ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

IJARMATE
Your uh-MATE Research Paper

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. IV, Issue IV, April 2018

CONTENT-BASED RETRIEVAL IN CLOUD IMAGE MINES

[1], Dhivya.M, [2] Piramu.M

[1], M.E. Student, Department of Computer Science and Engineering ,P.S.R Engineering college, Anna University, Chennai.

[2]Associate Professor, Department of Information Technology, Anna University, Chennai.

Abstract—Storage requirements for visual data has been increasing in recent years, following the emergence of many new services and applications for both personal and corporate use. This has been a key driving factor for the adoption of cloud-based data outsourcing solutions. However, outsourcing data storage to the Cloud also leads to new challenges that must be carefully addressed, specially regarding privacy. In this paper we propose a novel secure framework for outsourced and distributed privacy-preserving storage and retrieval in large image repositories. Our proposal is based on a novel cryptographic scheme, named IES-CBIR, specifically designed for media image data. Our solution enables both encrypted storage and querying using Content Based Image Retrieval (CBIR) while preserving privacy. We have built a prototype of the proposed framework, analyzed its security properties, and experimentally evaluated its performance and precision. Our results show that IES-CBIR allows more efficient operations than existing proposals, both in terms of time and space complexity, while enabling less restrictive use cases and application scenarios.

I. INTRODUCTION

Nowadays visual data is responsible for one of the largest shares of global Internet traffic in both corporate and personal use scenarios [1]. The amount of images, graphics, and photos being generated and shared everyday is growing at an ever increasing rate. The storage needs for such large amounts of data has been a driving factor for data outsourcing solutions such as the ones leveraging Cloud Storage and Computing services. Reports show that Cloud services specifically de-signed for image storage and sharing, such as Instagram and Flickr, are among the largest growing internet services today

[2]. Additionally, the availability of large amounts of images in public and private repositories opens the way for other interesting services, such as the retrieval of images based on their contents (CBIR).

Despite the fact that data outsourcing seems a natural solution to support large scale image storage and retrieval systems, it also raises new challenges in terms of data control and privacy. This is a natural consequence of outsourcing data, which usually implies releasing control (and some times even full ownership) over it [3]. Recent news have provided clear proofs that privacy should not be expected to be preserved from outsourced storage providers [4], [5]. Furthermore, malicious system administrators working for the providers have full access to data on the hosting cloud machines [6], [7]. Finally, external hackers can exploit software vulnerabilities to gain unauthorized access to servers.

The conventional approach to address privacy in this con-text has been to encrypt sensitive data before outsourcing it and run all computations on the client side [8]. However, this imposes too much client-overhead, as data must continuously be downloaded, decrypted, processed and then re-upload after encryption. Many applications cannot cope with this overhead, particularly online and mobile applications operating over very large datasets (e.g, searching), such as image repositories with CBIR services. A more viable approach would be to out-source computations, performing operations over the encrypted data on the server side. Existing proposals in this domain remain on the theoretical realm, namely those requiring fully homomorphic encryption, which is still computationally too expensive [9]. Nonetheless, partially homomorphic encryption schemes are an interesting alternative, with recent proposals pursued by the research community, yielding more practical results while providing a good tradeoff between privacy and usability [10]–[13]. Unfortunately, existing solutions are still computationally too complex for wide adoption, particularly regarding the support of private CBIR over large-scale image repositories.

To address these challenges we propose a novel Image Encryption Scheme with CBIR capabilities (IES-CBIR), which supports outsourcing of private storage and search/retrieval of images in the encrypted domain. Key to the design of our scheme is the observation that in images, color information can be separated from texture information, enabling the use of different encryption techniques with different properties for protecting each of these features. Following this observation, and considering that texture is usually more relevant than color in object recognition [14], in IES-CBIR we make the following tradeoff: we choose to prioritize the protection of image contents, by encrypting texture information with probabilistic encryption; then we somewhat relax the security on color features, by using deterministic encryption on image color information[1]. This combination provides support for privacy-preserving CBIR based on color information to be performed directly on outsourced servers, while still protecting the contents of images from the operators of these servers and users issuing queries.

In addition to IES-CBIR proposal, in this paper we also present the following contributions: (i) we show how IES-CBIR can be used in a flexible way to design outsourced image storage systems with CBIR support for the color domain, while avoiding complex computations to be performed by the client and therefore avoiding performance pitfalls that exist in other works [12], [13]; (ii) we present a security analysis focused on the privacy provided by our scheme for the considered adversary model; (iii) we experimentally show that when compared with competing alternatives [10], [11], IES-CBIR provides increased cryptographic throughput, lower ciphertext expansion, and lower computational overhead; and finally (iv) we show that the precision and recall [16] of a CBIR based on color features using IES-CBIR is comparable with current state of the art. The remainder of this paper is organized as follows: Section II overviews the relevant related work, comparing it to our solution; Section III presents the system model, use cases, and adversary model definitions subjacent to our proposal; Section IV describes the main building blocks that compose our solution; Section V evaluates our work in terms of privacy, retrieval precision, and performance; and Section VI concludes the paper.

