International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Efficient Similarity Search over Encrypted Data on Cloud

Ankur Verma†*, Bhagyashri Patki†, Priyanka Sawant†, Prajkta Waingankar† and Dike Onkar D†

†Department of Information Technology, Mumbai University, Rajendra Mane College of Engineering and Technology Ambav, Devrukh, India

Accepted 01 April 2015, Available online 06 April 2015, **Vol.5, No.2 (April 2015)**

Abstract

In today's world large amount of data can be stored on cloud. Therefore security over the cloud becomes a major issue. However services on cloud may have many advantages but security of the sensitive data is a major problem. To reduce this problem, it is preferable to store data in encrypted form. Encrypted data protects the data against illegal access. To retrieve search over encrypted data, many searchable encryption schemes have been used.
In this paper, we ensure the confidentiality of sensitive data by providing security. To do so we provide homomorphic security algorithm is use to protect data from illegal access. In this cloud computing takes encrypted data and performs all the processes to meets the user query without aware of its data, and retrieved encrypted data can be decrypted only by the authorized user who acquaint the request. This helps to client to depend on the services provided without concerning their privacy.

Keywords: *The* Cloud Computing, Security, Homomorphic Encryption, Locality Sensitive Hashing.

1. Introduction

In today's data environment cloud computing removes the burden of storing and retrieving of huge amount of data in less cost effective manner. Hence personal information is outsourced into cloud. At same time transferring of fragile data to the un-trusted cloud server leads to concern about its privacy. Also encryption provides protection from illegal access. Cloud should provide efficient search on encrypted data to ensure the benefits of cloud computing environment. In fact sizeable amount of algorithm have been proposed to support the task which are called searchable encryption scheme almost all such schemes are designed for exact query matching. They allow retrieval of exact data from cloud according to their existence. In today's world retrieval according to the similarity is specified features of the existence of it.

A similarity search include collection of huge data items according to their features, a query that specifics the value of the particular feature and measures the applicability between the query and the data items. Although exact query matching based searchable encryption methods not give correct result due cryptographic techniques used for similarity search over encrypted data. In this paper, we propose a secure encryption scheme to meets the requirements.

The basic building block of secure index is the state-of-art approximate near neighbor search algorithm in high dimensional space called as locality search algorithm. LSH is used for fast similarity search on plain data. Searchable encryption can be achieved generality using the work of Ostrovsky and Goldreich on software protection based on oblivious RAMs. In addition, we provide application of our schemes and verify the result with empirical analysis.

Homomorphic encryption: Homomorphic encryption system are used to perform operation on encrypted data without knowing the private key, the owner is the only have private key. In this we focus on homomorphic encryption on cloud to provide security, particularly to execute the calculations of confidential data encrypted without decrypting data. Homomorphic encryption enhances the security over un-trusted system or application that stores sensitive data.

Data Encryption Standard: DES is a block cipher block of 64bits. It produces 64 bit cipher text but its key length is 56 bits.

Fig.1 DES

The remaining bit position 8, 16, 24, 32, 40, 48, 56, 64 are discarded from the key length.

*Corresponding author: **Ankur Verma**

DES has two attributes of cryptography –Substitution and transposition. It further divides into symmetric and asymmetric encryption. Symmetric encryption uses the same key for encryption and decryption. Asymmetric encryption uses the same key for encryption but different key for decryption.

Locality Sensitive Hashing: LSH is widely use for fast similarity search over plan data. In our scheme, we utilize it in the context of encrypted data. To utilize the properties of the encrypted data, a secure LSH index and a searchable symmetric encryption schemes are used.

2. Existing System

In today's data environment cloud computing removes the burden of storing and retrieving of huge amount of data in less cost effective manner. Therefore huge amount of data can be stored on cloud. Storing huge sensitive data on cloud leads to concern with the security about the privacy of data. During transfer of sensitive data to un-trusted cloud severs lead to concern about it privacy. Although cryptographic techniques provide protection by complicates the computation on the data. It does not provide more security.

3. Proposed system

In proposed system, sizable amount of algorithms have been proposed to support the activity which is called as searchable encryption schemes. All such secure schemes are designed for exact query matching. This will helps to retrieve any selective data from the specified file name. so retrieval operation becomes more natural according to similarity of specified file name instead of existing one.

In this system, outsource the essential data into encrypted form so that so form of encryption will protect data against illegal access. Secure encryption is used to perform similarity search operation on the encrypted data. To perform similarity search, locality sensitive hashing algorithm is used which helps to retrieve similar data.

In proposed system, we provide strong security mechanism homomorphic encryption and prove security of proposed system under the provided mechanism and we eliminate problems which occur in existing system.

4. Algorithm Used

Data Encryption Standard (DES)

Originally designed by researchers at IBM in the early 1970s, DES was adopted by the U.S. government as an official Federal Information Processing Standard (FIPS) in 1977 for the encryption of commercial and sensitive yet unclassified government computer data.

DES is a symmetric key algorithm used for encryption of electronic data. DES used same private key for encryption and decryption. DES is block cipher. DES works on encrypt data in block size of 64 bit each. The basic principle of DES is input 64 bit plain text into DES which produces 64 bits of cipher text. The key length is originally 64 bit key in that 56 bit key and 8 bit is used for parity check. DES is achieved through diffusion, confusion.

