# Review on A Privacy-preserving and Copy-deterrence Content-based Image Retrieval Scheme in Cloud Computing

**[1]Priyanka R. More, [2]Dr. Khan Rahat Afreen**

[1]Research Scholar, [2]Associate Professor
Marathwada Shikshan Prasark Mandal's Deogiri Institute of Engineering & Management Studies,
Aurangabad Maharashtra state, India 2017-2018

***Abstract*: With the increasing importance of people's everyday imagery, Content Based Image Retrieval has been extensively studied. Compared to text documents, images take up more storage space. Therefore, maintenance is a typical example of cloud storage. For privacy purposes, sensitive images, such as medical images and personal images, must be encrypted prior to outsourcing, which prevents CBIR technology in plain text domains. We have studied the process that supports CBIR over encrypted images without deluting sensitive information to the cloud server. Most common approach is the feature vector is separated to display the matching image. Subsequently the process contains, a pre-filter table that is created by a resident combination to optimize the search. In addition, the feature vector is protected by a safe kNN algorithm and the image pixels are encoded by standard stream numbers. We have studied the protocol that uses watermarks to inhibit such illegal distribution. In protocols that use watermarks, unique watermarks are embedded directly into the images encrypted by the cloud server before the images are sent to the user.**

***Index Terms*: Searchable encryption, content-based image retrieval, secure kNN, CBIR,LES-CBIR, copy deterrence, watermark.**

## I Introduction

Demand for efficient storage and retrieval services increases with the addition of a large database of images in all areas. While after more than twenty years of development, Content Based Image Retrieval techniques showcase the potential of many real-world applications. For example, doctors can use CBIR to find similar patients and facilitate clinical decision-making.

Large image databases often contain millions of images. CBIR services are often complex to store and calculate. Cloud computing is an excellent opportunity for access to resources, computing and extensive data storage, which is an attractive alternative for storing images and images. CBIR Employment By outsourcing CBIR services to the cloud server, data owners are relieved of the need to maintain a local image database and interact with users. Privacy is becoming a major concern of CBIR outsourcing. For example, patients may not want to disclose their medical images to others unless the CBIR's specialized physician uses it to determine the problem. Two types of privacy threats First, a suspicious cloud server can be found on the owner database. Second, after receiving the image, the system may distribute these images to unauthorized persons.

Image data is part of one of the largest global Internet access situations in enterprise and personal use. [1] The number of graphics and images created and shared in daily life increases. Increasingly, the demand for such data storage is the driving force behind outsourcing services. Such services (such as Instagram and Flickr) have been reported as one of the most growing Internet services. [2] In addition, the availability of large images in state repositories. In addition, the private sector also makes use of search and retrieval solutions (CBIR) [3]. (Such as the use of a cloud infrastructure) seems to be a natural solution for storing and retrieving large image data. But it brings new challenges in terms of data control and privacy. This is a result of outsourced information, which usually refers to the release of control. [4] Recent news has provided clear evidence that it should not be expected to retain personal data from cloud providers [5], [6] in addition. Malicious administrators who work with providers have access to data on cloud hosting. [7] Finally, outside hackers can exploit software vulnerabilities to gain access to the server unauthorized without permission [9].

Recent incidents with iCloud image hosting services and celebrity image leaks [10] illustrate some of the importance of these threats for cloud storage. The general approach to identifying privacy in this context is to encrypt sensitive data before outsourcing and to run all calculations on the client side. However, the charge will cost customers [11] too much to download, decrypt, process, and upload again securely. Many applications are not able to cope with this cost, especially for online applications and mobile devices that use large data sets, such as image data storage with CBIR services. Proposals available in this domain remain in the theoretical domain, as the group requires full homomorphic coding, which is still too expensive [12]. However, some homomorphic encoding schemes [13][14][15][16] and symmetric-solution solutions designed to solve specific search problems.