II. Related Work

Previous proposals for supporting outsourced storage and search/retrieval of images in the encrypted domain can be di-vided in two classes: Searchable Symmetric Encryption (SSE) based approaches and Public-Key partially-Homomorphic ap-proaches (PKHE). SSE has been widely used in the past by the research community, both for image [12], [13], [17] and text [18]–[21] retrieval. These approaches force clients to generate indexing structures describing the data, and then separately encrypting both data and indexes and uploading them to the cloud. The data encryption is typically performed using con-ventional symmetric cryptography. In order to allow privacy-preserving search over them, indexing structures are encrypted with a symmetric-key cryptographic scheme displaying some form of homomorphic properties, such as determinism [22] or order-preservation [15].

SSE approaches in the image domains can be found in proposals by Lu et al. [12], [13], where different cryptographic algorithms are proposed to encrypt index and feature vectors extracted from image repositories while preserving the ability to perform CBIR based on color features. Yuan et al. [17] also present a similar approach, although more focused on recommendation and social discovery. Unfortunately, SSE-based approaches present significative limitations: (i) clients are required to compute and encrypt indexes locally, which entails the use of additional computational power and limits the scalability of the solution, specially for mobile clients; (ii) clients have to transfer additional data to the cloud, leading to additional bandwidth and storage space consumption in these servers, negatively impacting the storage operations latency perceived by users; (iii) finally, managing dynamic updates becomes unpractical, i.e. when a user edits his outsourced repository he may be required to download the indexing structures (and/or image feature vectors), decrypt them, up-date/recompute their scores, encrypt them again and re-upload them to the cloud, exacerbating the above drawbacks.

The alternatives to SSE that can be found in the literature [10], [11] are based on public-key partially-homomorphic encryption (PKHE) schemes such as Paillier [23] or ElGammal [24] (which allow only additions and multiplications on the encrypted domain, respectively). In these approaches clients encrypt images in a way that outsourcing servers can perform all image indexing and querying operations directly over the encrypted data, avoiding many of the practical issues of SSE-based solutions. Unfortunately, PKHE presents much higher time and space complexities when compared with SSE schemes. For instance, Hsu et al. [10] have designed a powerful CBIR algorithm for the encrypted domain (based on SIFT [10]) by resorting to the Paillier cryptosystem [23]. However, their approach results in significative ciphertext expansion (for a secure key size of at least 1024 bits, each pixel is transformed from its traditional 24 bits representation into 2048 ciphertext bits), slow encryption and decryption times (as we will experimentally show in our evaluation section V-B2), and in scalability issues (the “ciphertext blowup” problem [25]).

To overcome some PKHE limitations in image-processing, another approach was proposed by Zheng et al. [11]. In this work the authors rely in an additively homomorphic scheme for images also based on Paillier, where ciphertexts are replaced by pointers to a ciphertext table (built by mapping those pointers to all possible ciphertext pixel values). While this approach reduces the number of encryption operations required while also mitigating the ciphertext expansion problem, it still presents a significative computational overhead which limits its practicality.

In this work we propose IES-CBIR, a novel cryptographic scheme that allows us to design outsourced image repos-itory systems that support CBIR based on color features, while protecting the privacy of both image owners and users issuing queries. Comparing with the state-of-art, IES-CBIR has a computational complexity similar to that of SSE-based systems, while avoiding clients to perform computations to generate and update image indexing structures. IES-CBIR also minimizes ciphertext expansion and consequently bandwidth and outsourced space requirements, with positive impact on user-perceived latency. These benefits are illustrated in our experimental analysis in Sec. V, where the performance of a IES-CBIR system is compared against the state of the art SSE [13] and PKHE [10] based approaches.

III. System and Adversary Models

1) System Model: We start by describing a generic system model that we envision for the application of IES-CBIR to design outsourced image repositories with CBIR support and high privacy. In this model, we consider two main entities: the Cloud and (multiple) Users (Fig. 1). Images are outsourced to repositories which are managed by a cloud infrastructure operator (i.e., the cloud). Each repository is used by multiples Users, which can both add their own images to the repository and/or issue queries over the data stored in the cloud using an image (image query) or the unique identifier of an image (id query). Users can also request access to stored images. To ensure the privacy of users, all data sent to repositories (including new images being stored and query images) are encrypted using our image encryption scheme, IES-CBIR.

Repositories are created by a single user. Upon their cre-ation, a new trapdoor (or search) key is generated by the user.

