities.

In summary, our contributions can be summarized as:

1. We propose a novel modularized photo sharing and searching framework to let mobile device users outsource photos and the majority of heavy jobs (storage, access control and searching) to cloud servers, without breaching users’ photo-related privacy.
2. We design two outsourced vector distance computation
protocols for both real and binary vectors, which are the
core of the framework. Different from the existing multi-
party computation based methods, our protocols enable
efficient vector distance computation in a non-interactive
way, which means the photo owner does not have to interact
with the cloud or the querier.
3. Our framework is compatible with common image services
and feature based image search methods. To achieve nice
user experience, all privacy protection modules work auto-
matically and are transparent to users. After the privacy
setting, users can enjoy the sharing and searching services
as usual. We implement and evaluate our framework using
31,772 real-life images on both smartphones and laptops.
The evaluation shows that very low extra overhead is
incurred by our method.

II. BACKGROUNDs AND MotivATIOn

One of our main contributions is enabling efficient photo
sharing and searching on encrypted photos. In order to achieve
high search accuracy, Soutu leverages the state-of-the-art
image search technologies in computer vision field. Here, we
briefly review the search techniques, and then discuss the
privacy issues emerging from them in this section.

A. Descriptors Based Image Search

Images are usually searched by their contents. Different
types of visual descriptors are proposed to model the visual
characteristics of the image, e.g., color, intensity, texture
or objects within the image. Various image contents can be
recognized and localized (e.g., people [3] and face [4])
using visual descriptors. Among these, human face detection
received extraordinary attention and is one of the most mature
object detection techniques so far [4, 5].
The feature descriptor is usually constructed as a set of
numeric vectors, denoted as feature vectors. There are some
statistical feature vectors (e.g., intensity/color histograms).
Also, many well designed visual descriptors are proposed to
achieve accurate image search, e.g., SIFT [11] and SURF [2]. In
those works, each feature vector is generated from an interest
point of the image to describe the visual characteristics around
the point. Interest points are pixels containing distinguishing
information of the image [6]. In general, all feature vectors
belonging to the same descriptor have the same dimension
(e.g., SIFT has 128 dimensions and SUFR has 64 dimensions).
The numeric type of vectors may be real number [1, 2] or
binary [7, 8], and different types of vectors are used for
different applications. Specifically, with a little accuracy loss,
binary descriptors are usually more efficient in computation
and suitable for resource-restricted mobile applications. We
design our framework capable of dealing with both real
number and binary feature vectors.

Given a query image, one needs following three steps to
search the top-k similar images from the database. Firstly, pre-
defined image descriptor is extracted from the query image.
Secondly, each feature vector in the query image is compared
with feature vectors from the database images. Thirdly, simi-
larity score for every database image is measured based on
the vector comparison and finally the top-k high-score images
are returned to the querier.

B. Privacy Implications

Rich content of photos raises various privacy implications.
There are many mature techniques to detect and recognize
the objects within the photos as aforementioned. These tech-
niques can possibly be used to automatically analyze the
photos to mine sensitive information with various data mining
techniques. Combining the location stamps and time stamps
embedded in a photo, more sensitive information about the
person may be derived (e.g., home location, occupation, level
of incoming). Therefore, the private part (denoted as Region
Of Privacy (ROP) hereafter) of a photo needs to be protected,
so that no human or machine runnable algorithm can learn
sensitive information in the photo.

Besides the outsourced photos, the query sent to the cloud
side incurs privacy implications as well. Even though the
uploaded photos are well protected via encryption so that the
cloud does not gain useful information of them, their contents
can be easily deduced if the queries’ contents and results are
revealed to the cloud. Since the entire search process should be
outsourced to the cloud for resource saving, protecting queries’
contents as well as the results is equally important to protecting
the uploaded photos.

III. SYSTEM OVERVIEW

Soutu is a novel framework allowing clouds to provide
privacy preserving photo sharing and searching services to
the mobile devices users. It can attract users who need both
outsourced photo service and privacy protection. Figure 1
illustrates the architecture and workflow of our framework.

With this framework design, Soutu can provide the follow-

ingservices: (1)privacy-preserving photo storage outsourcing;
(2)fine-grained photo sharing with privacy protection enforce-
ment; (3)light-weight photo searching for mobile devices.

A. Privacy Preserving Photo Storage

Before users upload their photos to cloud servers for shar-
ing, the photos need to be pre-processed. Firstly, the region
of privacy (ROP), which is a rectangle defined by two pixel-
level coordinates (top-left and bottom-right) on the photo, is
either automatically or manually determined. In the automatic
manner, the user can define a category of objects as private
content, e.g., faces. Then all private objects will be detected
by object recognition algorithm and set as ROPs, e.g., the face in
Fig. 2. Otherwise, the owner can also manually define the ROP
by selecting a rectangle region on the photo. Then, the feature
vectors of the ROP are extracted according to the definition
of the image descriptor (Section II-A). Note that, hereafter we
use the human faces as example ROPs of photos in this work,
but other objects such as pedestrians and cars, can also be
defined as ROPs with corresponding recognition algorithms.
Moreover, except defining ROP, all the following operations

2We logically divide the cloud into sharing cloud and search cloud for
explanation purpose, but revealing this structure does not breach users’
privacy at all.
are conducted automatically by the system and transparent to users.

After the ROP is selected, it is separated into public part and secret part, where the public part doesn’t contain any sensitive information and the secret part is encrypted such that only the authorized users with keys can access to it and recover the original ROP. We review the following three different methods for the separation:

1. **Mask**: fills public part of ROP with solid black (all intensity values ‘0’) and takes the original ROP as secret part.

2. **P3** [9]: separates ROP based on a threshold in the DCT frequency domain; sets the higher frequency part as secret part and the remaining as public part.

3. **Blur** [10]: a normalization box filter is applied to ROP to generate the public part; subtracts the public part from ROP in a pixel-wise way to achieve the secret part.

Then, the public part of the whole photo is produced by replacing its ROP with the public part of ROP (as shown in Fig. 2). Our experiment (Section VI) shows that all three methods are resistant to automatic detection algorithms, but the blur based method outperforms others in the storage cost, hence we adopt the blur as the default separation method in SouTu. After extracting the private part from ROP, the owner encrypts the secret part as well as its image descriptor as a private bag, and uploads the private bag to the search cloud. Then, he also uploads the public part of the original photo as a public bag to the sharing cloud (Fig. 1(a)).

