Content-Based Retrieval in Cloud Image Repositories

M.Tech. Student Krishna S.S., Professor Asha A.S.
Department of Computer Science and Engineering,
Lourdes Matha College of Science and Technology, Trivandrum, Kerala, India,
krishnamalu53@gmail.com, Asha.AS@lmcst.ac.in

Abstract -Multimedia visual data have been increasing to store the cloud computing in recent years, following the emergence of many high interactive multimedia services and applications for mobile devices in both personal and corporate scenarios.Thishasbeenakeydrivingfactorforthe adoption of cloud-based data outsourcing solutions. Framework may fully supported in multi media data ex(images,video,audio)themain activity on this framework it may chunk the large amount of the data and using the Encryption for each chunks. The security process may high at this frame work high interactive multimedia services and applications for mobile devices in both personal and corporate scenarios. This has been a key driving factor for the adoption of cloud-based data outsourcingsolutions.

Keywords- IES-CBIR Framework.

I.INTRODUCTION

Visual data is responsible for one of the largest shares of global Internet traffic in both corporate and personal use scenarios. The amount of images, graphics, and photos being generated and shared every day, especially through mobile devices, is growing at an ever increasing rate. The storage needs for such large amounts of data in resource-constrained mobile devices has been a driving factor for data outsourcing services such as the ones leveraging Cloud Storage and computing solutions. Such services (e.g. Instagram and Flickr) have been reported to be among the largest growing internet services.

Addition- ally, the availability of large amounts of images in public and private repositories also leads to the need for content- based search andretrieval solutions (CBIR) Despite the fact that data outsourcing, especially to cloud computing infrastructures, seems a natural solution to sup-port large scale image storage and retrieval systems, it also raises new challenges in terms of data privacy control. This is a consequence of outsourcing data, which usually implies releasing control (and sometimes even effective ownership) over it. Recent incidents have provided clear evidence that privacy should not be expected to be preserved by cloud providers Furthermore, malicious or simply careless system administrators working for the providers have full access to data on the hosting cloud machines.

Finally, external hackers can exploit software vulnerabilities to gain unauthorized access to servers The recent incident with the Cloud image storage service and celebrity photo leakage illustrates the danger these threats pose For cloud-based visual data stores. The conventional approach to address privacy in this context is to encrypt sensitive data before outsourcing it and run all computations on the client side. However this imposes unacceptable client-overhead, as data must continuously be downloaded, decrypted, processed, and securely re-uploaded. Many applications cannot cope with this overhead, particularly online and mobile applications operating over very large datasets such as image repositories with CBIR services. A more viable approach would be to outsource computations and perform operations over the encrypted data on the server side. Existing proposals in this domain remain largely unpractical,namely those requiring fully holomorphic encryption, which is still computationally too expensive.

Nonetheless, partially holomorphic encryption schemes and symmetric- key solutions (or property-preserving schemes) supporting specific search patterns are interesting alternatives, yielding more practical results while providing a good tradeoff between security, privacy, and usability. Unfortunately, even these solutions are too computationally complex for wide adoption, particularly regarding the support of privacy-preserving CBIR over large-scale, dynamically updated image repositories. This prohibitive complexity is even further exacerbated if we consider mobile (resource constrained) clients, which are already responsible for more than 30% of internet traffic To address these challenges it propose a new se- cure framework for privacy preserving outsourced storage, search, and retrieval of large- scale, dynamically updated image repositories.

The proposal on IES-CBIR, a novel Image Encryption Scheme (IES) with Content- Based Image Retrieval (CBIR) properties. Key to the design of IES- CBIR is the observation that in image processing, distinct feature types can be separated and encrypted with different cryptographic algorithms. As an example, image color and texture data can be separated in such a way that CBIR in the encrypted domain can be performed on one feature type while the other remains fully randomized and protected with semantically-secure cryptography. Following this observation, and considering that texture is usually more relevant than color in object recognition in IES-CBIR we make the following security-oriented tradeoff: it choose to privilege the protection of image contents, by encrypting texture information with probabilistic (semantically-secure) encryption then itcontrollably relax the security on color features,by using deterministic encryption on image color information.