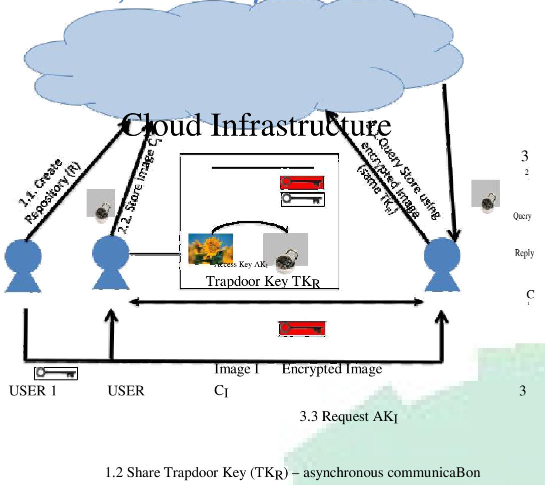

Fig. 1: System Model

This key allows users to issue search and retrieval operations over it's repository. The trapdoor key is then shared with other trusted users, allowing them to also store new images on the repository and search over it. Key sharing can be done asynchronously and out-of-band, for instance, by resorting to a key-sharing protocol with public-key authentication such as Needham-Schroeder-Lowe [26]. However we find that the discussion on how to securely share keys between Users falls outside the scope of this paper.

When adding images to a repository, a user will first encrypt them using the IES-CBIR encryption algorithm. To this end the user relies on the trapdoor key of the target repository, as well as an additional access key per image. An access key is generated by a user when storing a new image, and kept secret by him. The encrypted image is then transmitted to the cloud platform, which is responsible for extracting and indexing its relevant features [27].

Users that have access to a repository can also issue queries over it, using one of their already stored images or a new a query image. In the later case, the new query image is encrypted with IES-CBIR through the use of the repository's trapdoor key and a randomly generated access key. The cloud infrastructure is responsible for processing the query and sending the top k matches to the issuer of the query, where k is a configurable parameter per query.

The reply to a query will contain k pairs of ciphertext– metadata, which include the id of the user that stored the image and the image's id. In order to fully decrypt and access the contents of an image, the querying user will require not only the trapdoor key of the repository, but also the access key for that image. To that end, the user must contact the owner of that image, providing its identifier and requesting its corresponding access key. User interactions can be done asynchronously (and out-of-band) by resorting to a public-key authenticated protocol as discussed previously.

2) Relevant Use Cases: As a way to illustrate the broad applicability of our system model, we now briefly discuss two relevant use cases and explain the mapping of the concrete entities in these use cases to our model.

a) Personal Health Records: Personal health records (PHR) storage is being offered today as an outsourced service by major cloud operators [2]. PHR may contain both textual and/or image information (e.g. colored MRAs, skin cancer photos, among others) from previous medical consults or exams of several patients followed by different medical doctors at different healthcare centers. The availability of this information, not only ensures a better service towards patients, but also offers a high potential for the exchange of healthcare in-formation among different medical professionals and institutes to assist them in treating patients with similar conditions, as well as for research proposes. In this scenario, medical doctors are Users of the system, and outsource PHR of their patients to a cloud-based backend (the Cloud). In the cloud, PHR are organized in alliance-based repositories between cooperating professionals and/or medical specialty-based repositories. Be-cause PHRs contain sensitive information and belong to the patients, these records can be protected by an access key only known to the patient. The trapdoor key of a repository is shared among all cooperating medical doctors of all medical centers involved in this effort. Doctors can then perform search operations on these repositories, and indirectly request the keys to PHRs that might be of their interest, through the physician following the patient to whom those records belong to.

b) Storage for Mobile Users: Existing studies have shown that Internet users are increasingly mobile [1]. Since mobile clients usually have limited computational and storage resources, they tend to rely on cloud services for storing and processing bulky data such as images. In this scenario, mobile clients (Users) want to delegate their private image repositories storage to a Cloud provider, in order to cope with the limitations of their device's storage capability, computational power, and battery life. Additionally, clients might be interested in allowing their images to be searched (and eventually accessed) by other users (either friends, family, or co-workers). Privacy can be relevant for instance when a user is a public figure or has access to sensitive material. Additionally, one might imagine that some companies could have interest in accessing the images owned by a given user, for instance when performing background checks on prospective new employees, among other scenarios.

