A Dissertation Report
on

Privacy-Preserving Framework For Large-Scale Content-Based Information Retrieval

Submitted By:

Kartik

Enrollment No.: 16535020

Under the guidance of

Dr. Balasubramanian Raman

Associate Professor

Department of Computer Science and Engineering

Indian Institute of Technology, Roorkee

Roorkee- 247667, India

May, 2018

Declaration

I declare that the work presented in this dissertation with title **"Privacy-Preserving Content-Based Information Retrieval"** towards fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science & Engineering** submitted in the **Department of Computer Science & Engineering, Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during the period of **May 2017 to May 2018** under the supervision of **Dr. Balasubramanian Raman**, Associate Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, India. The content of this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

Date: .....
Place: ROORKEE

KARTIK
(16535020)
M.TECH (CSE)

Certificate

This is to certify that the statement made by the candidate is correct to the best of my Knowledge and belief.

Date: .....
Place: .....

Sign: .....
Dr. Balasubramanian Raman
(Associate Professor)
Indian Institute of Technology
Roorkee

Acknowledgement

Dedicated to my family and friends, for standing by me through thick and thin, without whom i would not have gotten this far. I would like to express my sincere gratitude to my advisor **Dr. Balasubramanian Raman** for the continuous support of my study and research, for his patience, motivation, enthusiasm and immense knowledge. His guidance helped me in all time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my study. I would like to express my sincere appreciation and gratitude towards **Mr. Amitesh Rajput** for their encouragement, consistent support and invaluable suggestions at the time I needed the most.

I am also grateful to the Department of Computer Science and Engineering, IIT Roorkee for providing valuable resources to aid my research.

KARTIK

Abstract

In this Internet era, multimedia content is drastically increasing over the Internet and distributed over thousands of servers. In order to find content over the large scale database, content-based search strategies(CBIR) have been implemented. Recently, issues have arisen along with CBIR- privacy of database and query information. A privacy issue arises when an untrusted party want to access the data of private information of another party. The main challenge is content based search should be performed without revealing the query content and the database information to server or any intruder because according to recent trend Google, face book, flicker, and other web applications are unknowingly collecting client interests, which are used for recommendation system, creating political polls, and many others. We are using robust hash values for preventing the query from revealing the original content and features of the original content along with client can randomly eliminate certain bit for creating then ambiguity for the server because server has to return all the possible combination of the omitted bits that will create ambiguousness for server, what is returned by database server to the client[17]. Since, both the parties, the client and the server, have only exchanged their hash values therefore privacy of client interest and database content are protected. There is a feature oftunablity, where we can adjust the privacy level. Random projection is the algorithm[18] for calculating the robust hash values. Furthermore, due to returning near duplicates will also lead to weaken the privacy, called as voting attack and due to omitting the bits of robust hash will cost more bandwidth and time. In addition, in our proposed work we are using encryption on fetch data by server prevent server's proprietary information and the server's data.

Keywords- Encryption, data privacy, image hashing, indexing.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Early attention on privacy preserving raised in biometric system[1, 3], in which the query and the database contain biometric information or identifiers. Biometric system rarely keep information clean because this information is highly vulnerable, due to fearing theft. Therefore, biometric system rely on cryptographic system to protect the database. In multimedia domain, the challenge is content based search, particularly, should be performed without revealing the query content and the database information to server or any intruder because according to recent trend Google, face book, flicker, and other web applications are unknowingly collecting client interests, which are used for recommendation system, creating political polls, and in many others applications[6].

In the recent years, the swift technological developments in the areas like social networking, internet applications, clouds computing, etc. have raised vital concerns related to the security and privacy of user-related data. With the advent and rapidly increasing popularity of social media, privacy-related incidents and harms are increasing significantly. One of the major privacy concerns exists when the outsourced image data may leak secret information of the user, like personal identity, locality, or economic profiles. Also, these extracted features from the data may reveal important private and personal information.

Thus, it has now become imperative to include privacy preserving techniques to the processing of user related data. Also, the sensitive information is prone to vulnerabilities during the communication and handling at the cloud servers. This has led to an inevitable need for implementing cryptographic techniques before the transmission and processing of data. One of the straight-forward ways of

safeguarding a digital image is to encrypt the data beforehand. Image processing in encrypted domain refers to the act of performing image processing operations on an encrypted image, so as to generate the desired results without decrypting and then performing the same operations on the actual image. It involves processing images while maintaining the security, privacy and integrity of the data. Due to this, researching privacy preserving image features on encrypted domain is of significant importance.

With the recent developments in the privacy preserving, biometric system, database content protections, scaling and cropping while preserving privacy, and in almost in every field privacy is become the first requirement. In information protection from server, only few work have been proposed. One is based one cryptographic system that requires highly computation, which is not feasible because of expensiveness, speed, and highly complicated implementations, second is, sending dummy queries to server with the thought that server will get confused between the dummy queries and the original queries, but original queries can be predicted with large probability using machine learning, third is, server can send its whole data, but this is not feasible in any perspective. As many PCBIR(privacy preserving information retrieval) have been proposed but still this is most trending research topic because all the techniques that are proposed are having issues like high implementation cost in terms of speed and time. The main common problem faced by PCBIR is, server is not trustworthy not only from the perspective of database owner,but also perspective of users. There are three categories, in which PCBIR technique can be applied i.e. when database having private information, when client's query having private information, and when CBIR technique having private information, e.g proprietary information. So far, there has been many approaches have been proposed for encrypted database. But new challenge has arisen for public database that is server should not be aware of what is happening between client and database server. In this work[17] we are dealing with both private and public database.