**B. Fine-grained Photo Sharing**

SouTu allows fine-grained photo sharing among users. The photo owner uses an access control scheme (e.g., [11], [12]) to encrypt the search keys so that only the authorized users with certain attributes can obtain search keys. As the Step 5 in Fig. 1(a) the owner encrypts the search keys under the access rule that he defines, and the encrypted search keys are uploaded to the sharing cloud and made published. Obtaining the search key, the authorized user can generate valid photo queries and decrypt the private part of ROP. The completed original images can be recovered simply by merging the public parts of images and the private parts of ROPs. Here, all these operations are also automatic and transparent to users, and the authorized user can browse the shared images as usual.

**C. Light-weight Photo Searching**

When a querier wants to search a photo among someone else’s photos, he pre-processes the querying photo to achieve the corresponding image descriptor. Then, only if satisfying the owner’s access rule, he can retrieve the search keys to search the owner’s photos, but it is the cloud who conducts the searching job and returns the result to the querier obliviously, i.e. without knowing contents of the owner’s ROPs or the contents of the query photo. After fetching the query result, as mentioned above, system generate the original image for the querier transparently. For the querier, the whole system appears like common image search systems. Fig. 1(b) illustrates the search procedure.

**D. System Design Goals**

Our system is designed to achieve efficiency, privacy protection and accuracy goals.

- **Efficiency**: To overcome the resource limitation of mobile client, operations at the user side should be light-weight, and most of the expensive computations should be outsourced to the cloud side.

- **Privacy Preservation**: Users outsource not only the storage of photos but also the searching to the cloud side in SouTu. Therefore, the framework is expected to protect users’ privacy in various aspects:
  1. **ROP Privacy**: Unauthorized party should not learn secret part of ROP including cloud servers,
  2. **Query Privacy**: Cloud servers should not learn query photos,
  3. **Result Privacy**: Cloud servers should not learn search results, which are all non-trivial challenges since cloud servers
are the party who conducts the searching jobs on the photos stored at his side.

- **Accuracy**: Introducing the privacy protection mechanism should not bring much accuracy loss. That is, the search result from SouTu should be comparable with traditional image search technologies conducted on plain texts of photos.

### E. Threat Model

W.l.o.g., we assume curious-but-honest cloud servers and malicious users in this work. Cloud servers will follow the protocol specification in general, but they will try their best to harvest useful information from the protocol communications in order to infer the secret part of ROP or the contents of queries, which is sensitive information to be protected. On the other hand, queriers may misbehave throughout the protocol to infer the search keys to forge a valid photo query, where the search keys are supposed to be kept secret as well.

### IV. SYSTEM DESIGN

In this section, we first present the building blocks of our system, and then give the detail of our non-interactive private image search protocol, which is the core of the system and one of our main contributions.

#### A. Building Blocks of Our System

SouTu is a modularized and well integrated image sharing and searching system, which consists of several building blocks.

1) **Image Search**: Image search is composed of three steps: image descriptor extraction, finding matching vector and similarity score calculation.

**Extracting Image Descriptor**. As described in Section III, the visual descriptor is extracted from the interest points of each photo, where the interest points are automatically detected (e.g., [6]). Then, the descriptor $X = \{x_1, x_2, \ldots\}$ of an image $I_x$ is extracted, where $x_i$ is a feature vector.

**Matching Feature Vector**. Given a feature vector $x \in X$ and another descriptor $Y$, let $d(x, y)$ be Euclidean distance between two feature vectors $x \in X$ and $y \in Y$. Then given the $x$’s nearest neighbor $y_{nn} \in Y$, $x$ and $Y$ are a matching pair iff:

$$\delta(x, Y) = \frac{d(x, y_{nn})}{\min_{y \in Y - \{y_{nn}\}} (d(x, y))} < \alpha$$

That is, iff the ratio between nearest distance and the second nearest distance is less than a threshold $\alpha$, $x$ and $Y$ are a matching pair. For most object recognition algorithms, $\alpha$ is set as $0.5$.

**Similarity Score**. Given a querying descriptor $X$ and a queried descriptor $Y$, the similarity score between $X$ and $Y$ are defined as the number of matching pair $X$ has, i.e.,

$$S(X, Y) = \sum_{x_i \in X, \delta(x_i, Y) < \alpha} 1$$

Given a querying image, matching images with high similarity score can be searched in a database.

2) **Cryptographic Tools**. Our system also takes advantage of rich cryptographic algorithms for privacy protection in cloud-based image search. It includes: homomorphic encryption, attribute based encryption and oblivious transfer.

**Homomorphic Encryption**. We employ Paillier’s cryptosystem [13] as a building block which has the following homomorphism:

$$\text{HE}.E(m_1)\cdot \text{HE}.E(m_2) = \text{HE}.E(m_1 + m_2)$$

and

$$\text{HE}.E(m_1)^m = \text{HE}.E(m_1m_2),$$

where $\text{HE}.E(m)$ denotes the ciphertext of $m$. Paillier’s cryptosystem is proven to be semantically secure against chosen plaintext attack (SS-CPA), which implies that any ciphertext of any message is indistinguishable to a randomly chosen element among the ciphertext space.

Note that the numeric type of feature vectors may be real number, but the Paillier’s cryptosystem is based on large integers, therefore we need to use integers to represent real numbers first. SouTu uses the fixed point representation to represent real numbers rather than floating-point representation due to its efficiency.

**Ciphertext-Policy Attribute Based Encryption**. We also adopt ciphertext-policy attributed based encryption (CP-ABE) [11] for access control due to its generality and security. Other attribute based encryption methods can also be adopted, e.g., [12]. In the CP-ABE, a trusted authority (not the image service provider) takes response of generating public parameters. Given the public parameters, a data owner can encrypt a message such that only the users satisfying a certain access rule can decrypt it. Secret keys of users contain attribute values for the key holders, and the access rule is expressed with boolean operators (AND, OR etc.) and attribute values. CP-ABE is proven to be IND-CCA1 secure, which implies the semantic security against chosen plaintext attack.

**Oblivious Transfer** The $k$-n oblivious transfer (OT) [14] let a receiver obtain any subset of $k$ items from the sender’s $n$ items, while the sender remains oblivious of the receiver’s selection, and the receiver remains oblivious of other items as well.