This methodology allows privacy- preserving CBIR based on color information to be performed directly on the outsourced servers with high security guarantees notably, the solution allows outsourcing servers to generate and update an index used to efficiently process and reply to queries, a task that in many state of art solutions must be managed by client devices. It show further ahead in the paper, new methodology leads to optimized computation and communication overheads with non-negligible impact on system performance and mobile battery consumption.

In summary, this paper makes the following contributions formally define IES-CBIR, a novel Image Encryption Scheme with Content-Based Image Retrieval properties, and propose an efficient construction that achieves its functionality show how to design an out-sourced image storage, search, and retrieval framework by leveraging IES-CBIR to avoid most heavy computations to be performed by the client (i.e. indexing of dynamically added/updated images), hence circumventing performance pitfalls that exist in current state of art proposals formally prove the security of framework and IES-CBIR experimentally show that when compared with competing alternatives framework provides increased scalability, performance (from user’s perspective), and lower bandwidth consumption, allowing client applications to be increasingly lightweight and mobile And finally it show that the retrieval precision and recall of the proposed solution is on par with the current state of art.

The work presented in this paper was first introduced in Here it extend exposition significantly by discussing two use cases where IES-CBIR and the proposed framework can be applied with immediate benefits. Then further provide a complete formal security evaluation of proposals and a performance analysis of the search operation of the framework in comparison with relevant previous works.

II. DOMAIN EXPLANATION

In Computer science, cloud computing describes a type of outsourcing of computer services, similar to the way in which electricity supply is outsourced. Users can simply use it. They do not need to worry where the electricity is from, how it is made, or transported. Every month, they pay for what they consumed. The idea behind cloud computing is similar: The user can simply use storage, computing power, or specially crafted development environments, without having to worry how these work internally. Cloud computing is usually Internet-based computing. The cloud is a metaphor for the Internet based on how the internet is described in computer network diagrams; which means it is an abstraction hiding the complex infrastructure of the internet.

It is a style of computing in which IT-related capabilities are provided “as a service”, allowing users to access technology-enabled services from the Internet ("in the cloud") without knowledge of, or control over the technologies behind these servers. According to a paper published by IEEE Internet Computing in 2008 "Cloud Computing is a paradigm in which information is permanently stored in servers on the Internet and cached temporarily on clients that include computers, laptops, handhelds, sensors, etc." Cloud computing is a general concept that utilizes software as a service (SaaS), such as Web 2.0 and other technology trends, all of which depend on the Internet for satisfying users' needs. For example, Google Apps provides common business applications online that are accessed from a web browser, while the software and data are stored on the Internet servers.

Cloud computing is often confused with grid computing (a form of distributed computing whereby a "super and virtual computer" is composed of a cluster of networked, loosely-coupled computers, working together to perform very large tasks), utility computing (the packaging of computing resources, such as computation and storage are provided as a measured service that have to be paid similar to a traditional public utility such as electricity) and autonomic computing (computersystems capable of self-management).Many cloud computing deployments are powered by grids, have autonomic characteristics and are billed like utilities, but cloud computing can be seen as a natural next step from the grid- utility model. Some successful cloud architectures have little or no centralized infrastructure or billing systems at all including peer-to-peer networks like Bit Torrent and Skype. The majority of cloud computing infrastructure currently consists of reliable services delivered through data centers that are built on computer and storage virtualization technologies.

The services are accessible anywhere in the world, with The Cloud appearing as a single point of access for all the computing needs of consumers. Commercial offerings need to meet the quality of service requirements of customers and typically offer service level agreements. Open standards and open source software are also critical to the growth of cloud computing. As customers generally do not own the infrastructure or know all details about it, mainly they are accessing or renting, so they can consume resources as a service, and may be paying for what they do not need, instead of what they actually do need to use. Many cloud computing providers use the utility computing model which is analogous to how traditional public utilities like electricity are consumed, while others are billed on a subscription basis. By sharing consumable and "intangible" computing power between multiple "tenants", utilization rates can be improved (as servers are not left idle) which can reduce costs significantly while increasing the speed of application development.