The work done by most researchers focus on privacy concern over private database. But the new concern for public databases is server should not be aware of what is happening between client and database[17]. On above issue, only few work have been proposed. One is based one cryptographic system that requires highly computation, which is not feasible all scenario because of expensiveness, speed, and highly complicated implementation, second is, sending dummy queries to server

with the thought that server will get confused between dummy queries and the original queries, but original queries can be predicted with large probability using machine learning, third is, server can send its whole data, but this is not feasible in any perspective. As many PCBIR(privacy preserving information retrieval) have been proposed but still this is most trending research topic because all the techniques that are proposed are having issues like high implementation cost in terms of speed and time. The main common problem faced by PCBIR is server is not trustworthy not only by the perspective of database owner,but also perspective of users.In other latest researches research gap is still exists, user's interest is still vulnerable to be analyzed by the server for the public usage like recommendation system, political poll perdition, etc. There are two approaches for PCBIR. Conventionally, PICBIR concentrates on the scenario where both the things, database and query, are private. Many solution have been proposed for the same. Some of the solution are based on SPEED *(Signal Processing in Encrypted Domain)*. The disadvantage of this approach was, this approach typically rely on highly cryptographic computations. Due to highly computations pr complicated implementations, speed is affecting at hight extent as it is very expensive on large scale. On other hand, advantage of SPEED is, it gives high protractions. Aside of SPPED approaches, other solutions are also been proposed base on SRR *Search with reduced index*. These approaches work based on secure index. Secure index gives information about query and its also called as reduced reference. The reduce information helps in providing the security of actual content and accelerate the database search. According to the privacy analysis of system[17], to have better protection from sever, the requirement of number of the omitting bits is more. Due to increasing the number of omitting bits will leads to increases the size of set A, which will cost more bandwidth, more time, and client requires more storage. In order to get protected client's interest from sever, database unnecessarily giving extra information to client that has required i.e. P_3 at risk.The proposed framework is essentially an SRR approach. The key components are robust hashing and piece-wise inverted indexing. The flexibility is mainly embodied in that any robust hash algorithm can be used as a module, and any feature can be converted to hash values. In addition, the level of privacy protection is controlled by a privacy policy. These elements work together according to a new PCBIR protocol. The performance of the framework has been evaluated by extensive experiments. We apply the framework to a concrete content identification scenario. It is tested in several cases where the database size ranges from 50 thousand to

five million. Two different robust hash algorithms are used to show the versatility of the framework. Overall, the retrieval performance matches state-of-the-art baseline algorithms. The privacy enhancement is effective and can be well tuned; it turns out to slightly improve the retrieval performance

1.1 *Contribution*

The symmetric key Encryption is added in this paper, it will help to enhance the existing model. This system works effectively for both the databases, private database and private database. In comparison of existing modal, our system is more attractive:

- It is flexible to large-scale public or private databases.
- It doesn't reveal extra information to client as the information, resulted by database but hasn't queried by user, is encrypted.
- It can provide the level of protraction on low cost.
- It prevent all information, intended and unintended, throughout the network .

The results show significant performance of the proposed model in terms of both, search efficiency and strength of encryption. Compare to existing modal, our modal gives nearly same search efficiency along-with privacy enhancements.

Chapter 2

RELATED WORK

2.1 Literature review

There are number of work have been done of privacy-preserving. Some application are based on biometric informations, face detection while preserving the privacy, fingerprint matching and ECG classification. In their papers [8, 15] has worked on hiding users interest from user and on providing to the point information to the user, but they are using public cryptography system. The disadvantage of this approach was, this approach typically rely on highly cryptographic computations. Due to highly computations pr complicated implementations, speed is affecting at hight extent as it is very expensive on large scale. On other hand, advantage of SPEED is, it gives high protractions.

Obfuscation-based private web search [2] has presented a method creating the Obfuscation to the server. The technique for Obfuscation is just simply sending of the many dummy queries along with the query, but this Obfuscation technique doesn't guarantee the good obscureness. This strategy blocks the sever by sending dummy queries and, there are many machine learning approaches which easily analyze the query correlation. Some of the solution are based on SPEED *(Signal Processing in Encrypted Domain)*. The disadvantage of this approach was, this approach typically rely on highly cryptographic computations. Due to highly computations pr complicated implementations, speed is affecting at hight extent as it is very expensive on large scale. On other hand, advantage of SPEED is, it gives high protractions. Aside of SPPED approaches, other solutions are also been proposed base on SRR *Search with reduced index*. These approaches work based

on secure index. Secure index gives information about query and its also called as reduced reference. The reduce information helps in providing the security of actual content and accelerate the database search.

A privacy-preserving framework for large-scale content-based information retrieval [17], the user creates a ambiguous query before sending to the server, after creation send to the server. The server generate all possible combination from client's ambiguous query, and fetch information for all extended queries. The server fetch information with all possible queries and send all fetched items back. User extract information based on the actual query form retrieved set. This model works well when omitting bits are less and data return by database sever is type of mix of many type of combinations, in that case it will be difficult for the sever to know the client interest but if the sever return almost similar results then *client's privacy while sending queries* won't be secured. The robust hashing[4, 17], it will map the multimedia object to the robust or compact hash values. A robust hash value is generally a string of independent bits. It is consistently and persistently enough to locate multimedia content independently, like biometric finger prints. The splendid property for robust hash value is similar hash values represents similar content or data. The benefit of robust hashing is fast search due to its compact in size and it is computationally difficult to obtain input from output. The privacy *Client's privacy while sending queries* and *server's privacy while receiving results* are preserved by robust hash value instead of actual query. The solution proposed by *Shashank et al.*[16]. This protocol can only retrieve one bit at a time. Along side user is maintaining the hierarchy information of database. Due to this server privacy is violated and all the searching burden is completely shifted to user. *Sabbu et al*[15] proposed relatively same solution but, it uses the homomorphic encryption and gives better privacy protection. In a recent survey by *Rane and Boufounos* [13], it is mentioned that there are three classes of privacypreserving nearest neighbor solutions: computational methods, information-theoretic methods, and randomized embedding methods. The first one corresponds to the SPEED approaches. The second one is only suitable for limited applications, because it requires a trusted third party for distance computation. The third one corresponds to SRR approaches. Among the SRR approaches, the difference usually lies in the generation of the index and the corresponding database structure. An insecure index (feature) can be converted to a secure one through some perturbation. *Lu et al.*[9] proposed two index perturbation methods for the Jaccard similarity between bags of visual words. One is based on random permutation and