Unfortunately, even these solutions are too complex for general use, especially with respect to CBIR support, which keeps personal information in a large image file that has been updated dynamically. And even more, if we consider that mobile customers already have more than 30% Internet access. By addressing these challenges, we offer a new security framework for efficient search and retrieval of search, recovery and recovery information. We have studied another Linear (Lucene) Encryption Scheme Content Based

Image Retrieval portfolio, a new Image Encryption Scheme that presents image recovery. The key to designing the LINEAR (LUCENE) ENCRYPTION SCHEME CONTENT BASED IMAGE RETRIEVAL is to observe that color images can be separated from surface data by using different encryption techniques, with different properties to protect these characteristics. [20] In LINEAR (LUCENE) ENCRYPTION SCHEME CONTENT BASED IMAGE RETRIEVAL, we grant the following priority: We choose to protect the content of the image. By surface coding. probabilistic encryption (semantically safe) [21]; Then we relaxed quite securely in the color properties by encryption. This image helps CBIR maintain its privacy, depending on the color information that will be processed directly on an external server with high security.

It should be noted that studied solution allows outsourced servers to generate and update the index of resources used to support queries, a task that in many next-generation solutions must be managed by the user's devices, which, as we will show later in the document, leads to an excess overhead computing and / or communication with impact on performance. This document makes the following contributions: (i) we formally define LINEAR (LUCENE) ENCRYPTION SCHEME CONTENT BASED IMAGE RETRIEVAL and propose an efficient construction that achieves its functionality; (ii) we show how to design an outsourced storage, search and recovery framework using Linear (Lucene) Encryption Scheme Content Based Image Retrieval to avoid most of the heavy calculations (that is, indexing of dynamically updated / added images) that the client will perform, thus avoiding the performance difficulties that exist in previous approaches [15] - [19].

## II Literature Review

The previous proposals to support external storage, search and retrieval of images in the encrypted domain can be divided into two classes: Approaches based on symmetric encryption (SSE) and partially homomorphic approaches of public key (PKHE). SSE has been widely used in the past by the research community, both for text [23] - [25] and for image search / retrieval. In SSE-based solutions, clients process and encrypt their data before outsourcing to the cloud. From this processing, an index is created, encrypted and stored in the outsourced infrastructure, which allows customers to search their data efficiently and safely. Normally, the data is encrypted with an encryption schema of symmetric probabilistic keys, while the index is protected by a combination of probabilistic and deterministic encryption (or even order preservation [26]). Unfortunately, SSE-based approaches generally share the following limitations: (i) clients require a reliable proxy [18] or have to index their images (and encrypt that index) locally [17], which implies the use of additional computational power on their side and limits the practicality of the solution for light and mobile devices. This is even more limiting if we consider dynamic application scenarios, where images are added, updated and deleted constantly. In such dynamic cases, SSE jobs require several rounds of communication to update image repositories and their indexes. For example, [17] uses statistics from the entire repository (inverse document frequencies), which change as the repositories are updated and require the reconstruction and re-encryption of the index that could require that clients performing such a task download and decrypt the complete content of the repository. In addition, in [17] the index values are encrypted with an encryption scheme that preserves the order whose security depends on the distribution of the simple text domain, and with multiple updates this distribution will change, also forcing the reconstruction and re-encryption of the index ; (ii) customers have to transfer additional data to the cloud (instead of simply loading images, they also have to recover and reload their encrypted index with each update of the repository). This leads to an additional use of bandwidth, which negatively affects the latency of storage operations as perceived by users;

(iii SSE uses deterministic identifiers and trapdoors to search for their functional performance, they leak the so-called search access, similarity, and pattern enhancements, [ [23] - [25] which intuitively refers to As they are revealed in chronological order: If a new questionnaire was sent before The images are returned by each query. Which image is similar to the specified search image? (In case of search / match rankings); [27] () In a long-lived system where multiple searches are performed and index entries are all accessed at certain intervals or when Face to face [28] These leaks have resulted in as many disclosures as would be expected of an entirely configurable encryption scheme, even if it had a value. The cost is much higher. However, the reader should know that the plan determines (And the plan to use SSEs that refer to leaks) can still prove secure - as long as high-level applications exploit them to control the amount of background information leaked to their opponents. The alternative to the SSE literature can be found in [13] based on locally-available homomorphic coding (PKHE) schemes such as Paillier [3] or ElGammal [14] In these methods, customers encrypt pixels by pixel with the PKHE project, which allows the cloud to process and index encoded images on their behalf. Many practical problems of SSE solutions. Unfortunately, PKHE works with more complex time and space. For example, Hsu et al. [15] offers high accuracy - CBIR al. The algorithm for encrypting the domain using the Paillier cryptosystem [13]. However, their results have a significant effect on the ciphertext expansion. (For a secure key size of at least 1024 bits, each pixel changes from a conventional 24 bit bitwise to a 2048 bit.) Encoding and decoding are slow. (As we will experiment in V-1 Evaluation) and on the ability to scale. In addition, their work has shown later that there is insecurity or can not be calculated for the cloud server. [29] Zheng et al. [16] Offers a variant of the function to overcome its limitations by replacing the ciphertexts with the ciphertexts instructions to the ciphertext table (created by mapping them all to the possible ciphertext pixel values). this Significant reduction of operating coding and reduce the expansion of the ciphertext, but it also shows the cost of computing, which is limited in practical use.