A side effect of this approach is that "computer capacity rises dramatically" as customers do not have to engineer for peak loads. Adoption has been enabled by "increased high-speed bandwidth" which makes it possible to receive the same response times from centralized infrastructure at other sites. Cloud computing is being driven by providers including Google, Amazon.com, and Yahoo! as well as traditional vendors including IBM, Intel, Microsoft and SAP. It can adopted by all kinds of users, be they individuals or large enterprises. Most internet users are currently using cloud services, even if they do not realize it. Webmail for example is a cloud service, as are Facebook and Wikipedia and contact list synchronization and online databackups. The underlying concept dates back to 1960 when John McCarthy expressed his opinion that "computation may someday be organized as a public utility" and the term Cloud was already in commercial use in the early 1990s to refer to large ATM networks. By the turn of the 21st century, cloud computing solutions had started to appear on the market, though most of the focus at this time was on Software as a service.

III. LITERATURE SURVEY

Previous proposals for supporting outsourced storage, search, and retrieval of images in the encrypted domain can be broadly divided in two classes: those based on Searchable Symmetric Encryption (SSE) techniques and those based on Public-Key partially-Homomorphic Encryption (PKHE). SSE has been widely used in the past by the research community, especially for text data.

In the image domain, even though not identified as SSE schemes, multiple systems use the same (or similar) techniques for image search/retrieval. For simplicity, we refer to these as SSE-based solutions. In SSEbased solutions, clients process their data before encrypting and outsourcing it to the Cloud. From this processing, an index is created, encrypted, and stored in the outsourced infrastructure, which allows clients to search their data efficiently and in a secure way. Data is typically encrypted with probabilistic symmetric-key encryption schemes, while the index is protected through a combination of probabilistic and deterministic (or even order preserving) encryption. Unfortunately, SSE-based approaches in general share the following limitations:

(i) Clients either require a trusted proxy or have to index their images (and encrypt that index) locally, which entails the use of additional computational power on their side and limits the practicality of such solutions for resource-constrained mobile devices. This effect is further exacerbated when considering dynamic scenarios, where images are constantly being added, updated, and removed. In such dynamic scenarios, SSE solutions usually require multiple rounds of communication for updating imagerepositories and their indexes.

For instance, a previous approach by Lu et al. uses repository-wide statistics (e.g. inverse-document frequencies), which change as the repositories are updated and thus force the reconstruction and re-encryption of the index, requiring clients to download and decrypt the full contents of the repository. Additionally index values are encrypted with an order-preserving encryption scheme that depends on plaintext domain distribution. With multiple updates this distribution changes, again requiring the re-construction and re-encryption of the index. This is an important issue from a security viewpoint. Other approaches from the literature require multiple rounds of communication for performing such operations;

(ii) Clients have to transfer additional data to the cloud, instead of just uploading images, they also have to retrieve and re-upload their encrypted index with each repository update. This leads to additional bandwidth usage, negatively impacting the latency of storage operations as perceived by users and being a particular issue for cloud backed deployments;

(iii) As SSE works use deterministic tokens to provide their functionality with practical performance. Deterministic tokens include unique document identifiers and deterministic encryptions of keywords. image (in case of similarity/ranked search); and which images (previously searched) are similar to a new image being inserted. These leakage patterns result in exposing as much information as a fully deterministic encryption scheme, albeit with much higher computational overhead. This is demonstrated in and is particularly evident in long-lived system with many queries being executed concurrently and all index entries being accessed. Nonetheless, the reader should note that deterministic schemes (and SSE-based schemes with the referred leakages) can still be provably-secure, as long as the higher-level applications leveraging them control the amount of background information leaked to adversaries (including plaintext distribution knowledge). The alternatives to SSE that can be found in the literature are based on public-key partially-homomorphic encryption

(PKHE) schemes such as Paillier or ElGammal, which allow additions and multiplications on the encrypted domain, respectively. In these approaches, clients encrypt images pixel by pixel witha PKHE scheme, allowing the cloud to process and index encrypted images on their behalf and thus avoiding many of the practical issues of SSE-based solutions. Unfortunately, PKHE works present much higher time and space complexities.