order preserving encryption. The other is based on min-Hash sketches. In another work [10], they proposed two general feature perturbation methods based on bit-plane randomization and randomized unary encoding. Recently, *Mathon et al.*[11] proposed to use quantization indices as queries but hide quantizers reconstruction points from the server. These perturbation schemes typically causes slight degradation in retrieval performance. When the database is public, oblivious retrieval is required. It makes PCBIR different from other privacy-preserving applications, because others typically assume that all parties can know the final output. It is intrinsically difficult for multimedia databases, because if the server does not read the whole database, something about the query is guaranteed to be revealed[16]. In general, the communication complexity is linear in the database size.

In two scenario communication can more efficient, one is when servers are non-colluding for the same database, second is when many users queried to same server. These technique are unimplementable in real world and out of scope of our work. For more information refer to[7, 12]. These are the shortcomings from existing work:

- Very expensive and complex implementations e.g. homomorphic encryption;
- Many works are not suitable for very large databases, see[5];
- Problem of colluding servers;
- Retrieval performance degraded, [11];
- Uneven load between user and server;

Chapter 3

PROBLEM FORMULATION

In our scenario we focused on two parties: the client and the server and our assumption is server having much more computation power than user along with network consideration is heterogeneous.

3.1 *Privacy Definition*

There are many definition of privacy and one of most broadly suitable definition is "the right to be left alone". This definition includes, user identity should be undetectable, user retrieved content should be confidential in communication channel.

The definition of privacy for short run, it is actual query content and information related to query in database. For long run, finding the client profile. Client profile can be find out tracking the all queries of client, and later it can be sell out to distributors.Distributors use client information for recommendation system where recommendation improved by linking client interest. Server side privacy includes proprietary CBIR strategies. And, client is supposed to get information that it needs.

3.2 *Threat Model*

From client perspective, server is the potential adversary. The threat is server collects and analyze the client information. Particularly, server can be potential adversary in two ways : 1) for analyzing the query information. 2) to extract the query features. The server can learn information and features of query one is, when it receive the query, and second is when server returns the matching results.

From Server perspective, client is the potential adversary. The threat is client gets too much information that hasn't been queried by client. Therefore, client knew extra information about database or client extract proprietary strategies. Our assumption is that both the parties are curious-but-honest.

3.3 *Application Scenario*

Proceeding for further analysis, we look on two major domains i.e. public database and private database.

1) *Private Database*: In private database, the content is encrypted or content is not in actual form or content restricted for public access. In case of public query, possible application can be searching in database. In song recognition, client send the record clip of audio to get the name of the song, in this case privacy concern is about client profile and database information. In case of private query, possible application can be cloud based, where client is the owner of the data and server's work is outsourcing the service of database usability. Here, privacy concern is both query and database. Another crucial application is remote face recognition in defense and government authorities wants to search the criminal record using the face image in remote database (confidential).

2) *Public Database*: In public database, this concern is mainly intended for client. In private query,the privacy concern is about client actual query only.In case of private query, possible application can be trademark search. Another possible application in remote diagnosis where client send patient images to database for matching the best result. Here privacy concern is, server should not analyze the query(which may contain the patient crucial report status). For example, if someone invented new application and he wants to search if similar applications exists

Scenarios	Private database	Public database
Private query	Secure database(outsource), Biometric recognition (P_1, P_2, P_3)	Remote diagnosis, trademark search (P_1, P_2)
Public query	Song recognition (P_1, P_2, P_3)	Search engine (P_1, P_2)
Notations	P_1 - Client's privacy while sending queries P_2 - Client's privacy while receiving results P_3 - Server's privacy while receiving results	

TABLE 3.1: Privacy abbreviations and Applications. Some applications can be possible in multiple scenarios

without disclosing the information.In case of public query, the privacy concern is server can analyze the client profile.In case of public query, possible application is our daily Internet activity.

A good system should keep P_1, P_2, and P3 sufficiently large. Note that P_1 is a necessary condition for P_2. Increasing P_1 also increases P_2 and decreases P_3, because the size of the matching set A increases. In practice, A is upper bounded by the available bandwidth, the computing power of the client, and the size of the database; it is lower bounded by the minimum privacy requirement. Specifically, there are the following requirements on the partial query:

1. It is difficult to infer the original query;
2. It is feasible to generate and perform search with the extended query list;
3. The properties of A, e.g. the size and the diversity, can be controlled by the partial query;
4. It is easy to estimate P_1.

There are the following requirements on the matching set A:

1. A should be compact enough to save bandwidth;
2. A contains the best answers, e.g., the (approximate) nearest neighbors;
3. The diversity of elements in A is sufficiently large;
4. The server cannot tell which are the best answers by analyzing A;

5. A should not reveal too much information about the database.

A solution for client privacy is that server send its replica to to the client, but this is not feasible due to limitation client's bandwidth, client's storage, and it also violates server privacy P_3.

Due to above issues client has to do a compromise is, sending the incomplete query information by user will create ambiguity for the server. The newly privacy enhanced protocol as follows[17]:

1. The user/client creates a ambiguous query by eliminating some of bits from original query before sending to the server, after creation send to the server.
2. The server generate all possible candidate's combination from client's ambiguous query, and fetch information for all extended queries.
3. The server fetch information with all possible queries and send all fetched items back.
4. User/client extract information based on the actual query form retrieved set.