3) Adversary Model: We aim at protecting the privacy of users' images and queries. In this work, we don't consider integrity or availability threats, as they can be handled by different mechanisms that are orthogonal to the contributions of this paper. Attacks on privacy may come from honest but cu-rious cloud administrators managing the cloud's infrastructure, malicious users that deviate from their expected behavior, or external hackers. Concerning a malicious cloud administrator, we assume that he may have access to all data stored on disk or in RAM on any device physically located at the cloud, and passing through the network from or to the cloud. We follow the Honest but Curious Cloud Model introduced by [3], meaning that it will perform operations as intended but may try to access encrypted image contents and related information. On the other hand, external malicious entities are assumed to only be able to access data passing through the network, or in the particular case of malicious users, being able to issue queries over a repository (with or without access to the correct

trapdoor key). Leveraging their capacities towards the system, the described adversaries attempt to break any secure protocols and/or cryptographic schemes to gain unauthorized access to data owned by (other) users and stored in the repositories at the cloud. We further assume that distinct adversaries can collude with each other (e.g. malicious users and cloud operators), as they won't gain any further advantage as a whole against the system. Although we do not explicitly assume a malicious cloud operator, we note that such an adversary could indeed compromise the availability and integrity of the data stored in the cloud, being however unable to compromise the privacy of stored content.

To protect against privacy attacks, we rely on a novel cryptographic scheme specifically designed for images (we cover the design of this algorithm in Sec. IV-A and its security discussion in Sec. V-A). This scheme reduces the Trusted Computing-Base to the users' devices and defines the cloud's infrastructure and communication channels as not trusted.

IV. IES-CBIR Design and Implementation

Our solution is composed of two main components: an image encryption component, that is executed on the users devices; and a storage, indexing, and searching component (in the encrypted domain), executed in the cloud. Next we present the design of our proposed solution, but first we discuss a brief definition of what should be privacy for images.

Definition 1 (Image Privacy). We define privacy for an image as the inability of an (unauthorized) entity to detect or identify objects (i.e, item, persons, etc) on the pixels that compose the image, given their color values in some color space and their positions within the image.

We remark that pixel color values define the color intensity of images as a whole, and that pixel positions, i.e. pixels and their adjacent pixels, allied with strong color changes in those adjacent pixels, define texture information (the presence of objects within the image). We also remark that usually, texture information is more relevant in image retrieval and object recognition [14]. Finally, we conclude that no sub-component alone (i.e. color or texture data) can be used to infer the precise contents of an image, as color data on itself is usually ambiguous (e.g. strong blue can mean sky, ocean, etc.) and texture data depends not only on pixel positions but also on their color values (we assume no previous background information is available on the image or its repository). In the light of this definition, we now present our solution in detail.

A. Image Encryption

The main component on the users side is based on a novel cryptographic scheme specifically designed for images and privacy preserving CBIR, which we dubbed IES-CBIR. Our scheme leverages the above definition on image privacy, by separating color from texture information and applying different levels of security when protecting each. Following the definition, more specifically the remark that texture is usually more relevant than color for object recognition, in IES-CBIR we protect image texture with probabilistic encryption and color information with deterministic encryption. Hence, privacy-preserving CBIR based on color can be performed on the cloud servers, without intervention of users, while fully protecting image texture (a detailed security evaluation can be found in sec. V-A). We define IES-CBIR as:

Definition 2 (IES-CBIR). An Image Encryption Scheme with CBIR capabilities (IES-CBIR) is a tuple (GenTK, GenAK, Enc, Dec, Trp) of five polynomial-time algorithms, where:

GenTK(sp_{tk}): is a probabilistic algorithm that takes as input the security parameter sp_{tk} 2 N and generates a trapdoor key tk with length polynomially bounded by it;

GenAK(sp_{ak}): is a probabilistic algorithm that takes as input the security parameter sp_{ak} 2 N and generates an access key ak with length polynomially bounded by it;

Enc(I; tk; ak): is an algorithm that takes as input an image I and the cryptographic keys ftk; akg and returns an encrypted image C_I ;

Dec(C_I ; tk; ak): is an algorithm that takes as input an encrypted image C_I and keys ftk; akg and returns the decrypted image I;

Trp(Q; tk): is an algorithm that takes as input a query image Q and a trapdoor key tk and returns a trapdoor T_Q;

Algorithm 1 Create New Repository

```
1: procedure User(U ).createRepository(R, $sp_{tk}$)   . User operation
2:     $tk_R$  GenTK($sp_{tk}$)
3:     cloud.CreateRepository(R)                  . Remote call to cloud
4:     return $tk_R$
5: end procedure
```

request full access to an image by asking its owners for the corresponding access key.

C. System Protocols

In this section we provide a system-centric description on the use of IES-CBIR to design a cloud-based image storage system with CBIR support over color features. To this end, we briefly describe the main operations available to users, accompanied by the algorithms used to implement them. The description provided here is consistent with the prototype used for our experimental work, whose results are reported in sec. V. We omit operations related with the request and sharing of keys, as these are orthogonal to the use of IES-CBIR.