For instance, Hsu et al. proposed a high-precision CBIR algorithm in the encrypted domain, by resorting to the Paillier cryptosystem. However, their approach results in significant cipher text expansion (for a secure key size of 1024 bits, each pixel is transformed from its traditional 24 bits representation into 2048 cipher text bits), slow encryption and decryption times, and in limited scalability (the "cipher text blowup" problem,i.e. when cipher text values reach their arithmetic group limits through multiple multiplications). Furthermore their work was later shown to be either insecure or computationally intractable for a typical cloud server.

Zheng et al. proposed a variant of that work, overcoming some of its drawbacks by replacing Paillier cipher texts with pointers to a cipher text table with all possible cipher text pixel values. This approach can potentially reduce the number of encryption operations and minimize cipher text expansion in some use cases. However Paillier encryptions still present a significant computational overhead, limiting the practicality of the approach.

Aside from the SSE and PKHE research directions, there have been other works following similar approaches to what they propose in this paper, although for different purposes. An example is the work by Nourian et al. which aims at providing privacy-preserving single image template matching performed by third-party clouds. However, this work doesn't support large-scale repositories, as it only allows linear searching, requires the template being matched to be re-encrypted for comparison with each different image in a repository, and requires the availability of public images as noise for encryption which can be easily found by an attacker using popular high-availability repositories for dictionary attacks (or by tracking users' traffic).

Another example is the more theoretical work by Chase et al. [32], which proposes a set of algorithms for the encryption of several data structures (including matrix-based datatypes such as images), while enabling queries to be performed over the cipher text. Their main motivation is to extract partial information about a single encrypted data object (such as the color of a given pixel in an image). In our proposal we focus on allowing the generation of indexes over large collections of encrypted images by an untrusted third party and the efficient and precise resolution of user queries over these large image collections.

IV. METHODOLOGY

Proposed System : IES-CBIR Framework the framework may secured to store the data and retrieve the data in cloud computing the data implement the concept (Framework) based on security purpose the IES-CBIR Framework may fully supported in multimedia data ex(images , video , audio)the main activity on this framework it may chunk the large amount of the data and using the Encryption for each chunks and generate a the chunk encryption key for user's the user may access the decryption data at using this chunk key's so the security process may high at this frame work Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in either privately owned, or third-par that may be located far from the user–ranging in distance from across a city to across the world.

Time and money on computer infrastructure. The amount of images, graphics, and photos being generated and shared every day, especially through mobile devices, is growing at an ever increasing rate. The storage needs for such large amounts of data in resource- constrained mobile devices has been a driving factor for data outsourcing services such as the ones leveraging Cloud Storage and computing solutions. Such services (e.g. Instagram and Flickr) have been reported to be among the largest growing internet services. Addition- ally, the availability of large amounts of images in public and private repositories also leads to the need for content-based search and retrieval solutions (CBIR) Despite the fact that data outsourcing, especially to cloud computing infrastructures, seems a natural solution to sup-port large scale image storage and retrieval systems, it also raises new challenges in terms of data privacy control.

Advantages:

1.The system may using the IES-CBIR, a novel Image Encryption Scheme that exhibits Content-Based Image Retrieval properties.

2.The framework enables both encrypted storage and searching using Content-Based Image Retrieval queries while preserving privacy against honest-but-curious cloud administrators.

3.Built a prototype of the proposed framework, formally analyzed and proven its security properties, and experimentally evaluated its performance and retrieval precision.

1. System Architecture:

System model and architecture envisioned for using our framework and IES-CBIR. In this model, It consider two main entities: the cloud and (multiple) users (Figure 1). Images are outsourced to repositories that reside in the cloud. Each repository is used by multiples Users, where they can both add their own images and/or search using a query image. Users can also request access to stored images from their creators/owners. The objective is to ensure the privacy of users, hence all data sent to the cloud is encrypted. Each repository is created by a single user.