3.4 *Preliminaries*

3.4.1 Encryption

Image Encryption refers to the concept of taking the pixel bits of an image and collectively rearranging or modifying their values using a defined logic, thereby leading to a completely new set of pixels, which is different from the original combination and thus, ensuing in obscuring visual information of the image. This way of encryption is unlike the conventional encryption algorithms like ECC, AES, etc. which increase the computations while deploying over images. The advantage of using such techniques for encryption is that they disturb the original sequence of pixel positions in the image, and hence the pixel neighborhood operations like SIFT, extraction of edges, etc., are not attainable anymore. But at times, these geometric information can be detected from images which were encrypted using ECC or AES algorithms, which is not desirable.

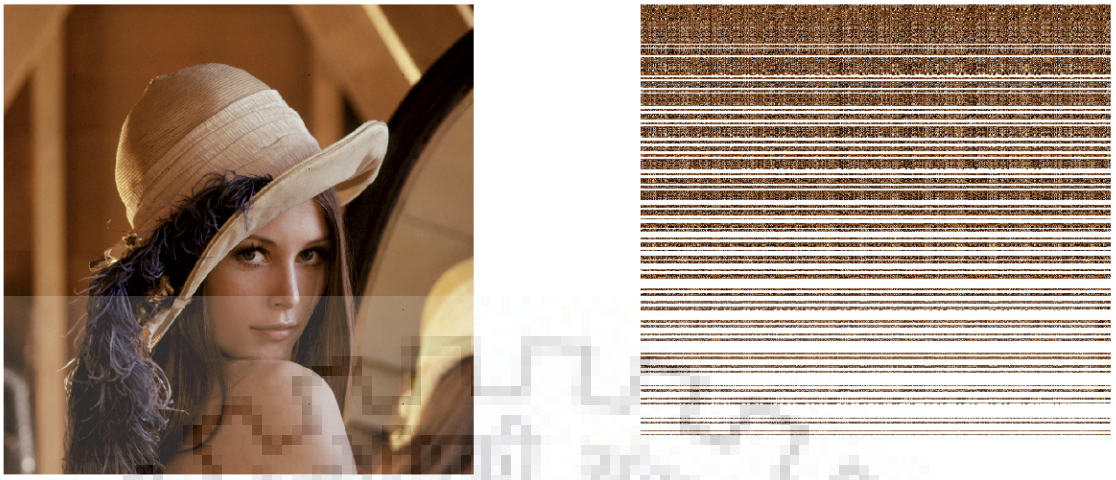

FIGURE 3.1: Actual image in database and sent image to the client

3.4.2 Query Generation

According to P_1, there is requirement for ambiguous query generation. Therefore, either the original content or features of original content should not be used for query generation because sometimes features reveals information of query content[17].In this report, we are generating queries from original content by robust hashing[4, 17]. It will map the multimedia object to the robust or compact hash values. A robust hash value is generally a string of independent bits. It is consistently and persistently enough to locate multimedia content independently, like biometric finger prints. The splendid property for robust hash value is similar hash values represents similar content or data. The benefit of robust hashing is fast search due to its compact in size and it is computationally difficult to obtain input from output. The privacy P_1 and P_3 are preserved by robust hash value instead of actual query.

3.4.3 Database Indexing

A technique called *piece-wise inverted indexing* has been used as database indexing. The *orthogonal transformation, feature extraction, and dimension reduction* are being used as preliminary steps of piece-wise inverted indexing[17]. Thereafter, we are left with significant features. All features are divided into the n groups. The robust hash value (h_i where i = 0, 1, 2, ..., n-1) will be calculated from reduced features. h_i is calculated by selecting i^{th} group. Finally, we will be having robust hash by concatenating of all the sub hash values is $H = h_0\ h_1\ .....h_{n-1}$.

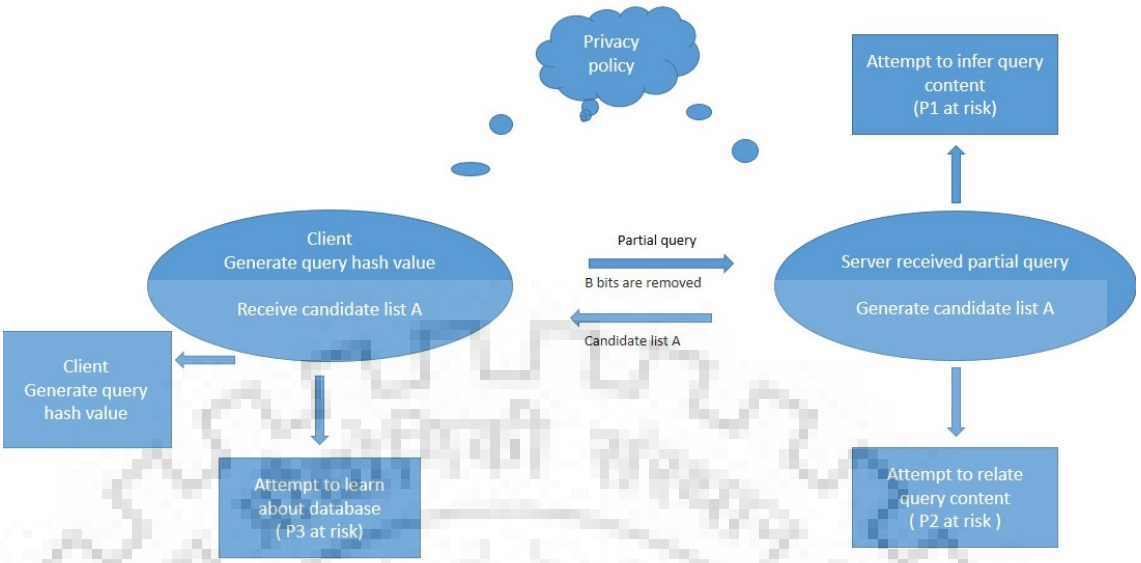

FIGURE 3.2: Depiction of existing procedure. Client send the incomplete or partial robust hash to server, server send a list of information, also including intended information and their corresponding hash value. User perform search into the retrieved list [17].