In addition to the SSE and PKHE research guidelines, there are other tasks that follow the guidelines we present in this article, although they have different purposes. The example is the work of Nourian et al. [20], which aims to provide privacy in a single-image match made by a third party. This work does not support large data stores, however, since only linear searches require a paired template to be encrypted again to compare against different images in the repository, and depending on availability. Public photo work is audio for encoding. This can be easily found by attackers using a highly available availability repository for dictionary attacks or by tracking user traffic. Another example is more theoretical work by Chase et al. [21], which offers a set of algorithms for multiple data encodings, including data types that use matrices, such as images, while enabling text. Searches are performed

over the ciphertext. However, their primary motivation is to retrieve some information about the data object that is encrypted. (Such as the color of the pixels specified in the image) while we focus on allowing indexing through an unreliable third-party encrypted image collection. The resolution of a user's search query in these large collections.

## III Available Approaches

We study a framework for storing, searching, and retrieving dynamic image data that dynamically changes. In this framework consists of two main components: the image encoding component that runs on the client device. And indexing and searching storage components. (In the encrypted domain), which is executed on the server. In this framework in a new encoding format designed specifically for images called Linear (Lucene) Encryption Scheme Content Based Image Retrieval. The Linear (Lucene) Encryption Scheme Content Based Image Retrieval allows us to design an external image capture system. Content-by-content (CBIR) support is based on color features and protects the privacy of the image owner and other users who are issuing the order. Compared to state-of-the-art, Linear (Lucene) Encryption Scheme Content Based Image Retrieval provides comparative data accuracy and higher computational efficiency than previous methods, as it is safe to calculate calculations. Indexing to the infrastructure of cloud providers and avoiding public and homomorphic encryption. Linear (Lucene) Encryption Scheme Content Based Image Retrieval also reduces the expansion of the ciphertext and makes Limit bandwidth and external space by adding a positive impact on user latency. These advantages are demonstrated in our experimental analysis in Sec. V, where the efficiency of the Linear (Lucene) Encryption Scheme Content Based Image Retrieval system is compared to the state of the art SSE [17] and PKHE [15].

Table1: Survey table

| Paper no | Name of methods | performance | Disadvantage |
|---|---|---|---|
| [13] | Public key cryptosystems based on composite degree | It investigates a novel computational problem | High computation require |
| [14] | A public key cryptosystems based on discrete logarithms | Reliable than composite degree | More calculations require |
| [15] | Image feature extraction | Secure than other extraction | Ignored in the multimedia community |
| [19] | A privacy preserving framework for Large scale CBIR | It allows more efficient operations than existing proposals | In terms of time and space complexity Less restrictive use cases |
| [20] | Semantics sensitive integrated matching | The system is fairly robust to image alterations | Time complexity is more |
| [24] | Efficient Similarity search over encrypted data | Sensitive data is more secure than other system | More storage require |

We present the system model for our proposal framework (Sec. III-A), along with the assumptions of the antagonist and the security assumptions (Sec. III-B) and, in the case of applications related to the application of our proposal. -C) For the rest of the paper, we use the following terminology:

Storage is a set of images stored in the cloud provider infrastructure. A cloud server or just a cloud is an outsourcing infrastructure that acts as a server for storing and compiling images. Users who are customers of our system may use a lightweight mobile phone. Each user accesses one or more of the storage locations, where they can find, add, and update images at any time. Storage Key is a

secret used to find, add, and update images in a repository. Each storage area has its own storage key. The image key is used to encrypt and decrypt the image in a storage location. (Each picture has its own photo key).