1) Instantiate a new Repository: We start by describing the operation used by a user to create a new repository (Alg. 1). The main operation associated with this task is the generation, by the user, of the trapdoor key tk_R for repository R, through the GenTK algorithm (line 2), given the security parameter sp_{tk} provided by the user. sp_{tk} will usually have the value of 101, in order to cover the range [0::100]. On the cloud's side, the operation CreateRepository simply creates an empty index, and pre-allocates some storage space for a repository R.

2) Upload New Image: Alg. 2 illustrates the procedure followed by a user U to store a new image on repository R in the cloud. U provides as input: repository's id R, an image I, R^{0}s trapdoor key tk_R, and the security parameter for generating a access key (sp_{ak}). The algorithm is straight-forward: first an access key for image I is generated (ak_I) through the algorithm GenAK (line 2). This key, together with the tk_R, is used to encrypt I to its ciphertext C_I using the Enc algorithm (line 3). C_I is then uploaded to the cloud for

storage (line 4) using the cloud's storeImage remote procedure. Upon receiving ciphertext C_I , the cloud generates an unique identifier ID_I , extracts from C_I the color feature vectors (three color histograms) and indexes them (lines 8-10). C_I is then persistently stored and ID_I is returned to the user (lines 11-12). Finally, the Upload procedure returns a pair containing the image identifier ID_I , and the access key ak_I , which are required for future accesses to the image.

3) Issue a Query (with an image): Alg. 3 sketches the procedure used by users (and the respective algorithm executed by the cloud) to perform searches over a repository R using image Q. As input for this operation the user has to provide the trapdoor key for repository R (tk_R) and also a parameter k, which is the number of results to be returned. User U starts by generating Q's trapdoor, T_Q, using IES-CBIR GENTRP algorithm with inputs Q and tk_R (line 2). T_Q is then sent to the cloud (line 3) as a parameter for the Search remote invocation. The cloud computes the feature vectors of T_Q, find the image's representation in the BOVW model and then accesses the index, finding the top k most similar images and returning their unique identifiers and ciphertexts to the query issuer U (lines 7-10).

Algorithm 2 Store New Image

```
1:  procedure USER(U ).UPLOAD(R; I; $tk_R$; $sp_{ak}$) . Performed by a user
2:      $ak_I$  GENAK($sp_{ak}$)
3:      $C_I$      ENC(I; $tk_R$; $ak_I$ )
4:      $ID_I$      cloud.storeImage(R; $C_I$ )
5:      return f$ID_I$ ; $ak_I$ g
6:  end procedure
7:  procedure CLOUD.STOREIMAGE(R; $C_I$ )          . Performed by the cloud
8:      $ID_I$  GenerateID($C_I$ )
9:      $hists_I$ = f$hist_H$ ; $hist_S$; $hist_V$ g          ExtractFeatureVectors($C_I$ )
10:     IndexFeatures(R; $ID_I$ ; $hists_I$ )
11:     Put(R; $ID_I$ ; f$C_I$ ; $hists_I$ g)
12:     return $ID_I$
13: end procedure
```

Algorithm 3 Issue Query

```
1:  procedure USER(U ).SEARCH(R; Q; $tk_R$; k)   . Performed by a user
2:      $T_Q$      GenTrp(Q; $tk_R$)
3:      rankedImgDistances  cloud.Search(R; $T_Q$; k)
4:      return rankedImgDistances
5:  end procedure
6:  procedure CLOUD.SEARCH(R; $T_Q$; k)          . Performed by the cloud
7:      $hists_{T_Q}$ = f$hist_H$ ; $hist_S$; $hist_V$ g          ExtractFeatureVectors($T_Q$)
8:      $BOVW_{T_Q}$        R.BOVWRepresentation($hists_{T_Q}$ )
9:      KMostSimilar  Sort(R.KMostSimilarInIndex(BOV $W_{TQ}$ ; k))
10:     return KMostSimilarImages
11: end procedure
```

4) Access an Image: Alg. 4 illustrates the mechanism to access an image I. This algorithm can be executed by any user that has gained access to it's encrypted counterpart C_I , including it's owner or a user that has searched the repository R (using a trapdoor key tk_R) and later obtained from I's owner the corresponding access key ak_I . The algorithm is a straightforward application of the IES-CBIR DEC algorithm introduced earlier in the paper. Note however, that in order to access the plaintext of image I, both the trapdoor key of the repository where the image was stored, and the access key (unique) for the image are required.

V. EVALUATION

In this section we evaluate our proposal. We start by analyzing the security properties of IES-CBIR.

A. Security Analysis

We start this security analysis by clearly defining what we allow to be leaked to a Probabilistic Polynomial-Time bounded Adversary (PPTA, representing our adversary model from sec. III-3: malicious cloud administrator, malicious user or internet hacker). Then, we discuss IES-CBIR security properties, by analyzing its robustness against key and ciphertext cryptanal-ysis in the adversary and leakage models defined.