#### B. System Join

Whenever a new user joins the system, he generates a pair of Pailler Keys $PK, SK$ and picks a random vector $r$, which has the same dimension of the feature vector. Then, he uses CP-ABE to encrypt $PK, SK, r$ under the access rule he wishes to enforce (i.e. who can search on his images). He uploads the following to the sharing cloud, which are the search keys to be used in the photo searching later.

$$\text{ABE.r}\{PK, SK, r \mod n\}$$

#### C. Public & Private Bag Generation

When an owner wants to upload his photo $I_x$, the ROP $R(I_x)$ is selected either automatically or manually, and the image descriptor $X$ of ROP is extracted. $X$ is a set of fixed-dimension feature vectors $X = \{x_1, x_2, x_3, \ldots\}$. A photo may have

3Computation is conducted in a finite cyclic group, and modular operations are followed. We omit the modular operations for the sake of simplicity and defer the detailed description on the finite group selection to Section V.
between feature descriptor
the top-2 nearest distances to compute the similarity scores
I
with
querier uses
specific owner’s images, he extracts corresponding image
cloud conducts several homomorphic operations to achieve the
storage of the feature vectors in the querying image as follows:
and further proceed. Next, he encodes every single dimension
D. Cloud-based Image Search

Once the querier receives the cloud server’s encrypted responses,
he can decrypt the results using his secret key and receive
the matching images.


e.g.
popular
images.



Protocol 1 Secret & Search bag generation
1: The owner of \( I_x \) randomly picks a symmetric key \( K_e \) and uses symmetric
encryption (AES in this paper) to encrypt the private part of the ROP as
\( AES \cdot E_{K_e}(S(R(I_x))) \). \( K_e \) is encrypted via CP-ABE under his privacy
policy as \( AB.E.E(K_e) \).
2: For every dimension \( x(k) \) in every vector \( x \in X \), he computes the
following homomorphic ciphertexts using his \( PK \) and \( r \):

\[
HE \cdot E \left( (x(k))^2 \right), HE \cdot E \left( -r(k) \cdot x(k) \right)
\]

After the public bag is uploaded, the owner encrypts the
private part of ROP as the private bag using symmetric
encryption such as AES-256. Also, for the cloud-based search,
he homomorphically encrypts the feature descriptor, which are
stored in the search bag (Protocol 1).

Then, the private bag and the search bag of \( I_x \) are:

\[
I_{x,pr1} = \left\{ \begin{array}{c}
AES \cdot E_{K_e}(S(R(I_x))) \\
AB.E.E(K_e)
\end{array} \right\}
\]

\[
I_{x,src} = \left\{ \begin{array}{c}
HE \cdot E \left( X^2 \right) \\
HE \cdot E \left( -r \odot X \right)
\end{array} \right\}
\]

where \( X^2 \) and \( -r \odot X \) represent the sets \( \left\{ (x_i(k))^2 \right\}_{\forall i,k} \) and
\( \left\{ -r(k) \cdot x_i(k) \right\}_{\forall i,k} \) (Hadamard product between \( -r \) and each \( x_i \)) respectively. The private/search bag are uploaded to the
sharing/search cloud respectively.

D. Cloud-based Image Search

When a querier wants to search an image \( I_y \) among a
specific owner’s images, he extracts corresponding image
descriptor \( Y \) and obtains the owner’s search keys \( AB.E.(PK, S.K, r) \) from the server. If he is authorized to search on the
owner’s images, he will successfully decrypt the search keys
and further proceed. Next, he encodes every single dimension of the feature vectors in the querying image as follows:

\[
C_1(y_j(k)) = r(k)^{-1} \cdot y_j(k)
\]

\[
C_2(y_j(k)) = HE \cdot E \left( y_j^2 \right)
\]

Consequently, the querier achieves two sets of encoded feature descriptors \( C_1(Y) \), \( C_2(Y) \) corresponding to \( I_y \). He
then sends these two sets to the cloud server to outsource the
image search. After receiving the encoded descriptors, the
cloud conducts several homomorphic operations to achieve the
encrypted pairwise distances between \( x_i \) and \( y_j \) for all \( i, j \)
with \( I_{x,src} \) in the search cloud (Protocol 2). Then, he sends all the ciphertexts of results back to the querier.

Upon receiving the ciphertexts of pairwise distances, the
querier uses \( SK \) to decrypt every \( d^2(x_i, y_j) \). Then, he finds
the top-2 nearest distances to compute the similarity scores
between feature descriptor \( X \) and every \( Y \) according to Eq. 1

E. Image Retrieval

Based on the similarity scores, the querier requests the
public bags as well as the private bags of the top-k similar images from the sharing cloud (e.g., by requesting the URLs).
However, explicit request reveals the search result to the server.
Even if every secret part of ROP is encrypted and the query
contents are well protected, cloud may infer side information
by gathering the statistics of the image retrieval (e.g., popular
images and frequently visited images). Thus, we need to hide
the retrieval pattern as well.

To achieve this requirement, we employ the \( k-n \) OT (Section
IV). Since it is extremely expensive to construct a \( k-n \) OT with a large \( n \), we do not directly run a \( k-n \) OT across
the whole database to obliviously retrieve \( k \) images. Instead,
we try to find a trade-off between privacy and performance as
follows. The querier determines a random subset \( \sigma' \subseteq DB \)
which contains the set of images \( \sigma \) that he wants to retrieve.
The sizes of \( \sigma \) and \( \sigma' \) are \( k \) and \( n \) respectively. Then, the
querier and the sharing cloud engage in a \( k-n \) OT to let the
querier obliviously select the \( k \) images.

V. Security Analysis and Refinement

A. Security Analysis

Firstly, the secret part of ROP is well protected by the
symmetric encryption, whose key is encrypted with CP-ABE
proven to be semantically secure. Besides, the search keys are
also protected by the CP-ABE. Therefore, clouds cannot infer
sensitive information from its storage in SouTu.

Then, we design a game to prove that SouTu reveals no
sensitive information to the cloud servers during the photo
searching procedure theoretically. We omit the proof here
due to space limitation, readers can refer to the appendix of ( [13])
for the detail of proof.

Besides the adversarial cloud servers, we have also assumed
malicious queries in our adversarial model. However, unauthorized malicious users are not as threatening as cloud servers
since they never get involved in any transaction with valid
users. All they can do except compromising the server is
to try man-in-the-middle attacks to sniff the search results,
but this can be trivially prevented by introducing secure
communication channel. Even if they compromised a server,
CP-ABE guarantees the indistinguishability of the ciphertexts.
In conclusion, malicious users do not learn about sensitive
information either.