In this format, we discuss two examples of systems, architectures, and frameworks for using the Linear (Lucene) Encryption Scheme Content Based Image Retrieval framework. Photos were outsourced to a cloud-managed repository. Each storage is used by multiple users, both of whom can add their own images and / or searches using the search term format. Users can also request access to archived images from their creators / owners.

Our purpose is for the privacy of our users. All data sent to the cloud is encrypted. Storage created by a single user. When rebuilding a repository, the new repository key is created by that user and shared with other trusted users so that they can search in the repository and store the new image. Use an additional image key (specific) for that image. Image keys are kept confidential by the user, which means that even users can search in the repository. (Such as accessing the storage key) must ask the specific image owner to access them. Note that using per-image keys should be an option in our framework, for example, if the storage user wants to avoid additional key management tasks and is willing to sacrifice the fine-tuned access control they can use. Same image key All images or groups of images When the cloud gets an encrypted image for storage, it parses the relevant properties out. (In our framework, we use global color features [23]) and index images based on these attributes. The same operation as a query image, which is encrypted by a user with a storage key, is processed by the cloud and has features that are separate and matched to the storage index. Response to a search query is the number of k encoded images and metadata associated, including the id of each image and the id of the individual owner of the image. To completely decode and access the content of the image, in addition to the storage key, the user will need to use the image key for that image.

It should be noted that all critical interactions can be made asynchronous and out-of-band using shared key sharing with public authentication, such as Needham-Schroeder-Lowe [21] Permission and revocation of user rights can be easily achieved through sharing. (And renewal upon user revocation) of the token, only the repository between the trusted user and the frame execution request. However, we find that these discussions are orthogonal to the focus of this document, as the relevant mechanisms can easily be incorporated into our framework.

At this event, we are committed to protecting the privacy of users' images and search queries. The main enemy that we consider to be the malicious cloud administrator, which uses the cloud infrastructure and servers. [18], [23] - [26] We consider the cloud model to be faithful but questionable. [4] The cloud is seen as an enemy. disguise Expected to be able to perform correctly when asked. (Eg, fulfill the terms of the agreement), but may interfere with the user's information. We expect that malicious cloud administrators will have access to all data stored on disk or RAM on any device that is in the cloud and transmitted over the network from or to the cloud.

The stronger opponents that should be considered in this work are malicious users such as system users who deviate from expected behavior. Malicious users are open to many user applications because they may have access to multiple buckets and image keys before they are discovered and can easily be scanned for other users. At this event, we are focused on protecting our offerings and proving their security to malicious cloud operators and letting malicious users come up with an open and interesting future research approach. However, we recognize that different types of orthogonal mechanisms can be used to reduce the threat posed by malicious users, including access control techniques, retention of access to repositories, and keyword revitalization. We also predict that different opponents. (Such as malicious users and cloud administrators) can unite because they do not receive any additional benefits from the system. In addition, we have not considered the completeness or readiness of the implementation because it can be handled by different mechanisms in line with the involvement of this document.

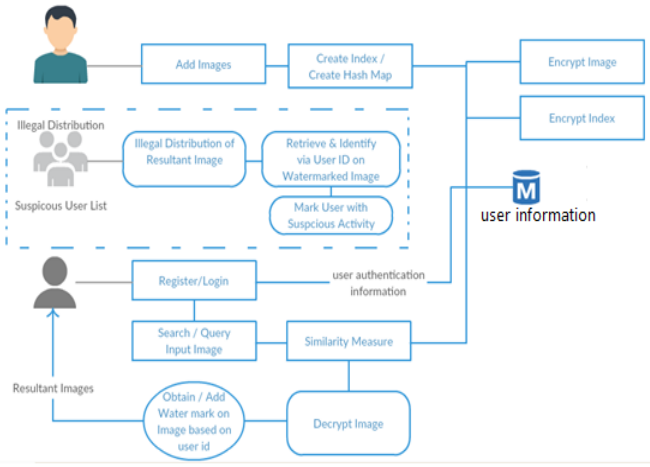


Figure 1 Basic Flow Diagram

The main element that users take advantage of is the new encryption scheme designed specifically for CBIR image and privacy protection, called Linear (Lucene) Encryption Scheme Content Based Image Retrieval. Before explaining Linear (Lucene) Encryption Scheme Content Based Image Retrieval in detail, we define the definition and underline our work. Informally, we define