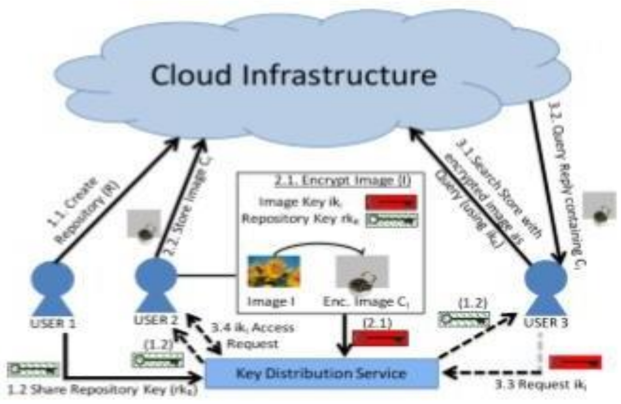


Fig. 1 System model overview of the proposed framework.

Upon the creation of a repository, a new repository key is generated by that user and then shared with other trusted users, allowing them to search on the repository and add/update images. To add/update images (but not search), a user further needs an image key generated for that image. Image keys are kept secret by their users, meaning that even users capable of searching in a repository (i.e. with access to the repository key) willneed to ask the owners of specific images for access to them. Note that using specific keys per-image should be seen as an option in our framework, i.e. if the users of a repository prefer to avoid further key management overhead and are willing to sacrifice fine-grained access control, they can use the same image key for all images in a repository.

When the cloud receives an encrypted image for storage it extracts its relevant features (in our framework, it use global color features) and indexes the image based on these features. The same action is performed for a query image, which after being encrypted by a user with a repository key, is then processed by the cloud and has its features extracted and matched with the repository's index. The reply to a query will contain k (a tunable system parameter) number of encrypted images and respective metadata, which include each image's id and the id of the user that owns each of the images. To fully decrypt and access the contents of an image, besides the repository key, the querying user will further require the image key for that specific image.

It should be noted that all key sharing interactions can be done by resorting to a key distribution service, implemented either in a centralized way (using protocols such as Kerberos or in a distributed fashion (through asynchronous communications or protocols such as Diffie-Hellman). User authorization and revocation can also be easily achieved, for instance, through the sharing (and refreshment when user revocations are issued) of repository-specific tokens between trusted users, and its request in the framework operations. Nonetheless e find these discussions to be orthogonal to the main focus of this contribution, as the mechanisms involved can be easily integrated into our framework.

V. A PRIVACY-PRESERVING CBIR FRAMEWORK

The main component on the users' side leverages a novel cryptographic scheme specifically designed for images and privacy preserving CBIR, dubbed IES-CBIR. Before describing IES-CBIR in detail, we give a definition of image privacy that underlines our work. Informally, it define image privacy as the ability to keep the contents of an image secret to public (or simply unauthorized) disclosure. Generally speaking, image contents are characterized by the combination of its color and texture information. These two components form what one can readily identify in an image: objects, people, etc. As such, safeguarding image privacy entails preventing unauthorized entities from recognizing objects in those images. The further remark that image color and texture informations can be separated from each other. Indeed, color information is given from pixel color values in the different channels of a particular color model; while texture information is given by the (relative) position of pixels and strong color changes across neighboring pixels.

It also remark that texture information is usually more relevant in images for object recognition. Finally, conclude that no sub- component alone (i.e. color or texture information) can be used to infer the precise contents of an image, as color information on itself is usually ambiguous (e.g. strong blue can translate into sky, ocean, etc.) and texture information depends not only on pixel positions but also on their color values. These observations are further supported by the most recent works in image reconstruction , which not only depend on local features extracted from sub-segments of images (in this work focus on global features extracted from each image as a whole), but also on those local features not being encrypted.

Leveraging the previous definition and remarks the design IESCBIR, an image encryption scheme that separates color from texture information, applying different encryption techniques for protecting each. Emphasizing that texture is usually more relevant than color for object recognition, the design IES-CBIR to protect image texture with probabilistic encryption and color information with deterministic encryption. This way, content-based image indexing and retrieval, based on color information, can be performed on the cloud servers in a privacy-preserving way and without intervention of users, while texture information remains protected with the highest level of security.