Inverted index[17], it is for identification purpose of multimedia data corresponds to the sub-hash value and the size of sub hash value x depends on features of the data i.e. significance of the corresponding sub-hash feature.

3.4.4 Database Search

Without database privacy, this solution will work as better as a normal CBIR technique. There are different methodologies for distance computations, they can be features of the data, hash value space, or any other features that defines the data uniquely. In our scenario, we are using hash query. It can be generated by either server or client doesn't matter, P_1 will be affected. But in latter case, client was using the original details for generating the input query to the sever, no privacy P_1 was guaranteed for the client[17].

Approximate Nearest Neighbor Search

On receiving the client's request, server checks the hash bucket or inverted index list for each and every sub-hash values and find nearest neighbor by hamming

distance or by feature vector within hamming sphere i.e within small radius r. All the retrieved entities hash values are collected in list A. After, list A of hash value is sorted by distance from currently query hash value and it is defined as L_1 distance[17].

$$D(H_1, H_2)|_{L1} = \sum_{i=0}^{n-1} |d_H(h_{1i}, h_{2i})| \tag{3.1}$$

Here d_H denotes two sub-hash values hamming distance. Generally we assumed that similar sub-hash values belongs to similar multimedia content. Therefore, the sorted list of hash values will results nearest matched.

Approximate Nearest Neighbor Search With Privacy

To have privacy has to be ON, the hash value should be generated by the client. Thereafter, partial query generation accomplished by omitting few bits either from same sub-hash or from multiple sub-hash, its depends on how hard privacy has to be achieved. The omitting of more bits give more privacy to client i.e (P_1, P_2)[17].

The incomplete hash value queried to the sever with the position of absent bits. If b bits are omitted from each sub-hash and size of bucket is n, 2^b * n combination of hash value has to check by sever. After, all the retrieved hash values and the data are sent to the server and client will perform search within retrieved hash-values by his own query hash value.

Chapter 4

PROPOSED APPROACH

4.1 *Approach For Voting Attack*

The performance of this attack actually depends on the nature of the database. One important factor is whether there are near-duplicates. In the case of no near-duplicates, a nonempty hash bucket only contains distinct items, and so does the candidate list. The majority voting is unlikely to succeed in this scenario. In the other case, a non-empty hash bucket may contain some near-duplicates. The candidate list is likely to have a non-uniform distribution, which might facilitate the majority voting attack.

When the server returns almost similar results for all the combination of hash values then this scenario will enable server to learn the client's interest, this issue is known as voting attack i.e P_1 at risk. Therefore, the existing system is still vulnerable to be analyzed by the sever for the public usage like recommendation system, political poll perdition, etc. According to the privacy analysis of existing system, to have better protection from sever, the requirement of number of the omitting bits is more. Due to increasing the number of omitting bits will leads to increases the size of set A, which will cost more bandwidth, more time, and client requires more storage. In order to get protected client's interest from server, database unnecessarily giving extra information to client that has required i.e. P_3 at risk.

This model works well when omitting bits are less and data return by database sever is type of mix of many type of combinations, in that case it will be difficult

for the sever to know the client interest but if the sever return almost similar results then P_1 won't be secured.

We proposed a encryption solution using symmetric key cryptography. Figure 4.1 is the architecture of the proposed approach that deals with both of vulnerabilities. After candidate list set A generated by server, all the items of candidate list are to be encrypted using symmetric key algorithm. The symmetric key of each item of candidate list will be their respective complete hash value. All the encrypted item of candidate list are sent to the client. Client will perform decryption operation on candidate list using his query original hash value. At the time of decryption client can decrypt only the data that has belongs to him, and other information will get scattered or shuffled internally because of wrong decryption key. Due to encryption, omitting of less number of bits can provide better P_1 then existing solution because there is no issues left of similar result return by the database as all the item of candidate list is encrypted and will only be decrypted by correct key only that key only client knows. Parallelly, the problem of client's bandwidth and client's storage have been reduced along with P_3 is also secured because client can only decrypt only data that has belongs to him.

4.2 *Architecture of proposed System*

FIGURE 4.1: Architecture of proposed System.

The steps of proposed System as follows:

1. The user/client creates a ambiguous query before sending to the server, after creation send to the server.
2. The server generate all possible combination from client's ambiguous query, and fetch information for all extended queries.
3. The server encrypts all possible results return by extended query by their respective complete hash value.
4. The server send all encrypted fetched items back.
5. User/client decrypt information using the actual hash value.
6. The correctly decrypted data belongs to User/client.

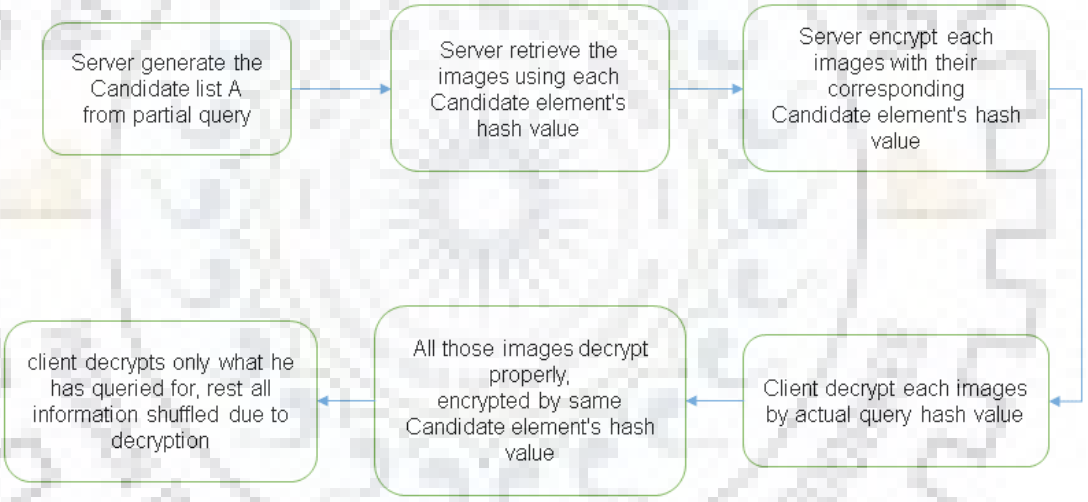

FIGURE 4.2: Algorithm flow of the proposed System

4.3 *Algorithm of proposed System*

1. Client extract features of the query image and generate 128 bit hash query using LSH (locality sensitive hashing).

   Actual query=b_1 b_2 b_3 b_4 b_5 b_6.....b_{127} b_{128}.