the privacy of images as the ability to store the contents of public images. In general, image content is separated from color and texture. Both of these elements are easily identifiable in the image: objects, people, etc., etc., to protect the privacy of the image, thereby preventing unauthorized objects from recognizing objects in those images. We also note that color and texture data can be separated. In fact, the color information is derived from the color values of the pixels in the channels of the color model. While the texture data is determined by the pixel position (relative) and the strong color changes across the adjacent pixels. We also noted that texture information is often associated with images for object recognition. [20] It can be concluded that we cannot use only sub-components. (Such as color or texture information) to accurately summarize the content of the image, as its color information is often unclear. (Such as dark blue can translate into the sky, ocean, etc.) and surface information depends not only on But in the pixel position. But also about their color values. These conclusions are supported by recent work on image restoration, [18] which not only depends on the particular feature extracted from the subsection of the image. (In this work, we focus on global features that are separate from each other as a whole, but also on specific features that are not encrypted. Using previous definitions and observations, we designed the Linear (Lucene) Encryption Scheme Content Based Image Retrieval, a color-coded image format that separates the surface data by using different encryption techniques to protect each one, emphasizing the texture. [20] Linear (Lucene) Encryption Scheme Content Based Image Retrieval to protect image surfaces with possible encoding and color information with custom encodings. In this way, indexing and retrieval of content-based content can be done on a cloud server in a way that preserves personal information and does not interfere with the user while surface data is still protected.

Definitions of Terminologies:

- (Linear (Lucene) Encryption Scheme Content Based Image Retrieval). 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 with length polynomially bounded by sprk;
- GENIK(spik): is a probabilistic algorithm that takes as input the security parameter spik ∈ N and generates an image key ik with length polynomially bounded by spik;
- ENC(I, rk, ik): is an algorithm that takes as input an image I and the cryptographic keys {rk, ik} and returns an encrypted image CI ;
- DEC(CI , rk, ik): is an algorithm that takes as input an encrypted image CI and keys {rk, ik} and returns the decrypted image I;
- TRPGEN(Q, rk): is an algorithm that takes as input a query image Q and a repository key rk and returns a searching trapdoor CQ;

**1) Key Generation:**
Linear (Lucene) Encryption Scheme Content Based Image Retrieval works with two different cryptographic keys, the rk key and the ik key, generated by GENRK and GENIK algorithms respectively in the design and implementation of Linear (Lucene) Encryption Scheme Content Based Image Retrieval. We identified that the storage key was generated by performing three random algebraic randomization in all ranges in [0..100] random algebra (PRG) [21] G, parameterized with random seed (in use. Our work generates G with AES-based PRG [21]). The range [0..100] represents the total possible color values in the HSV color space (H). The saturation (S) of the brightness (S) (V) / brightness value), and each subkey is defined as a color field. We chose to maintain the same domain of possible values by encrypting to reduce the expansion of the ciphertext and to be able to process images as they are, including CBIR. Indexing and compression of images / Thus, the key rk = {rkH, rkS, rkV} allows the color coding to be determined by (pseudo) sampling all pixel values in the 3 HSV3 color field. In this case, sprk uses 303 × 8 = 2424 bits.

rkH, rkS, rkV ← RandP erm(G, [0..100]), rk = {rkH, rkS, rkV } ... (1)
On the other hand, an ik key is generated by requesting a 128 bit (spik) bit to G. ik is used as a cryptographic key for the Linear (Lucene) Encryption Scheme Content Based Image Retrieval encryption.