1. Definition 1 (IES-CBIR). An Image Encryption Scheme with CBIR properties is a tuple (GENRK, GENIK, ENC, DEC, TRPGEN) of five polynomial-time algorithms run by a user, where:

- **GENRK(sprk):** is a probabilistic algorithm that takes as input the security parameter sprk∈ N and generates a repository key rk;
- **GENIK(spik):** is a probabilistic algorithm that takes as input the security parameter spik ∈ N and generates an image key ik;
- **ENC(I,rk,ik):** takes as input an image I and the cryptographic keys {rk,ik}, returning an encrypted image CI;
- **DEC(CI,rk,ik):** takes as input an encrypted image CI and keys {rk,ik}, returning the decrypted image I;
- **TRPGEN(Q,rk):** takes as input a query image Q and a repository key rk, returning a searching trapdoor CQ;

2. Key Generation

IES-CBIR works with two different types of cryptographic keys, repository keys (rk) and image keys (ik), which are generated by the GENRK and GENIK algorithms respectively. Repository keys deterministically map a pixel's color value in a color channel to some new random value4. To prevent images from increasing in size after encryption (i.e. prevent cipher text expansion), encrypted pixels should be in the same range of values as their original plaintexts (usually 8 bits per color channel).

As such, we build repository keys in IESCBIR by performing random permutations of all possible pixel color values in each color channel. Leveraging the HSV color space ((H) hue, (S) saturation, (V) value/brightness), we perform three independent random permutations of the values in range [0..100]. This range represents all possible color values in the HSV color space, and each permutation is used for a different color channel, resulting in 3 repository subkeys: rkH, rkS, rkV . Permutations are performed by a Pseudo-Random Generator (PRG) G parameterized the encrypted domain without alterations, including image indexing, searching, and compressing operations.

3. Encryption

Image encryption in IES-CBIR is achieved through two main steps and a final (optional) step: i) pixel color values encryption, ii) pixel positions permutation, and iii) image compression. The goal of the first step is to protect image color features, through the application of a Pseudo-Random Permutation (PRP) P on all pixel color values. Although wecould use a standard PRP construction to instantiate P (such as an AES-based PRP), It chose to conceive a specific color-domain PRP, allowing us to preserve the format of encrypted images. Our construction encrypts pixel color values by deterministically replacing them, in each color channel, using repository key.

$rk = \{rk_H, rk_S, rk_V\}.$

This step of encryption securely hides color values of encrypted pixels. However, due to the deterministic properties of P (a requirement to enable CBIR in the encrypted domain), patterns present in the original image (which denote its texture) will remain visible. To fully protect image contents, we rely on a second probabilistic step in our encryption algorithm: (pseudo)random pixel position permutation, through pixel rows and columns shifting. In this step a PRG G is instantiated with a previously generated image key ik (operation GENIK above) as cryptographic seed. Then, for each pixel column we request from G a new pseudorandom value r between 1 and the image height, shifting that column r positions downward, overflowing to its beginning. After all columns have been randomly shifted, we repeat the procedure for the rows (with random values ranging between 1 and the image width). where w and h are, respectively, the width and height of image I. Note that this encryption algorithm has no ciphertext expansion (i.e, after encryption the image has the same width and height as before).

$$C_I(x,y) \leftarrow C_I(x,(y+r) \bmod h) : \forall x \in \{1,\dots,w\},$$

$$C_I(x,y) \leftarrow C_I((x+r) \bmod w,y) : \forall x \in \{1,\dots,w\},$$

The above step is probabilistic, as each new image will have a new pseudorandomly generated ik, even if the same image is stored multiple times with different names (if the same image key is used for all images, then a random iv must also be used as input to G). Moreover, this step effectively hides existing texture patterns in the image, making it computationally unfeasible to extrapolate correlations between plaintext and

4. Decryption

The decryption algorithm applies the different steps of encryption in the inverse order, or more formally, through the ordered application of the transformations denoted by Equations. 5, 6, and 7 (after decompressing the ciphertext if required). Note that the r random values must be generated in the same order as in the encryption

$$C_I((x+r) \bmod w,y) \leftarrow C_I(x,y) : \forall x \in \{1,\dots,w\}, \forall y \in \{1,\dots,h\} ($$

$$C_I(x,(y+r) \bmod h) \leftarrow C_I(x,y) : \forall x \in \{1,\dots,w\}, \forall y \in \{1,\dots,h\}$$

5. Searching-Trapdoor Generation

The TRPGEN algorithm generates searching trapdoors that users can leverage to search over image repositories.