2. Client omit some bits from actual query to create a ambiguous query before sending to the server, after creation send to the server.

   Ambiguous query = b_1 * b_3.....b_{13} *..... b_{22} *.....b_{53} *.....b_{75} *.....b_{128}.

3. The server generate all possible combination from client's ambiguous query.

   Extended queries = b_1 (0 or 1) b_3.....b_{13} (0 or 1)..... b_{22} (0 or 1) .....b_{53} (0 or 1) .....b_{75} (0 or 1) .....b_{128}.

4. Server fetch information for all extended queries.

5. The server encrypts all possible result return by extended query by their respective key of complete hash value.

   Encryption Key Generation:- i) Approach is to convert each 128 bit extended hash into 40 bit hash.

   ii) Select all the changing bits (b_1 to b_{10}, b_{13} to b_{30}, b_{52} to b_{62}, b_{72} to b_{82}) and combine in the order of most significant bit first.

   iii) Convert 40 bit hash into decimal value.

   iv) EncryptionKey = String combination 7 random digits of the decimal value in the order of most significant digit first.

   v) Encryption steps as follows:

```
1: procedure OriginalImage(length, width) ▷ Original image with its dimensions
2:     RedColor ← OriginalImage(:, :, 1)
3:     GreenColor ← OriginalImage(:, :, 2)
4:     BlueColor ← OriginalImage(:, :, 3)
5:     l, w ← 1
6:     while l ≠ length do
7:         while w ≠ width do
8:             p ← (l * EncryptionKey) mod lenght
9:             q ← (w * EncryptionKey) mod width
10:            temp ← RedColor(l, w)
11:            RedColor(l, w) ← (RedColor(p, q) + p) mod 255
12:            RedColor(p, q) ← temp
13:            temp ← GreenColor(l, w)
14:            GreenColor(p, q) ← (GreenColor(l, w) + q) mod 255
15:            GreenColor(p, q) ← temp
16:            temp ← BlueColor(l, w)
17:            BlueColor(l, w) ← (BlueColor(p, q) + p + q) mod 255
18:            BlueColor(p, q) ← temp
19:            width ← width + 1
```

20: **end while**
21: $length \leftarrow length + 1$
22: **end while**
23: **return** $rgbImage$ ▷ The rgbImage is combination of all the color lists
24: **end procedure**

6. The server send all encrypted fetched items back.

7. User/client decrypt information using the actual hash value.

Decryption Key Generation:- i) Approach is to convert 128 bit actual hash into 40 bit hash.

ii) Select all the changing bits (b_1 to b_{10}, b_{13} to b_{30}, b_{52} to b_{62}, b_{72} to b_{82}) and combine in the order of most significant bit first and positions of all seven random bit should be same as position of encryption key.

iii) Convert 40 bit hash into decimal value.

iv) DecryptionKey = String combination of seven random digits of the decimal value in the order of most significant digit first and positions of all seven random digit should be same as position of encryption key.

v) Decryption steps as follows:

1: **procedure** RGBIMAGE($length, width$) ▷ Original image with its dimensions
2: $RedColor \leftarrow rgbImage(:, :, 1)$
3: $GreenColor \leftarrow rgbImage(:, :, 2)$
4: $BlueColor \leftarrow rgbImage(:, :, 3)$
5: **while** $lenth \neq 1$ **do**
6: **while** $width \neq 1$ **do**
7: $p \leftarrow (length * DecryptionKey)$ mod $lenght$
8: $q \leftarrow (width * DecryptionKey)$ mod $width$
9: **if** $(RedColor(length, width) - p) < 0$ **then**
10: $temp \leftarrow 255 + RedColor(length, width) - p$
11: **else**
12: $temp \leftarrow RedColor(length, width) - p$
13: **end if**
14: $RedColor(length, width) \leftarrow (RedColor(p, q))$
15: $RedColor(p, q) \leftarrow temp$

16: **if** $(GreenColor(length, width) - q) < 0$ **then**
17: $temp \leftarrow 255 + GreenColor(length, width) - q$
18: **else**
19: $temp \leftarrow GreenColor(length, width) - q$
20: **end if**
21: $GreenColor(length, width) \leftarrow (GreenColor(p, q)$
22: $GreenColor(p, q) \leftarrow temp$
23: **if** $(BlueColor(length, width) - p - q) < 0$ **then**
24: $temp \leftarrow 255 + BlueColor(length, width) - q$
25: **else**
26: $temp \leftarrow BlueColor(length, width) - p - q$
27: **end if**
28: $BlueColor(length, width) \leftarrow (BlueColor(length, width))$
29: $BlueColor(p, q) \leftarrow temp$
30: $width \leftarrow width - 1$
31: **end while**
32: $length \leftarrow length - 1$
33: **end while**
34: **return** $rgbImage$ ▷ The rgbImage is combination of all red, green and blue color lists
35: **end procedure**

8. The correctly decrypted data belongs to User/client and other extra information gets shuffled more.

Chapter 5

Experimentation and Results

5.1 *Data-set*

For validation of proposed framework the experiment have been performed. In order to have solid ground truths, we make a concrete example by applying the framework to a nearduplicate detection scenario, i.e., finding similar copies of the same content in a database. In the following, we first describe the databases for experiments. Then we demonstrate the effectiveness of the indexing and retrieval schemes without considering privacy. Afterwards, we turn on privacy protection and consider both the privacy-preserving performance and the impact on retrieval.