B. Refinements for Binary Descriptor

Some image retrieval systems use binary image feature
descriptors because they are more compact and computationally
manageable than real number ones, with a little accuracy loss in content recognition [15]. It is more suitable for resource-limited applications. However, directly applying SouTu in mobile platforms with binary descriptors does not fully exploit the advantage of it. The exponentiation operations contribute to majority of the computation overhead in our cryptographic building blocks, but both image owners and queriers need \( \Theta(\alpha) \) exponentiations throughout the protocol where \( \alpha \) is the number of interest points in an image.

To relax this bottleneck, we further design our framework for the special case where binary descriptors are used (refer to the system as SouTuBin). Note that for any two vectors \( x, y \), we have:

\[
d^2(x, y) = \sum_k (x(k) - y(k))^2 = \sum_k x(k) \oplus y(k)
\]

where \( x(k) \) is the \( k \)-th bit of \( x \) and \( \oplus \) is the bitwise XOR operator. Therefore, we consider using a succinct garbled circuit in combination with homomorphic encryption to achieve a lightweight and non-interactive framework dedicated to binary descriptor based search, which is one of our contributions.

1) Yao’s Garbled Circuit: To enhance the understanding, we briefly review Yao’s garbled circuit (GC), and we direct the readers to relevant literal works [17] for technical details. Yao’s Garbled Circuit is designed for two-party computation, where \( P_a \) and \( P_b \) wish to jointly compute a function \( F \) over their private input \( x \) and \( y \) using a garbled boolean circuit. Here we use an XOR gate as an example.

![Fig. 3: Gate](image)

Two random values \( k_i^0, k_i^1 \) are chosen to represent the bit values 0 and 1 for each wire \( w_i \). Then, the shuffled Table I represents the garbled XOR gate (shuffled so that inputs are not inferred from the row number). Given two garbled inputs, the evaluator can obliviously evaluate the boolean gate by looking up the shuffled table and decrypting the output to get a garbled output.

2) System Join: The new joiner generates a pair of Paillier keys \( PK, SK \) and picks two symmetric encryption keys \( K^0, K^1 \) as well as a random seed \( s \). Then, he defines a privacy policy to specify which group of people are authorized to search on his images. \( PK, SK, K^0, K^1 \) and \( s \) are encrypted using CP-ABE as \( \text{ABE.E}\{\{PK, SK, K^0, K^1, s\}\} \), which are uploaded to the sharing cloud as his search keys. Finally, he uses \( PK \) to encrypt 0,1 homomorphically for later use, i.e., \( \text{HE. E}(0), \text{HE. E}(1) \).

3) Public & Private Bag Generation: To upload a photo \( I_x \), the owner extracts the ROP \( R(I_x) \) as well as the binary image descriptor \( X \), and generates the public bag as in the original framework SouTu. After uploading the public bag to the sharing cloud, he symmetrically encrypts the private part of ROP, and keeps it as well as the key in the private bag.

### Protocol 3 Secret & Search Bag Generation

1. The owner randomly picks a key \( K_e \) and uses symmetric encryption (AES in this paper) to encrypt the private part of the ROP as \( \text{AES.E}_{K_e}(S(R(I_x))) \), \( K_e \) is encrypted via CP-ABE under his privacy policy as \( \text{ABE.E}(K_e) \).

2. For every bit \( x(k) \) in every vector \( x \in X \), he generates and shuffles the following table:

| \( x(k) = 0 \) | \( x(k) = 1 \) |
|----------------|----------------|
| \( y_0 = H^k(s) \cdot K^0 \) | \( \text{AES.E}_{K_e}(\text{HE. E}(0)) \) |
| \( y_1 = H^k(s) \cdot K^1 \) | \( \text{AES.E}_{K_e}(\text{HE. E}(1)) \) |

which represents \( x(k) \)'s garbled gate \( G(x(k)) \).

Then, he uses a collision-resistant hash function \( H(.) \) and the search keys to garble each bit as a garbled gate (Protocol 3), where \( H^k(.) \) denotes applying the hash function for \( k \) times. From the protocol, the feature vector \( x \) is encrypted to a series of garbled gates (Fig. 4), and the following are corresponding private bag and search bag of \( I_x \):

| \( y(1) \) | \( y(2) \) | \( y(3) \) |
|----------|----------|----------|
| \( \text{HE. E}(y(1) \oplus y(1)) \) | \( \text{HE. E}(y(2) \oplus y(2)) \) | \( \text{HE. E}(y(3) \oplus y(3)) \) |

Fig. 4: Garbled gates \( G(x) \) from Protocol 3

4) Cloud-based Image Search: To search a photo \( I_y \) from other’s ones, the querier extracts corresponding descriptor \( Y \) and obtains the owner’s \( \text{ABE.E}\{\{PK, SK, K^0, K^1, s\}\} \). If he successfully decrypts it, he further uses \( H^k(s)K^0 \) or \( H^k(s)K^0 \) to encode each \( k \)-th bit \( y(k) \) as the garbled input \( \text{GI}(y(k)) \) to finally achieve the set of garbled inputs \( \text{GI}(Y) \), which is uploaded to the cloud. The cloud server conducts homomorphic operations to achieve \( \text{HE. E}(d^2(x_i, y_j)) \) for all \( i,j \) without interacting with the requester or the image owner (Protocol 4). Then, he sends the ciphertexts back to the querier, who proceeds as SouTu.

### Protocol 4 Privacy-preserving Distance Calculation

1. For every garbled gate \( G(x_i(k) \oplus y_j(k)) \) in \( G(x_i) \), the cloud server looks up and symmetrically decrypts \( \text{HE. E}(x_i(k) \oplus y_j(k)) \) from the shuffled table.

2. Then, he computes:

\[
\sum_k \text{HE. E}(x_i(k) \oplus y_j(k)) = \text{HE. E}(\sum_k x_i(k) \oplus y_j(k)) = \text{HE. E}(d^2(x_i, y_j))
\]

VI. IMPLEMENTATION AND EVALUATION

A. Development Environment

We implemented both client side and cloud side of SouTu. The client side program is developed for Android smartphones and the commodity laptops for performance comparisons, and
the cloud side program is developed only for the laptops. We used HTC G17 (1228Hz CPU, 1G RAM) and ThinkPad X1 (i7, 2.7GHz CPU, 4G RAM).