Trapdoor generation requires a query image Q as input, as well as the repository key rk. This means that users with access to rk will be able to access color values of all images stored in that repository. However, users can't access texture information (and hence full image contents) without the corresponding image keys, and can't use rk to search other repositories. Given rk, the TRPGEN algorithm operates in a similar fashion to the ENC algorithm (Equation 8, where the image key is substituted by a new ik randomly generated for the query).

This means that searching trapdoors are also decryptable, and can be stored in the repositories as new images as long as users locally save the image keys generated for the queries. Their representation as color histograms. For each encrypted image and each HSV color channel, the cloud server builds a color histogram by counting the number of pixels in each intensity level. In our model, this yields 3 color histograms with entries in range [0,100], which are the admissible values for each HSV channel (i.e. each histogram has 101 entries). Upon extracting these features, the cloud can perform feature indexing to speedup query execution. In this work, then use the Bag-Of-Visual-Words (BOVW) representation to build a vocabulary tree and an inverted list index for each repository.

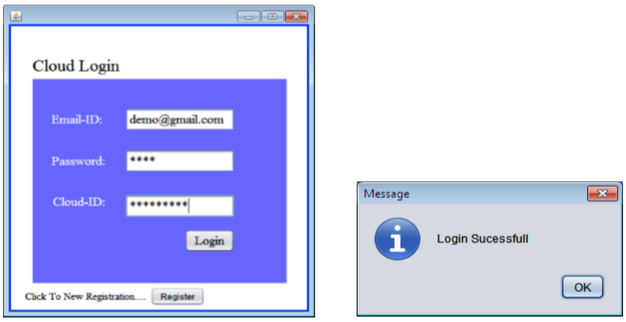
It choose this approach for indexing as it shows good search performance and scalability properties. In the BOVW model, feature-vectors are hierarchically clustered (for instance, using the k-means algorithm) into a vocabulary tree (also known as codebook), where each node denotes a representative feature-vector in the collection and leaf nodes are selected as the most representative nodes (called visual words). This clustering step requires a training dataset, so in the prototype implementation of our framework based on IESCBIR, we request an initial image collection from users when creating a new repository. After the creation of the codebook, additional images can be stored dynamically by hierarchically stemming them against it. This stemming returns the closest visual words to the image, according to some distance function (in our prototype we use the Hamming Distance). Finally, the cloud server builds an inverted list index, with all visual words as keys and the list of images most close to them (plus a frequency score) as values.

This type of list is known as a Posting List. After processing and indexing encrypted images, the cloud server can receive search requests from users, through the submission of search trapdoors for some query images of their choice. When a new search trapdoor is received, the cloud server extracts its color feature-vectors and finds their closest visual words by stemming them against the codebook. The query's visual words are used to access the repository's index, obtaining the corresponding posting lists in the process. Then, for each image referenced in at least one posting list, a search score is calculated for that image .Finally, the cloud returns the top k images to the user, according to their scores (k is a configurable parameter). The BOVW approach guarantees that only the most relevant images (a fraction of the repository) have to be compared in the scoring step (key to ensuring scalability). After receiving search results, users can explicitly request full access to images by requesting the corresponding image keys from their owners.

VI.IMPLIMENTATION

IES-CBIR Framework the framework may secured to store the data and retrieve the data in cloud computing the data implement the concept (Framework) based on security purpose the IES-CBIR Framework may fully supported in multimedia data ex(images , video , audio)the main activity on this framework it may chunk the large amount of the data and using the Encryption for each chunks and generate a the chunk encryption keyfor user's the user may access the decryption data at using this chunk key's so the security process may high at this frame work Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in either privately owned, or third-par that may be located far from the user–ranging in distance from across a city to across the world.