**2) Encryption:**
The Linear (Lucene) Encryption Scheme Content Based Image Retrieval encoding process is accomplished through two main steps and the final step (optional): i) pixel pixel encoding ii) pixel position change and iii) target image compression. The first is to protect the color features of the image using Pseudo-Random Permutation (PRP) [21] P in all pixel values. Although we will use standard PRP creation to create P (eg, AES PRP [21]), we choose to set the PRP of a particular color domain, which allows us to preserve the format of the encrypted image. Our constructs encode pixel values by assigning these values to each color channel using the rk storage key: {rkH, rkS, rkV}. EQ 2 implies this operation, where Prk (px) is the encoding. The pixel value p in the elements x to P and the key rkx.

CI ← Prkx (px) : ∀x ∈ (H, S, V ), ∀p ∈ I .....(2)

This encoding process securely hides the color values of the encoded pixels. However, due to the properties of P (the requirement to enable CBIR in encrypted domains), the format exists in the original image. (Which represents the texture) will still appear. To completely protect the content of the image, we need to use the possible algorithm in our encoding algorithm (pseudo) to change the position of the pixel randomly through the rows of pixels and columns that move. This procedure consists of the following: PRG G was created with the previously generated ik key (the GENIK function above) as an encrypted seed. Then for each pixel we ask that G be a new pseudorandom value r between 1 and the height of the image and make a change to the column of position r which overflows to the beginning. After moving all columns, we will repeat the procedure for the rows. EQS 3 and 4 officially describe this process, where w and h are the widths and heights of the image I. Please note that this encryption algorithm There is no expansion of the ciphertext (eg, after encoding, the width and height are the same).
This second step is probably because each new image creates a new ik created by the assumption, even if the same image is stored multiple times with different names. (If the same image key is used for all images, then iv must be input to G.) In addition, this step effectively hides the surface texture contained in the image. Computationally we cannot predict the relationship between the plaintext and the ciphertext. We choose to replace the rows and columns instead of using all single-pixel positions, assuming that they are more efficient (only the w + h pseudorandom only requires w × h).

Finally, the necessary step in the encryption algorithm is to compress the image. This is possible due to the Linear (Lucene) Encryption Scheme Content Based Image Retrieval format retention feature and can be achieved by using a non-lossless compression format like PNG directly through encrypted images. (This can also be used to compress common files such as ZIP or RAR.) This procedure allows for control over the exchange of data between the computer and the time it encrypts data with the network traffic. Requirements for cloud storage

**3) Decryption:**
The decryption algorithm uses different cryptographic sequences in reverse or more formal order using the conversion commands shown with Eqs 3,4 and 5 (after loosening the message if Required) Note that r random values must be generated in the same order as in the encoding.

$$CI((x + r) \bmod w, y) \leftarrow CI(x, y) : \forall x \in \{1, .., w\}, \forall y \in \{1, .., h\} \quad \text{.....} \quad (3)$$
$$CI(x,(y + r) \bmod h) \leftarrow CI(x, y) : \forall x \in \{1, .., w\}, \forall y \in \{1, .., h\} \quad \text{.....} \quad (4)$$
$$I \leftarrow Prkx(cpx) : \forall x \in (H, S, V), \forall cp \in CI \quad \text{.....} \quad (5)$$

**4) Searching-Trapdoor Generation:**
The TRPGEN algorithm creates trapdoors that the user can use to search the image store. Trapdoor creation requires the Q query pattern as an input, including the rk repository key. This means that users with rk access will have access. The color values of all images stored in the archive. However, the user can not access the texture information. The TRPGEN algorithm works similarly to the ENC algorithm (Equation 8, where the image key was replaced with a new random ik). This means trapdoors. The search will be decryptable and can be stored in a new image archive as long as the user queries the local image created.

