**Practical Privacy-Preserving Content-Based Retrieval in Cloud                                        Image Repositories**

**Alternative Title:** A Secure Framework for Content Based Image Retrieval in Cloud Repositories

**Aim:**

To propose a secure framework for outsourced privacy-preserving storage and retrieval in large shared image repositories.

**Synopsis:**

Cloud computing is a new form of internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is the delivery of hosted services over the internet. Cloud computing services can be public, private or hybrid. The growing industry of cloud has provide a service paradigm of storage/computation outsourcing helps to reduce users’ burden of IT infrastructure maintenance, and reduce the cost for both the enterprises and individual users. The three main benefits of cloud computing are self-service provisioning, elasticity, pay per use. The three broad categories of cloud computing are Infrastructure as a Service, Platform as a Service and Software as a Service.

**Existing System:**

Storage requirements for visual data have been increasing in recent years, following the emergence of many highly interactive multimedia services and applications for mobile devices in both personal and corporate scenarios. Existing proposals in this domain remain largely unpractical, namely those requiring fully homomorphism encryption, which is still computationally too expensive. 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, while coping with the limitations of their device’s storage capability, computational power, and battery life.

**Problem Statement:**

* In general, Encryption techniques in image processing lead to change in the size of an encrypted image. So, retrieval cannot be achieved properly.
* User’s privacy is affected due to the carelessness of cloud service provider.
* Images are leaked due to the lowest security level in the cloud.

**Proposed System:**

Our proposal is based on IES-CBIR, a novel Image Encryption Scheme that exhibits Content-Based Image Retrieval properties. The framework enables both encrypted storage and searching using Content-Based Image Retrieval queries. 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. Each repository is created by a single user. 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 in the repository and add/update images. In this work, we use the Bag-Of-Visual-Words (BOVW) representation to build a vocabulary tree and an inverted list index for each repository. We choose this approach for indexing as it shows good search performance and scalability properties. In the BOVW model, feature-vectors are hierarchically clustered 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).

**Modules:**

* **Create Repository & Upload Images**
* **Codebook & Index generation**
* **Add Image/Query to cloud**
* **Content Based Searching & Retrieval**

**Create Repository & Upload Images:**

Repository is storage space of collection of data. Each repository is created by single user. He is the owner of that repository. Then, he generates a key for that repository by using RSA algorithm and shared with the users who are all have an account to access it. Now, Repository can be accessed by multiple users with the permission of an owner. Then, owner upload huge amount of image datasets as zip file into the cloud.

**Codebook & Index generation:**

The admin of cloud has responsibility to create documents based on images which is useful for searching of images by users. So, he extracts zip file and applying CBIR Encryption technique. It encrypts images based on color values and texture features and also shuffling the pixels in column-wise as well as row-wise. Then, he creates codebook, index and image key for those encrypted images. These files are used to improve the searching efficiency of cloud and also manage the time properly while retrieving answer.

**Add Image/Query to cloud:**

Now, Users can access the cloud to add their own images into the repository. So, if that cloud has ‘n’ number of users, then repository has chance to increase rapidly. Now, the repository has collection of ‘n’ number of images in different domains. All the images are stored in encrypted format for security. Then, user has to ask query to cloud. Its take query is in the format of encrypted image using CBIR encryption technique.

**Content Based Searching & Retrieval:**

After receiving encrypted image query, the cloud extracts the features of an original image. Now applying content based searching on the codebook and image index by using that extracted features. Obviously, now searching results will be an encrypted image. This resulted answer will send to that corresponding user. Now, user can apply CBIR decryption technique to decrypt the retrieved images. So, the answer will be very fine and delicious due to huge dataset.

**Conclusion:**

Thus, a secure framework for outsourced privacy-preserving storage and retrieval in large shared image repositories has been implemented successfully.

**Software Requirements:**

* Windows 7 and above
* JDK 1.7
* Tomcat 6.0
* MySQL5.0

**Hardware Requirements:**

* Hard Disk : 500GB and Above
* RAM : 4 GB and Above
* Processor : I3 and Above

**Algorithm Used:**

* RSA(Repository Key)
* AES (Image Encryption & Decryption)

**Technology Used:**

* J2EE (Servlet & JSP)
* Java
* HTML

**Architecture Diagram:**

5. Create codebook & index

Cloud

Repository

1. Create 2.Provide 6.Add Image 7. Resulted Images

Repository key for Repository (or)

Ask query as

Encrypted Image

Owner

User1, 2, 3, n

4. Upload Images

3.Share Repository Key