ILSVRC'2012[14]

We have used a database of collection of 50,000 images from ImageNet(ILSVRC'2012). This database is type of public domain collection. Database consists of 1000 categories and each category contain image-set of 50 images. We are representing each image by 128 bit robust-hash value. We are separating the database into three part and along with the existence of near-duplicates. Each case of database containing the query set which are used for searching and the data set within search will be performed.

Performance evaluation is done by simply mean average of all the queries (50 queries based on californiaND data set, 100 queries based on ILSVRC'2012 data set, and 5 queries based on manual data set) result, individually for each dataset.

5.2 *Results*

Retrieval Performance

The table 5.1 shows the accuracy of all the approaches individually. We have maintained the retrieval performance same as the existing modal. Moreover we enhanced the privacy of the existing system.

Method	Accuracy (%)
LSH	63
LSH MP1	81
LSH MP2	89
DWT	70
DWT MP1	87
DWT MP2	95

TABLE 5.1: Retrieval performance.MP x means multi-probing within Hamming radius x. DWT generally performs the best, followed by LSH. The best recall is given by MP2

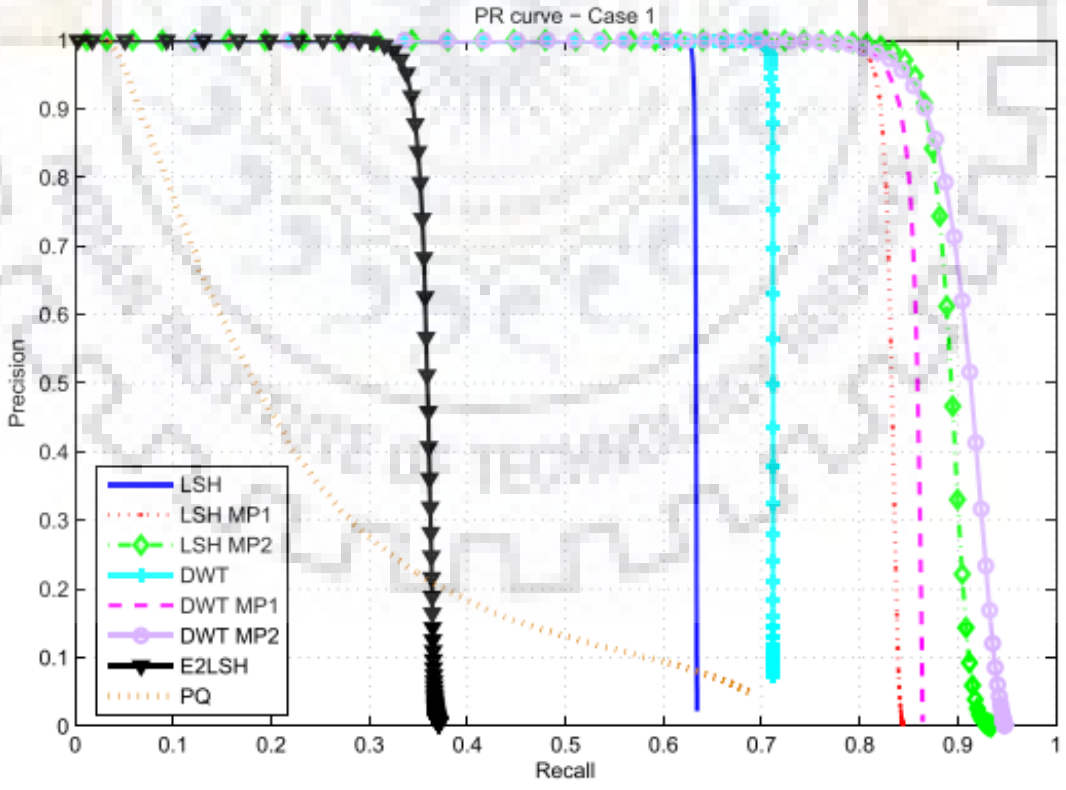

FIGURE 5.1: Retrieval performance.MP x means multi-probing within Hamming radius x. DWT generally performs the best, followed by LSH. The best recall is given by MP2

Privacy-Preserving Performance

We consider the privacy-preserving performance in terms of P2, because P3 is inversely proportional to P2, and P1 is decided by the system parameters. Privacy-enhanced retrieval is carried out according to different privacy policies. First, we randomly omit b = 1, , 8 bits from each sub-hash value. We refer to these policies as the baseline policies. Since the server puts all candidates into a list, two metrics are used: 1) The number of candidates in the list; 2) The entropy of the candidate categories in the list.

1. *Influence on Retrieval:* What is the influence of privacy enhancement on retrieval performance? Since a privacy policy is essentially a particular multi-probing strategy, one can imagine that privacy enhancement actually forces the server to behave like multi-probing. Therefore, privacy enhancement should improve retrieval performance. Indeed, the retrieval performance increases with the level of privacy protection and approaches the performance of multi-probing.

2. *Majority Voting Attack:* The majority voting attack has been applied to estimate the querys category and ID. Specifically, the most frequent category or ID in the candidate list is considered as the one of the query. First, majority voting indeed works to some extent when there are near-duplicates. That means the majority of a candidate list is likely to be the near-duplicates of the query. In our proposed model, we shuffled the data in such a way that, it is vary difficult know by server that whats the actual query the user has queried. Second, note that the success rate decreases when the number of omitted bits increases. Therefore, in order to prevent majority voting, the number of omitted bits should not be too small. Moreover, in our proposed model omitting of few bits can make server in trouble for knowing the actual content.