CBIR in encrypted domains in the field of cloud. Encrypted images will be processed and indexed for CBIR before being stored continuously. Linear (Lucene) Encryption Scheme Content Based Image Retrieval enables these operations. (For color properties) performs through their ciphertext using an algorithm that works with non-encoded images and does not require any modifications. Encoded image processing has two main steps: And feature indexing. Feature extraction consists of image processing and retrieval of sets of vector features described. In this work we focus on color features in HSV color schemes and color rendering as histograms. For encrypted images and each HSV color channel, the cloud server generates color histograms by counting the number of pixels in each level. This data displays three color histograms, each with 101 items. When these features are fetched, the cloud can index features to perform accelerated searches. In this work, we use representations. Bag-Of-Visual-Words (BOVW) [25] to create an inverted glossary and index structure for each repository. We chose this method for indexing because it demonstrates search efficiency and scalability in the BOVW vector format. The attributes are grouped hierarchically. (Eg, using the k-means algorithm [26]) into the vocabulary structure. (Also known as codebook), where each node represents the attributes represented in the vector in the collection and leaf. The node is selected as the most representative node. This grouping process requires a training set. Therefore, in adopting our framework based on Linear (Lucene) Encryption Scheme Content Based Image Retrieval , we obtain the initial image collection from the user when creating a new archive. After creating a code, additional images are dynamically stored by hierarchical ordering. This derivation of stemming will return the visual words closest to the image, according to some distance function (in our prototype we use the distance Hamming / L1). Finally, the server in the cloud creates an inverted list index, with all the visual words as keys and, as values, the list of images.
After processing and indexing the encoded image, the cloud server can receive user search queries by sending a search trap for the selected query image. When a new search trap is obtained, the Cloud server pulls the color feature vector and searches for the closest possible keyword by stopping the code. The query's visual query is used to access the index of the repository by obtaining a consistent list of publications in the process. Then, for each image referenced in one or more of the publications, the search result is calculated for that image. (In our implementation, we use the "tlf-idf at scale" certification function. [17]) Finally, the cloud returns the best k-value to the user based on the score (k is the parameter that can be configured. The BOVW method ensures that only the most relevant images are compared. After receiving these classification results, users can request full access to the image by requesting a related image key from the owner.

**Conclusion**

In this paper, we have studied different security frameworks for storing, searching and retrieving data with a large, dynamic privacy hold. We will try to reduce customer overhead. We have also studied and analyzed the security of different approaches formally and further observed and evaluated the prototype used to show that how an approach can make the difference between precision and recall in CBIR. The study discussed in this paper is powerful and scalable when compared to alternative solutions.

**References:**

[1] Zhihua Xia, Member, IEEE, Xinhui Wang, Liangao Zhang," A Privacy-preserving and Copy-deterrence Content-based Image Retrieval Scheme in Cloud Computing",2016
[2] Global Web Index, "Instagram tops the list of social network growth," http://blog.globalwebindex.net/instagram-tops-list-of-growth, 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 on Chats," 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, pp. 91–98.
[9] National Vulnerability Database, "CVE Statistics," http://web.nvd.nist.gov/view/vuln/statistics, 2014.
[10] D. Lewis, "iCloud Data Breach: Hacking And Celebrity Photos," 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, pp. 10–18.
[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. ACM, Oct. 2013.
[17] 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.
[18] X. Yuan, X. Wang, C. Wang, A. Squicciarini, and K. Ren, "Enabling Privacy-preserving Image-centric Social Discovery," in ICDCS'14. IEEE, 2014, pp. 198–207.
[19] L. Weng, L. Amsaleg, A. Morton, and S. Marchand-maillet, "A Privacy-Preserving Framework for Large-Scale Content-Based Information Retrieval," TIFS, vol. 10, no. 1, pp. 152–167, 2015.
[20] 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.
[21] J. Katz and Y. Lindell, Introduction to Modern Cryptography. CRC PRESS, 2007.
[22] H. Muller, W. M ¨ uller, D. M. Squire, S. Marchand-Maillet, and T. Pun, "Performance evaluation in content-based image retrieval: overview ¨ and proposals," Pattern Recognit. Lett., vol. 22, no. 5, pp. 593–601, 2001.
[23] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Denitions and Efcient Constructions," in CCS'06, 2006, pp. 79–88.
[24] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient Similarity Search over Encrypted Data," in ICDE'12, 2012, pp. 1156–1167.
[25] F. Hahn and F. Kerschbaum, "Searchable Encryption with Secure and Efficient Updates," in CCS'14. ACM, 2014, pp. 310–320.
[26] R. A. Popa, F. H. Li, and N. Zeldovich, "An Ideal-Security Protocol for Order-Preserving Encoding," IEEE S&P, pp. 463–477, May 2013.