Definition 3 (Allowed Information Leakage). In IES-CBIR, we define that we allow the leakage of search patterns and image similarity based on color features. Search pattern leakage is common to all efficiently searchable encryption schemes [10]– [13], [17]–[22], and is only addressed by Oblivious RAM works [30] at much higher costs in terms of computation and bandwidth. Image similarity based on color features is the desired property we are outsourcing to the cloud, allowing us to minimize users computational overhead and to address the use cases described in sec. III-2. Furthermore, this similarity leakage is only statistical (true color values of images and their pixels are always encrypted), and, following our image privacy definition (def. 1) and our security evaluation (bellow), won't suffice for a PPTA to access encrypted image contents.

Algorithm 4 Access an Image

```
1: procedure ACCESS(I; $tk_R$; $ak_I$ )               . Performed by a User
2:      I  DEC($C_I$ ; $tk_R$; $ak_I$ )
3:      return I
4:  end procedure
```

1) Key Security: IES-CBIR depends on two different keys: trapdoor and access keys. Trapdoor keys are composed of three sub-keys (tk = ftk_H ; tk_S; tk_V g), where each sub-key is a random permutation of all values between [0::100]. An attack on each sub-key requires trying 101! possible values (each sub-key is a random permutation of 101 values). As the trapdoor key is composed of 3 equally complex sub-keys, where all 3 are used in image encryption, the complexity to search the full space of the trapdoor key is $(101!)^3$ which is computationally unfeasible for any PPTA. Access keys (ak) have a size of 128 bits, and are used as a cryptographic seeds for a PRNG function. These keys are only used as entropy generators for the PRNG, and their size can be parametrized in function of the PRNG algorithm used. Nonetheless, we consider a size of 128 bits (which is also the smaller key size allowed for the AES algorithm) to be secure for a PRNG algorithm implementation, such as RC4-based PRNG, for any PPTA.

2) Encrypted Images Security: In our adversary model, PP-TAs will (cloud operators) or may (malicious users and internet hackers) have access to all encrypted images in the different repositories. As such, we must analyze what can be observed by these adversaries. Since IES-CBIR separates image color from texture information, using different

encryption algorithms to protect each, we separately analyze their security properties.

In terms of texture information, reconstructing an original image is almost as hard as solving a random jigsaw puzzle. The random values used to shift row and column positions range between 1 and the image width w and height h, respectively. To find the lower security-bound values w and h for this algorithm, consider the particular example of a small image of 16 16 pixels. In this case, IES-CBIR requests from the PRNG function 16 + 16 = 32 random values in the range [1::16]. Since 4 bits are enough to represent all values up to 16, a PPTA would need to try $2^{(32\ 4)} = 2^{128}$ combinations in order to generate all possible permutations of pixels in the image. More generally, one can show that for an image of size w l, the difficulty to reconstruct it from its ciphertext can be deduced to be $w^{w}\ 1^{l}$. Assuming that a space of $16^{16}\ 16^{16} = 2^{128}$ is secure against PPTAs (as this would require an effort of around 10^{17} years for an adversary with 10^{6} cpus, each performing 10^{6} IES-CBIR encryption operations per second[4]), we can conclude that our approach will be secure for images as small as 16 16 pixels[5]. Interestingly we note that for images beyond this threshold size, attacking the access key is always easier (albeit also unfeasible as seen above).

In terms of color information, as we defined in def. 3, some statistical properties will be leaked due to the deterministic algorithm used in IES-CBIR encryption. In fact, a PPTA with access to ciphertexts will be able to extract color histograms from the encrypted images and identify similarity among images relatively to their color. This however, is a desirable feature to support privacy-preserving CBIR in the cloud, as discussed previously. Nonetheless, even if a PPTA has access to a large number of ciphertexts, it will still be impossible to disclose the real color values of pixels and decrypt any image.

Furthermore, we remark that even if a PPTA has managed to gain access to a repository's trapdoor key, some access keys and/or pairs of plaintext-ciphertext images (for instance, a malicious user), he won't gain any advantage over other encrypted images in that repository or over other repositories, as long as the PRNG and the Symmetric-Key Generator functions used are secure. These functions are used as is, and so IES-CBIR inherits their security properties directly.