3. *Server Privacy (P3):* We assume that the clients interest is to know what is in the database. In existing modal, to know the information of database proprietary was easy due to omitting the bits from actual query, sever returns extra information. Therefore, in our proposed encryption modal, it is computationally difficult for the client to guess the database content from received encrypted set.

FIGURE 5.2: This figure of result shows sample encryption and decryption of our proposed approach

FIGURE 5.3: This figure of result shows sample encryption along with their histogram of all the colors (Red, Blue, Green) and both the image (encrypted, decrypted).And, Histograms depicts that color properties of encrypted image has been vanished.

FIGURE 5.4: This figure of result showing that all results return by sever are encrypted by single encryption key and all the results are decrypted by using same key as encryption key. Here, in whole process only one single of original hash is participating. Therefore, the threat can be possible.

FIGURE 5.5: This figure of result showing that all results return by sever are encrypted by their corresponding encryption key.

FIGURE 5.6: This figure of result showing that only those results are successfully decrypted for which user has queried and extra information return by sever have been shuffled more.

Conclusion

In this work, we propose a privacy-enhancing framework for large-scale content-based information retrieval. It can be used for any CBIR system based on features and similarity search. The framework is mainly based on robust hashing and piece-wise inverted indexing. Our work addresses the issue of voting attack, the privacy of server proprietary. We proposed a encryption solution using symmetric key Encryption. After the candidate-list set A generated by server, all the items of candidate list are to be encrypted using symmetric key algorithm. The symmetric key of each item of candidate list will be their respective complete hash value. All the encrypted item of candidate list are sent to the client. Client will perform decryption operation on candidate list using his original query hash value. At the time of decryption client can decrypt only the data that has belongs to him, and other information will get scattered or shuffled more, because of wrong decryption key. Due to encryption, omitting of less number of bits can provide better P_1 then existing solution because there is no issues left of similar result return by the database as all the item of candidate list is encrypted and will only be decrypted by correct key only that key only client knows. Parallelly, the problem of client's bandwidth and client's storage have been reduced along with P_3 is also secured because client can only decrypt that only data that has belongs to him.. Finally, with the help of symmetric key encryption, we resolve the issue voting attack, the privacy of server proprietary and the privacy of server's information up-to good extent.

References

[1] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *ACM Sigmod Record*, volume 29, pages 439–450. ACM, 2000.

[2] E. Balsa, C. Troncoso, and C. Diaz. Ob-pws: Obfuscation-based private web search. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 491–505. IEEE, 2012.

[3] J. Bringer, H. Chabanne, and A. Patey. Privacy-preserving biometric identification using secure multiparty computation: An overview and recent trends. *IEEE Signal Processing Magazine*, 30(2):42–52, 2013.

[4] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM, 2004.

[5] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft. Privacy-preserving face recognition. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 235–253. Springer, 2009.

[6] G. Fanti, M. Finiasz, and K. Ramchandran. One-way private media search on public databases: The role of signal processing. *IEEE Signal Processing Magazine*, 30(2):53–61, 2013.

[7] W. Gasarch. A survey on private information retrieval. *The Bulletin of the EATCS*, 82(72-107):1, 2004.

[8] R. L. Lagendijk, Z. Erkin, and M. Barni. Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation. *IEEE Signal Processing Magazine*, 30(1):82–105, 2013.

[9] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu. Enabling search over encrypted multimedia databases. In *Media Forensics and Security*, volume 7254, page 725418. International Society for Optics and Photonics, 2009.

[10] W. Lu, A. L. Varna, A. Swaminathan, and M. Wu. Secure image retrieval through feature protection. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1533–1536. IEEE, 2009.

[11] B. Mathon, T. Furon, L. Amsaleg, and J. Bringer. Secure and efficient approximate nearest neighbors search. In *Proceedings of the first ACM workshop on Information hiding and multimedia security*, pages 175–180. ACM, 2013.

[12] R. Ostrovsky and W. E. Skeith. A survey of single-database private information retrieval: Techniques and applications. In *International Workshop on Public Key Cryptography*, pages 393–411. Springer, 2007.

[13] S. Rane and P. T. Boufounos. Privacy-preserving nearest neighbor methods: Comparing signals without revealing them. *IEEE Signal Processing Magazine*, 30(2):18–28, 2013.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2012.

[15] P. R. Sabbu, U. Ganugula, S. Kannan, and B. Bezawada. An oblivious image retrieval protocol. In *Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on*, pages 349–354. IEEE, 2011.

[16] J. Shashank, P. Kowshik, K. Srinathan, and C. Jawahar. Private content based image retrieval. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[17] L. Weng, L. Amsaleg, A. Morton, and S. Marchand-Maillet. A privacy-preserving framework for large-scale content-based information retrieval. *IEEE Transactions on Information Forensics and Security*, 10(1):152–167, 2015.

[18] W. Zhang, K. Gao, Y.-d. Zhang, and J.-t. Li. Data-oriented locality sensitive hashing. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1131–1134. ACM, 2010.

Privacy-Preserving Framework For Large-Scale Content-Based Information Retrieval

ORIGINALITY REPORT

6%	2%	6%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

Weng, Li, Laurent Amsaleg, April Morton, and Stephane Marchand-Maillet. "A Privacy-Preserving Framework for Large-Scale Content-Based Information Retrieval", IEEE Transactions on Information Forensics and Security, 2015. Publication	4%
Submitted to Savitribai Phule Pune University Student Paper	1%
Li Weng, Laurent Amsaleg, Teddy Furon. "Privacy-Preserving Outsourced Media Search", IEEE Transactions on Knowledge and Data Engineering, 2016 Publication	<1%
hal.inria.fr Internet Source	<1%

Exclude quotes	On	Exclude matches	< 20 words
Exclude bibliography	On		