The CP-ABE is implemented based on the PBC library, and other building blocks (Section IV) are implemented in Java, including the AES (128-bit), Paillier’s cryptosystem (512-bit primes p, q), k-n oblivious transfer and the fixed point operations. Based on these building blocks, we implemented the core protocols in both variants SouTu and SouTu\textsubscript{bin}. The automatic ROP detection is implemented with cascade object detection (e.g., face detection) [4]. We employed widely used 64-dimensional SURF descriptor [2] and 128-dimensional SIFT descriptor [1] for the variant of real number descriptors (SouTu\textsubscript{bin}), and 64 bit binary SURF and 128 bit binary SIFT for SouTu\textsubscript{bin}. Although our evaluation is conducted with these descriptors, our system is compatible with other vector-based descriptors too. Both the object detection and descriptor extraction are implemented using the image process library OpenCV for Windows and Android. ROP separation (Mask, P3 [9], and Blur) is also implemented with it.

B. Real-life Datasets

To measure the privacy protection and the cost of SouTu, we used the well-known Labelled Faces in the Wild (LFW) dataset [13], which consists of 30,281 real life images collected from news photographs. We detect all human faces automatically and set those faces as ROPs of images, and 9 feature vectors are extracted as their image descriptor [19]. On average, SouTu occupies less than 20% of each image for 80% images. We also used the INRIA Holidays dataset [20], which contains 1,491 high-resolution personal photos taken during their holidays (majority with resolution 2560px×1920px). We set the entire image of the INRIA as the ROP.

C. Image Recognition on Public Part

ROPs are separated in three different methods (Mask, P3 and Blur) respectively. To evaluate the safety against the object detection algorithms, we ran face detection [4] and feature points detection [21] algorithms on the public part of ROPs. On average, there are 1.1 faces in each original image in LFW, but only 0.017, 0.029 and 0.028 faces are detected in the public part of the ROPs generated by Mask, P3 and Blur respectively, and our manual examination shows that majority of the detections were false positives (e.g., some textures being detected as faces). Therefore, we conclude that almost no faces are detected in the public parts of images by algorithm. Also, no matched feature points are detected in the public parts of ROPs for both LFW and Holiday datasets as well. As a conclusion, all three methods provide good privacy protection against face/feature detection algorithms.

We also compare the computation cost and storage cost of three methods. Figure 5 illustrates the CDF of run time for processing each image with three methods. On average, Mask has the minimum computation cost with 0.0002s per image in LFW Dataset, and 0.02s per image in Holiday Dataset; Blur needs 0.037s for LFW Dataset and 0.68s for Holiday Dataset; P3 needs 0.35s for LFW Dataset and 24.9s for Holiday Dataset. This result also confirms that protecting the entire image is much more expensive than protecting the sub-regions of the image. Figure 6 and 7 present the normalized storage cost of three methods for LFW and Holiday. The sizes of Blur-processed images are only 73% of the original ones in Holiday dataset on average. In conclusion, Mask and Blur outperforms P3 in computation performance while Blur has the best storage performance, therefore SouTu uses Blur as the default method.

D. Search Accuracy

In SouTu, the search procedure follows exactly the same vector-based similarity comparison as typical image search technologies (e.g., [8]). Also, the accuracy loss introduced by the fixed point representation is almost negligible (less than \(\frac{1}{\text{base}^{\text{scale}}+1}\) in each value where base is often 10 and scale is greater than 5), therefore SouTu provides a comparable accuracy as existing image search techniques.

E. Client Side Performance

1) Computation Overhead: For the photo owner, the computation overhead mainly comes from the following operations: (1) object detection and descriptor extraction; (2) ROP
it takes for mobile devices. After the querier obtains the search result, becomes negligible in existing optimized search mechanisms such as k-d tree [22].

is promising and not trivial to reduce the complexity with all photos of an owner with a linear complexity, and it non-negligible extra overhead comes from the linear search if owners have hundreds of photos on average. However, this takes dataset, if a querier searches a photo among 1,000 photos, it needs to process the search result on laptops and smartphones on average. It is slightly beyond acceptable

Table II(a) shows that protecting subregions (e.g., faces) of studied objects like faces (9 feature vectors in a descriptor), the distance between two real number feature vectors (query) is approximately 36 KB in LFW Dataset, the extra communication cost brought by SouTu or SouTu_{bin} is roughly 6% of that for system without privacy consideration. But for the Holiday Dataset, our method actually save the communication cost by 27%.

The communication overhead for uploading the encoded feature descriptors (query) is approximately 36 KB in SouTu and 9 KB in SouTu_{bin}, which are reduced to 350B and 90B respectively after compression. The communication overhead for downloading the similarity result is 128B for each compared image in the database, and the one for downloading each image is similar to the uploading overhead of the image owner. Note that, to achieve k-n oblivious transfer, the querier needs to download (n - k) extra images from the search server to hide the search pattern, where n can be specified according to the trade-off between privacy and performance.

**F. Cloud Side Performance**

On the clouds, similarly, the image storage and communication is 6% more for LFW Dataset and 27% less for Holiday Dataset. The main computation overhead is from the distance computation. We evaluated the search performance on laptops, so the actual performance when deployed in more powerful cloud servers will be significantly improved. Our privacy-preserving distance protocols take nearly 0.18s to calculate the distance between two real number feature vectors (SouTu) and only 0.018s for binary feature vectors (SouTu_{bin}). For well studied objects like faces (9 feature vectors in a descriptor), and for each owner, there are usually hundreds of images on

Thus, the scalability can be achieved using those search algorithms.

2) **Communication Overhead:** The communication overhead for the image owner mainly comes from uploading public bag and private bag to the cloud. When using Blur to separate ROPs, as presented in Figure 5 and 7, the size of the public part is 90% of the original image in LFW Dataset and only 9.8% in Holiday Dataset. The size of the secret part is 16% in LFW Dataset and 63% in Holiday Dataset. The average size of encrypted descriptor is 72KB per image for both variants, and this can be further reduced to 690B per image when using a common lossless compression, e.g., ZIP. As a summary, for the LFW Dataset, the extra communication cost brought by SouTu or SouTu_{bin} is roughly 6% of that for system without privacy consideration. But for the Holiday Dataset, our method actually save the communication cost by 27%.