The user login and registration process is a security process it may avoid the UN authentication process so the cloud user may only use this process.The new user may register the all user information in to the cloud server. The cloud server may generate the cloud user key. the user may login to the cloud the cloud key may necessary to login so the cloud key may important in our process this key may generated by cloud server.



Cloud Login

Email-ID: demo@gmail.com

Password: ****

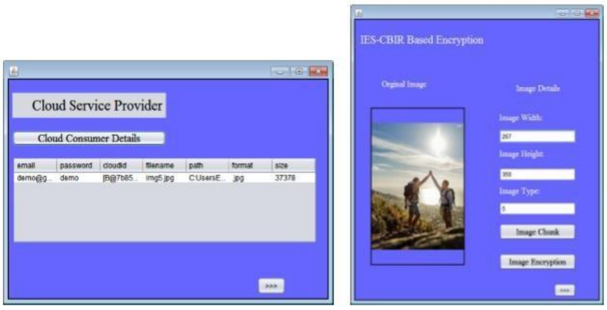
Cloud-ID: *********

Click To New Registration....

Login Sucessfull

Fig.2 Key may generated by cloud server.

This module is an user module the cloud authentication process may completed the user may upload the multimedia files this files may up load the cloud server.
The user may select any media files ex (image, video, audio) files etc..., the files all information may completely store into a cloud server's.



Cloud Service Provider

Cloud Consumer Details

email	password	cloudid	filename	path	format	size
demo@g...	demo	[B@7b85...	img5.jpg	C:\UsersE...	jpg	37378

IES-CBIR Based Encryption

Orginal Image:

Image Details

Image Width: 267

Image Height: 350

Image Type: 5

Fig.3 All information may completely store into a cloud server’s.

The CSP (Cloud service Provider) may provide cloud key. The user data may store for cloud server the csp provider encrypt the data’s at based on IES-CBIR Framework then the encryption may completed the image may split into an chunks and next the image may encrypted the decryption key may send at cloud user.

VII.CONCLUSION

The proposed framework for the privacy-preserving outsourced storage, search, and retrieval of large-scale, dynamically updated image repositories, where the reduction of client overheads is a central aspect. In the basis of the framework is a novel crypto-graphic scheme, specifically designed for images, named IES- CBIR. Key to its design 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, and allowing privacy-preserving Content-Based Image Retrieval to be performed by third-party, untrusted cloud servers.

REFERENCES

[1] Bernardo Ferreira, Joao Rodrigues, Joao Leitao and Henrique Domingos, “Practical Privacy-Preserving Content- Based Retrieval in Cloud Image Repositories ” IEEE Transactions on Cloud Computing , DOI 10.1109/TCC.2018.2669999.

[2] Global Web Index, “Instagram tops the list of social network growth,” http://tinyurl.com/hnwwlzm, 2013.

[3] C. D. Manning, P. Raghavan, and H. Schutze,¨ An Introduction to Information Retrieval. Cambridge University Press,2009, vol. 1.

[4] 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.

[5] D. Rushe, “Google: don’t expect privacy when sending to Gmail,” http://tinyurl.com/kjga34x, 2013.

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

[7] A. Chen, “GCreep: Google Engineer Stalked Teens, Spied onChats,” http://gawker.com/5637234, 2010.

[8] J. Halderman and S. Schoen, “Lest we remember: cold-boot attacks on encryption keys,” in Commun. ACM, vol. 52, no. 5, 2009.

[9] National Vulnerability Database,“CVE Statistics,” http://web.nvd.nist.gov/view/vuln/statistics, 2014.

[10] D. Lewis, “iCloud Data Breach: Hacking And CelebrityPhotos,”https://tinyurl.com/nohznmr, 2014.

[11] 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.

[12] C. Gentry, S. Halevi, and N. P. Smart, “Homomorphic evaluation of the AES circuit,” in CRYPTO’12. Springer, 2012, pp. 850–867.

[13] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in EUROCRYPT’99, 1999, pp. 223–238.

[14] T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” in Adv. Cryptol. Springer, 1985.

[15] 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.

[16] P. Zheng and J. Huang, “An efficient image homomorphic encryption scheme with small ciphertext expansion,” in MM’13, 2013