B. Experimental Evaluation

To access the benefits of leveraging IES-CBIR to design privacy-enhanced image outsourced storage we have imple-mented a prototype system. The prototype, developed in the Java language, implements IES-CBIR and the protocols de-scribed in sec. IV-C. Through this prototype, we conducted an experimental evaluation of the security, performance and pre-cision of the proposed solution. All experimental assessments were carried out through Amazon EC2 instances, both for user simulation and for the cloud storage. To simulate geographic distance, user computations were done in Oregon's data-center instances, and the cloud's computations were done in a North-Virginia's data-center instance. Furthermore, all instances were of the general-purpose m3.large type[6]. For testing purposes, we used two image datasets: the Wang dataset [14], containing 1000 low-resolution images with JPEG compressed size of 29.8 MB; and the Inria Holidays Dataset [31], containing 1491 high-resolution images with total JPEG compressed size of 2.8 GB. We present our results in the following order: first we start by discussing the experimental security evaluation from the statistical analysis of entropy provided by IES-CBIR. Then we discuss the performance evaluation, comparing our solution with the relevant related works. Finally we show the retrieval precision achieved, also comparing with the related approaches.

1) Experimental Security Evaluation: For this evaluation we performed a statistical analysis to assess the level of entropy in encrypted images. The assessment consisted in analyzing the level of correlation between all horizontally, vertically and diagonally adjacent pixels, for original plaintext images, at different steps of IES-CBIR encryption process, and for a complete random permutation of all pixel positions. We used the correlation function of [32], where the obtained values range between [1; :::; 1], and the images with higher entropy get closer to 0. For this test we used the low-resolution Wang Dataset, proving that IES-CBIR can achieve high levels of entropy even for smaller images. All pixels of all images in the dataset were considered, being the average results presented in Fig. 2. The first point in the figure represents the plaintext images; the second represents IES-CBIR color encryption only; the third is color encryption plus columns shifting; the fourth is color encryption plus columns and rows shifting (i.e., full IES-CBIR encryption); and the last point is random permutation of all pixel positions between each others. The results show that color encryption alone reduces much of pixel correlation levels but its not enough (avg. 0; 25 correlation). By adding columns and rows random shifting (texture encryption), correlation level is brought to close to 0 values (0; 0006 for vertical and diagonal correlation and 0; 02 for horizontal, as most images in the dataset have less height than width). Furthermore, with random permutation of all pixels we can further decrease correlation by one order of magnitude (0; 0001 and 0; 00003), but at a much higher performance cost (w l random numbers and permutations required

2)

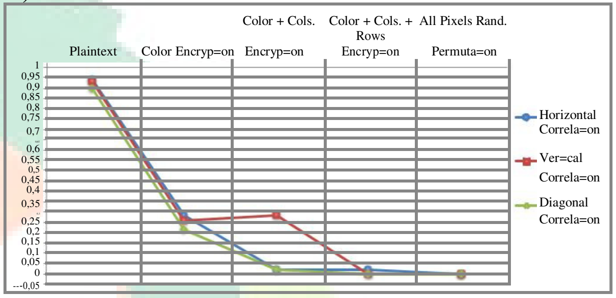

Fig. 2: Average Vertical, Horizontal and Diagonal Correlation between all pixels of all images in the Wang Dataset

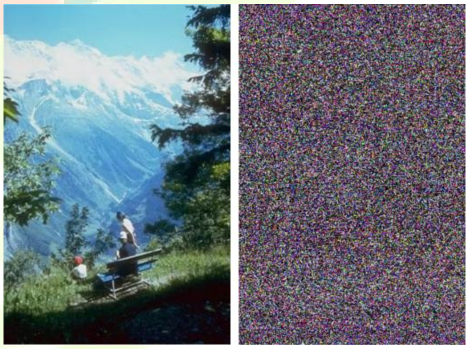

Fig. 3: Example of IES-CBIR image encryption

instead of w + 1) and with little gain as correlation is already very close to absolute 0. To conclude this section, we present in Fig. 3 an example of IES-CBIR encryption applied on an image from the used dataset.

VI. Conclusions

In this paper we have proposed a novel cryptographic scheme, named IES-CBIR, to support privacy-preserving out-sourcing of storage and search/retrieval of images in the encrypted domain, implemented with readily-available storage clouds. Key to the design of IES-CBIR is the observation that in images, color information can be separated from texture information, enabling the use of different encryption techniques with different properties for each one. Leveraging IES-CBIR, we designed a secure framework focused on dynamic, multi-tenant image outsourcing, focused on retrieval scenarios, where the reduction of network traffic and client's computational overhead were central aspects of the design. To validate our proposal in terms of security, efficiency, and precision, we presented a formal analysis and experimental evaluation of a prototype system leveraging IES-CBIR, comparing the results with relevant alternatives from the literature. Obtained results show that our approach is secure and doesn't leak private information of involved parties while having good retrieval precision, and better performance with lower computational overhead imposed on clients.

This work's main focus was on privacy issues. However one also has to consider issues related to reliability and availability as complementary dependability criteria. While we don't address these explicitly in this paper, we plan to focus on them in future work, by combining different cloud in-frastructures with independent administration and independent failures, exploring diversity to improve dependability criteria, fault-tolerance and intrusion-resilience.