The communication overhead for uploading the encoded feature descriptors (query) is approximately 36 KB in SouTu and 9 KB in SouTu_{bin}, which are reduced to 350B and 90B respectively after compression. The communication overhead for downloading the similarity result is 128B for each compared image in the database, and the one for downloading each image is similar to the uploading overhead of the image owner. Note that, to achieve k-n oblivious transfer, the querier needs to download (n - k) extra images from the search server to hide the search pattern, where n can be specified according to the trade-off between privacy and performance.

**Public & Private Bag Generation** Binary feature vector reduces the owner’s computation overhead by half to 0.51s per feature vector. In a typical scenario in LFW dataset, there are only 9 feature vectors for each face, if we use 64 dimensional SURF descriptor, it takes 9.5s on laptops and 17s on smartphones to generate the public bag, private bag and search bag. When we use binary descriptor [8], the cost is reduced to 4.9s on laptops and 5.6s on smartphones, which is a significant reduction.

**Cloud-based Photo Searching** It takes a querier roughly 1s to encode the querying descriptor in SouTu. The run time becomes negligible in SouTu_{bin}, and this is especially desirable for mobile devices. After the querier obtains the search result, it takes 0.016s on laptops and 0.024s on smartphones to decrypt each encrypted distance in both variants. In the LFW dataset, if a querier searches a photo among 1,000 photos, it takes 14s to process the search result on laptops and 22s on the smartphones on average. It is slightly beyond acceptable if owners have hundreds of photos on average. However, this non-negligible extra overhead comes from the linear search in all photos of an owner with a linear complexity, and it is promising and not trivial to reduce the complexity with existing optimized search mechanisms such as k-d tree [22].

Thus, the scalability can be achieved using those search algorithms.

2) **Communication Overhead:** The communication overhead for the image owner mainly comes from uploading public bag and private bag to the cloud. When using Blur to separate ROPs, as presented in Figure 5 and 7, the size of the public part is 90% of the original image in LFW Dataset and only 9.8% in Holiday Dataset. The size of the secret part is 16% in LFW Dataset and 63% in Holiday Dataset. The average size of encrypted descriptor is 72KB per image for both variants, and this can be further reduced to 690B per image when using a common lossless compression, e.g., ZIP. As a summary, for the LFW Dataset, the extra communication cost brought by SouTu or SouTu_{bin} is roughly 6% of that for system without privacy consideration. But for the Holiday Dataset, our method actually save the communication cost by 27%.

The communication overhead for uploading the encoded feature descriptors (query) is approximately 36 KB in SouTu and 9 KB in SouTu_{bin}, which are reduced to 350B and 90B respectively after compression. The communication overhead for downloading the similarity result is 128B for each compared image in the database, and the one for downloading each image is similar to the uploading overhead of the image owner. Note that, to achieve k-n oblivious transfer, the querier needs to download (n - k) extra images from the search server to hide the search pattern, where n can be specified according to the trade-off between privacy and performance.

**F. Cloud Side Performance**

On the clouds, similarly, the image storage and communication is 6% more for LFW Dataset and 27% less for Holiday Dataset. The main computation overhead is from the distance computation. We evaluated the search performance on laptops, so the actual performance when deployed in more powerful cloud servers will be significantly improved. Our privacy-preserving distance protocols take nearly 0.18s to calculate the distance between two real number feature vectors (SouTu) and only 0.018s for binary feature vectors (SouTu_{bin}). For well studied objects like faces (9 feature vectors in a descriptor), and for each owner, there are usually hundreds of images on

Thus, the scalability can be achieved using those search algorithms.

2) **Communication Overhead:** The communication overhead for the image owner mainly comes from uploading public bag and private bag to the cloud. When using Blur to separate ROPs, as presented in Figure 5 and 7, the size of the public part is 90% of the original image in LFW Dataset and only 9.8% in Holiday Dataset. The size of the secret part is 16% in LFW Dataset and 63% in Holiday Dataset. The average size of encrypted descriptor is 72KB per image for both variants, and this can be further reduced to 690B per image when using a common lossless compression, e.g., ZIP. As a summary, for the LFW Dataset, the extra communication cost brought by SouTu or SouTu_{bin} is roughly 6% of that for system without privacy consideration. But for the Holiday Dataset, our method actually save the communication cost by 27%.

The communication overhead for uploading the encoded feature descriptors (query) is approximately 36 KB in SouTu and 9 KB in SouTu_{bin}, which are reduced to 350B and 90B respectively after compression. The communication overhead for downloading the similarity result is 128B for each compared image in the database, and the one for downloading each image is similar to the uploading overhead of the image owner. Note that, to achieve k-n oblivious transfer, the querier needs to download (n - k) extra images from the search server to hide the search pattern, where n can be specified according to the trade-off between privacy and performance.

**F. Cloud Side Performance**

On the clouds, similarly, the image storage and communication is 6% more for LFW Dataset and 27% less for Holiday Dataset. The main computation overhead is from the distance computation. We evaluated the search performance on laptops, so the actual performance when deployed in more powerful cloud servers will be significantly improved. Our privacy-preserving distance protocols take nearly 0.18s to calculate the distance between two real number feature vectors (SouTu) and only 0.018s for binary feature vectors (SouTu_{bin}). For well studied objects like faces (9 feature vectors in a descriptor), and for each owner, there are usually hundreds of images on

Thus, the scalability can be achieved using those search algorithms.

2) **Communication Overhead:** The communication overhead for the image owner mainly comes from uploading public bag and private bag to the cloud. When using Blur to separate ROPs, as presented in Figure 5 and 7, the size of the public part is 90% of the original image in LFW Dataset and only 9.8% in Holiday Dataset. The size of the secret part is 16% in LFW Dataset and 63% in Holiday Dataset. The average size of encrypted descriptor is 72KB per image for both variants, and this can be further reduced to 690B per image when using a common lossless compression, e.g., ZIP. As a summary, for the LFW Dataset, the extra communication cost brought by SouTu or SouTu_{bin} is roughly 6% of that for system without privacy consideration. But for the Holiday Dataset, our method actually save the communication cost by 27%.

The communication overhead for uploading the encoded feature descriptors (query) is approximately 36 KB in SouTu and 9 KB in SouTu_{bin}, which are reduced to 350B and 90B respectively after compression. The communication overhead for downloading the similarity result is 128B for each compared image in the database, and the one for downloading each image is similar to the uploading overhead of the image owner. Note that, to achieve k-n oblivious transfer, the querier needs to download (n - k) extra images from the search server to hide the search pattern, where n can be specified according to the trade-off between privacy and performance.