References

[1] M. Meeker and L. Wu, "Internet Trends," in D11 Conf., 2013.

[2] Global Web Index, "Instagram tops the list of social network growth," http://blog.globalwebindex.net/instagram-tops-list-of-growth, 2013.

[3] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina, "Controlling data in the cloud: outsourcing computation without outsourcing control," in CCSW'09, 2009.

[4] D. Rushe, "Google: don't expect privacy when sending to Gmail," http://tinyurl.com/kjga34x, 2013.

[5] G. Greenwald and E. MacAskill, "NSA Prism program taps in to user data of Apple, Google and others," http://tinyurl.com/oea3g8t, 2013.

[6] A. Chen, "GCreep: Google Engineer Stalked Teens, Spied on Chats," http://gawker.com/5637234, 2010.

[7] J. Halderman and S. Schoen, "Lest we remember: cold-boot attacks on encryption keys," in Commun. ACM, vol. 52, no. 5, 2009, pp. 91–98.

[8] P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, and M. Walfish, "Depot: Cloud Storage with Minimal Trust," ACM Trans. Comput. Syst., vol. 29, no. 4, pp. 1–38, Dec. 2011.

[9] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the AES circuit," in CRYPTO 2012. Springer, pp. 850–867.

[10] C.-Y. Hsu, C.-S. Lu, and S.-c. Pei, "Image Feature Extraction in Encrypted Domain With Privacy-Preserving SIFT," IEEE Trans. Image Process., vol. 21, no. 11, pp. 4593–4607, 2012.

[11] P. Zheng and J. Huang, "An efficient image homomorphic encryption scheme with small ciphertext expansion," in Proc. 21st ACM Int. Conf. Multimed. - MM '13. ACM Press, Oct. 2013, pp. 803–812.

[12] W. Lu, A. L. Varna, A. Swaminathan, and M. Wu, "Secure image retrieval through feature protection," 2009 IEEE Int. Conf. Acoust. Speech Signal Process., pp. 1533–1536, Apr. 2009.

[13] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, "Enabling Search over Encrypted Multimedia Databases," in IS&T/SPIE Electron. Imaging, Feb. 2009, pp. 725 418–725 418–11.

[14] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries," IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 9, pp. 947–963, 2001.

[15] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in Proc. 2004 ACM SIGMOD Int. Conf. Manag. data. ACM, 2004, pp. 563–574.

[16] H. Muller,¨ W. Muller,¨ D. M. Squire, S. Marchand-Maillet, and T. Pun, "Performance evaluation in content-based image retrieval: overview and proposals," pp. 593–601, 2001.

[17] X. Yuan, X. Wang, C. Wang, A. Squicciarini, and K. Ren, "Enabling Privacy-preserving Image-centric Social Discovery," in ICDCS. IEEE, 2014, pp. 198–207.

[18] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," in Proc. 13th ACM Conf. Comput. Commun. Secur. - CCS'06, 2006, pp. 79–88.

[19] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 8, pp. 1467–1479, Aug. 2012.

[20] S. Kamara and C. Papamanthou, "Parallel and dynamic searchable symmetric encryption," Financ. Cryptogr. Data Secur. FC, pp. 1–15, 2013.

[21] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 1, pp. 222–233, Jan. 2014.

[22] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in S&P. IEEE, 2000, pp. 44–55.

[23] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," EUROCRYPT'99, pp. 223–238, 1999.

[24] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in Adv. Cryptol. Springer, 1985, pp. 10–18.

[25] J. R. Troncoso-Pastoriza and F. Perez-Gonzalez, "Secure signal processing in the cloud: enabling technologies for privacy-preserving multimedia cloud processing," IEEE Signal Process. Mag., vol. 30, no. 2, pp. 29–41, Mar. 2013.

[26] W. Stallings, Cryptography and Network Security, Principles and Practices, 4th ed. Pearson International Edition, 2006.

[27] M. J. Swain and D. H. Ballard, "Color indexing," Int. J. Comput. Vis., vol. 7, no. 1, pp. 11–32, 1991.

[28] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers, "Format-preserving encryption," in Sel. Areas Cryptogr. Springer, 2009, pp. 295–312.

[29] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in CVPR, vol. 2. IEEE, 2006, pp. 2161–2168.

[30] E. Stefanov, M. Van Dijk, E. Shi, C. Fletcher, L. Ren, X. Yu, and S. Devadas, "Path oram: An extremely simple oblivious ram protocol," in CCS. ACM, 2013, pp. 299–310.

[31] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in Comput. Vision-ECCV. Springer, 2008, pp. 304–317.

[32] Z.-l. Zhu, W. Zhang, K.-w. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," Inf. Sci. (Ny)., vol. 181, no. 6, pp. 1171–1186, 2011.