**F. Cloud Side Performance**

On the clouds, similarly, the image storage and communication is 6% more for LFW Dataset and 27% less for Holiday Dataset. The main computation overhead is from the distance computation. We evaluated the search performance on laptops, so the actual performance when deployed in more powerful cloud servers will be significantly improved. Our privacy-preserving distance protocols take nearly 0.18s to calculate the distance between two real number feature vectors (SouTu) and only 0.018s for binary feature vectors (SouTu_{bin}). For well studied objects like faces (9 feature vectors in a descriptor), and for each owner, there are usually hundreds of images on
the cloud. The computation time for a laptop to process one request is less than one minute. When there are large-scale complicated images whose ROPs may contain random objects other than faces, the optional optimization methods introduced may be introduced to reduce the query response time.

VII. RELATED WORK

Image Privacy Protection A set of solutions are proposed to mask sensitive contents of images, e.g., human faces, to prevent any potential breach of owners’ privacy, e.g., [23] and [24]. P3 [9] proposes to separate an image into a private part and a public part and simply encrypted the private part. But the produced public parts of those works are of limited utility and disable search on them. There are some literal works providing privacy-preserving face recognition in a face photos database [25]. Those methods provide privacy protection to the requested images as well as the outcome, but the result is not secure against photo service provider and those works do not consider personal photo storage and sharing. Supporting privacy-preserving image search with untrusted server is still an open problem.

Privacy Preserving Cloud Services Many research efforts have been devoted to provide secure cloud-based storage, sharing and searching services to users. Those privacy preserving outsourced storage and sharing systems, e.g., [26] and [27], provide well access control to private data, but cannot support search on encrypted data. Searchable encryption is proposed to enable secure search over encrypted data via keywords. But the existing approaches, e.g., [28]–[31], are focus on keywords search by examining the occurrences of the searched terms (or words). They are not suitable for content-based image search since they cannot measure the distance between encrypted feature vectors.

Privacy-preserving Euclidean Distance Euclidean distance can be computed privately among parties using secure multi-party computation (SMC) methods [17]. However, it requires online interaction between the image owner and queriers, and is unsuitable for the cloud based image service, where the owners are not guaranteed to stay online. [32] proposes an approach using Fourier-related transforms to hide accurate sensitive data and to approximately preserve Euclidean distances among them. It works well for some data mining purposes on common datasets, but for feature vectors the distances still reveal information of the objects in images.

VIII. CONCLUSION

We present a framework SouTu, which enables cloud servers to provide privacy-preserving photo sharing and searching service to mobile device users who intend to outsource photo management while protecting their privacy in photos. Our framework not only protects the outsourced photos so that no unauthorized users can access them, but also enables users to encode their image search so that the search can also be outsourced to an untrusted cloud server obliviously without leakage on the query contents or results. Our analysis shows the security of the framework, and the implementation shows a small storage overhead and communication overhead for both mobile clients and cloud servers.

REFERENCES

[1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” IJCV, vol. 60, no. 2, pp. 91–110, 2004.
[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (surf),” CVIU, vol. 110, no. 3, pp. 346–359, 2008.
[3] J. Leibe, E. Seemann, and B. Schiele, “Pedestrian detection in crowded scenes,” in CVPR. IEEE, 2005.
[4] P. Viola and M. J. Jones, “Robust real-time face detection,” International journal of computer vision, vol. 57, no. 2, pp. 137–154, 2004.
[5] M. Turk and A. Pentland, “Eigenfaces for recognition,” Journal of cognitive neuroscience, vol. 3, no. 1, pp. 71–86, 1991.
[6] K. Mikolajczyk and C. Schmid, “Scale & affine invariant interest point detectors,” IJCV, vol. 60, no. 1, pp. 63–86, 2004.
[7] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” in ECCV, 2010.
[8] K. Peker, “Binary sift: Fast image retrieval using binary quantized sift features,” in CBMI, 2011.
[9] M.-R. Ra, R. Govindan, and A. Ortega, “P3: Toward privacy-preserving photo sharing,” in NSDI. USENIX, 2013.
[10] M. McDonnell, “Box-filtering techniques,” Computer Graphics and Image Processing, vol. 17, no. 1, pp. 65–70, 1981.
[11] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in S&P. IEEE, 2007, pp. 321–334.
[12] M. Chase and S. S. Chow, “Improving privacy and security in multi-authority attribute-based encryption,” in CCS. ACM, 2009.
[13] M. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in EUROCRYPT. Springer, 1999, pp. 223–238.
[14] J. Camenisch, G. Neven et al., “Simulatable adaptive oblivious transfer,” in EUROCRYPT. Springer, 2007, pp. 573–590.
[15] “Extended version with security proof.” https://www.dropbox.com/s/c721a3gov75yv/SouTu.pdf.
[16] A. Alahi, R. Ortiz, and P. Vandergheynst, “Freak: Fast retina keypoint,” in CVPR. IEEE, 2012.
[17] Y. Lindell and B. Pinkas, “A proof of yao’s protocol for secure two-party computation.” IACR Cryptology ePrint Archive, p. 175, 2004.
[18] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” Tech. Rep.
[19] J. Luo, Y. Ma, E. Takikawa, S. Lao, M. Kawade, and B.-L. Lu, “Person-specific sift features for face recognition,” in IJCAI. IEEE, 2007.
[20] H. Jegou, M. Douze, and C. Schmid, “Hamming embedding and weak geometric consistency for large scale image search,” in ECCV, 2008.
[21] T. Lindeberg, “Feature detection with automatic scale selection,” IJCV, vol. 30, no. 2, pp. 79–116, 1998.
[22] D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration,” in VISAPP (1), 2009, pp. 331–340.
[23] E. M. Newton, L. Sweeney, and B. Malin, “Preserving privacy by de-identifying face images,” TKDE, vol. 17, no. 2, pp. 232–243, 2005.
[24] W. Zhang, S.-C. S. Cheung, and M. Chen, “Hiding privacy information in video surveillance system.” in ICIP, 2005.
[25] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, “Efficient privacy-preserving face recognition,” in Information, Security and Cryptology. Springer Berlin Heidelberg, 2010, pp. 229–244.
[26] Y. Tang, P. P. Lee, J. C. Lui, and R. Perlman, “Secure overlay cloud storage with access control and assured deletion,” TDSC, vol. 9, no. 6, pp. 903–916, 2012.
[27] B. Wang, B. Li, and H. Li, “Oruta: Privacy-preserving public auditing for shared data in the cloud,” in CLOUD. IEEE, 2012.
[28] C. Wang, N. Cao, K. Ren, and W. Lou, “Enabling secure and efficient ranked keyword search over outsourced cloud data,” TPDS, vol. 23, no. 8, pp. 1467–1479, 2012.
[29] M. Li, S. Yu, W. Lou, and Y. T. Hou, “Toward privacy-protected cloud data services with flexible search functionalities,” in ICDCSW. IEEE, 2012.
[30] G. Wang, K. Ren, S. Yu, and K. M. R. Urs, “Achieving usable and privacy-assured similarity search over outsourced cloud data,” in INFOCOM. IEEE, 2012, pp. 451–459.
[31] Y. Ren, Y. Chen, J. Yang, and B. Xie, “Privacy-preserving ranked multi-keyword search leveraging polynomial function in cloud computing,” in Globecom. IEEE, 2014.
[32] S. Mukherjee, Z. Chen, and A. Gangopadhyay, “A privacy-preserving technique for euclidean distance-based mining algorithms using fourier-related transforms,” VLDB Journal, vol. 15, no. 4, pp. 293–315, 2006.
APPENDIX

Firstly, we prove by the following game that SouTu is secure against adversarial cloud servers who are not authorized for image search.

Initialize: System is initialized, and relevant cryptosystems (Paillier’s cryptosystem, CP-ABE, OT etc.) are initialized by the challenger $C$. $C$ publishes relevant public keys to the adversary $A$.

Setup: $C$ generates/encrypts the search keys, and preprocesses/encrypts a set of images $I$ by the specification of SouTu such that $A$ cannot search on him. Then, he publishes the encrypted search keys and public/private/search bags to $A$.

Phase 1: $A$ achieves polynomial number of encoded descriptors (encoded with $C$’s search keys) without knowing corresponding original descriptors.

Challenge: $A$ submits two images $I_0, I_1$ to $C$. $C$ selects a bit $y \in \{0, 1\}$ uniformly at random, and generates two sets of encoded feature descriptor $C_1(\mathbf{f}(Y))$, $C_2(\mathbf{f}(Y))$ corresponding to $I_y$ (Section [IV-D]), which are given to $A$.

Guess: $A$ gives a guess $y'$ on $y$.

The advantage of $A$ in this game is defined as

$$\text{adv} = \Pr \left[ A \leftarrow C_1(\mathbf{f}(Y)), C_2(\mathbf{f}(Y)) \mid y' = y \right] - \frac{1}{2}$$

It is not hard to see this is an adversarial cloud server’s advantage, since the game is designed to ‘mimic’ a cloud server’s transaction.

Theorem 1: Any probabilistic polynomial time adversary (PPTA) has at most negligible advantage in above game.

Proof: We define two PPTAs $A_1, A_2$, and define their advantages $\text{adv}_i$ as:

$$\text{adv}_i = \Pr \left[ A_i \leftarrow C_i(\mathbf{f}(Y)) \mid y'_i = y \right] - \frac{1}{2}$$

That is, $\text{adv}_i$ is the advantage of $A_i$ when he is only given $C_i(\mathbf{f}(Y))$ and gives a guess $y'_i$ on $y$. Since $A$ is given both adversaries’ views, if $A_1, A_2$ agree on the same guess, he will also give the same guess, otherwise his advantage does not change. Then, we have the following probabilities for four cases:

$$\Pr [y' = y | y'_1 = y \land y'_2 = y] = 1$$
$$\Pr [y' = y | y'_1 = y \land y'_2 \neq y] = \Pr [y' = y]$$
$$\Pr [y' = y | y'_1 \neq y \land y'_2 = y] = \Pr [y' = y]$$
$$\Pr [y' = y | y'_1 \neq y \land y'_2 \neq y] = 0$$

Since $A_1$ and $A_2$ gives their guesses based on independent views, we have

$$\Pr [y'_1 = y \land y'_2 = y] = \Pr [y'_1 = y] \cdot \Pr [y'_2 = y]$$

which also applies to the other three cases. Given those conditional probabilities, the total probability is ($\text{adv} = \Pr [y' = y] - \frac{1}{2} = 1 - \Pr [y' \neq y] - \frac{1}{2}$):

$$\Pr [y' = y] = \frac{1}{2} + \text{adv}$$
$$= 1 \cdot \left( \frac{1}{2} + \text{adv}_1 \right) \left( \frac{1}{2} + \text{adv}_2 \right)$$
$$+ \left( \frac{1}{2} + \text{adv} \right) \left( \frac{1}{2} + \text{adv}_1 \right) \left( \frac{1}{2} - \text{adv}_2 \right)$$
$$+ \left( \frac{1}{2} + \text{adv} \right) \left( \frac{1}{2} - \text{adv}_1 \right) \left( \frac{1}{2} + \text{adv}_2 \right)$$
$$+ 0$$

which leads to

$$\text{adv} = \frac{(\frac{1}{2} + \text{adv}_1)(\frac{1}{2} + \text{adv}_2) - \frac{1}{2}}{1 - 2\text{adv}_1\text{adv}_2}$$

Both Paillier’s cryptosystem and CP-ABE are proved to be semantically secure against chosen plaintext attack (SS-CPA)\(^3\). Therefore, $A$ does not have a significant chance to get $SK, r$ in $\text{ABE. E(PK, SK, r)}$ or $\mathbf{f}(Y)$ in $C_2(\mathbf{f}(Y))$, which means $\text{adv}_2$ is negligible. Recall that the function family $x \rightarrow \mu x \mod n$ is $\epsilon$-pairwise independent for negligible $\epsilon$, and $\mu^{-1}x \mod n$ is close to uniform in $\mathbb{Z}_n$. Therefore, he does not have a significant chance to get $\mathbf{f}(y_j(k))$ in $C_1(\mathbf{f}(y_j(k)))$ either, which implies a negligible $\text{adv}_1$. Since both $\text{adv}_1, \text{adv}_2$ are negligible, $\text{adv}$ is negligible too.

Besides adversarial cloud servers, we assumed malicious users in our adversarial model. However, unauthorized malicious users are not as threatening as cloud servers since they never get involved in any transaction with valid users. All they can do is to try man-in-the-middle attacks sniff the search results, which can be trivially prevented with secure communication channel.

\(^3\)CP-ABE is proved to achieve IND-CPA, which implies SS-CPA