Fig. 2 Encryption and Decryption with DES

Steps in DES

1 Divide the text into 64-bit (8 octet) blocks.
2 Initial permutation of block.
3 Break the 64 bit block into two parts: Left and Right name as L and R.
4 Repeated permutation and substitution steps 16 times called rounds.
5 Re-joining the left and right parts then final permutation are performed.

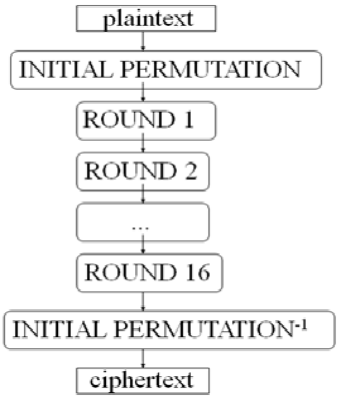


Fig. 3 General Structure of DES

Block cipher: 64 bits at a time
Initial permutation rearranges 64 bits (no cryptographic effect)
Encoding is in 16 rounds

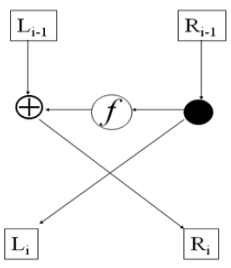
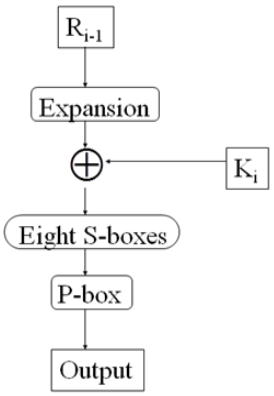


Fig. 4 Example of DES

64 bits divided into left, right halves
Right half goes through function f, mixed with key
Right half added to left half
Halves swapped (except in last round)



Expand right side from 32 to 48 bits (some get reused)
Add 48 bits of key (chosen by schedule)
S-boxes: each set of 6 bits reduced to 4
P-box permutes 32 bits

Advantages

1 Secure: hard to attack
2. easy to implement in both hardware and software.
3 It is easy to analyze.

Disadvantages

DES is considered to be insecure in much application but Triple DES is considered to be secure.
Strength of DES: The Strength of DES is 56-bit keys have 2^{56} = 7.2 x 10^{16} values.

Attacks on DES

Six ways to break DES:
1. Exhaustive key search: In that we break DES with complexity 2^{56} .
2. A dedicated machine: It can break DES faster at the expense of more memory.
3. A huge cluster of computers:
4. A time-memory tradeoff:
5. Differential cryptanalysis: It can break DES with 2^{47} chosen plaintext (full 16-round)
6. Linear cryptanalysis: can break DES with 2^{43} chosen plaintext (full 16-round)

Modes of DES

In DES specifies the four modes of operation: ECB, CBC, OFB and CFB.

Locality sensitive Hashing (LSH)

LSH is reducing dimensionality of High dimensional data. The high dimensional data means large amount of data.LSH is widely used for nearest neighbor search.

Example of LSH Algorithm

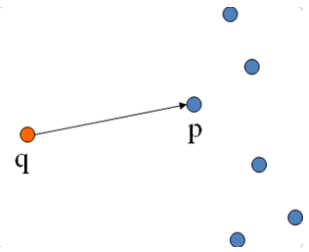


Fig. 5 Example of LSH

In this example set P of n points in R^d .and q is query point. We find the nearest neighbor of p and q.
Using hashing we find the nearest neighbor are as follows:

Hash functions are locality sensitive, if for a random hash function h, for any pair points p, q. we have:

Pr[h(p)=h(q)] is high if p is “close” to q
Pr[h(p)=h(q)] is low if p is “far” to q

Variants of nearest neighbor

1. Near neighbor (Range search): Find one or all points in P within distance r from q.
2) Spatial Join: Given two sets P and Q. find all pairs of p in P and q in Q, such that p is within distance from q.
3) Approximate Near Neighbor: Find one or all points p in P, whose distance to q is at most(1+p) times the distance from q to its nearest neighbor.

Application: Hierarchical clustering

In data mining hierarchical Clustering is also known as HCA (hierarchical cluster Analysis).It is method of cluster analysis which seeks build hierarchy of cluster. Cluster is nothing but set or group of similar object.

Hierarchical clustering generally falls into two types:
Agglomerative: This is “bottom up” approach method where clusters have sub-clusters.
Divisive: This is a "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

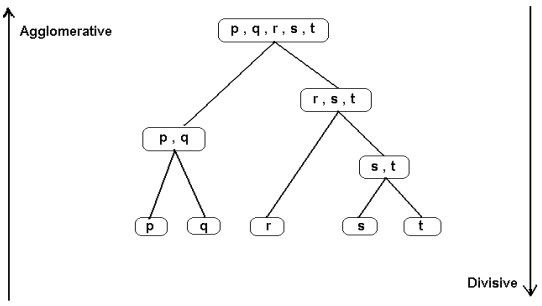


Fig. 6 Hierarchical clustering

2) Image similarity Identification

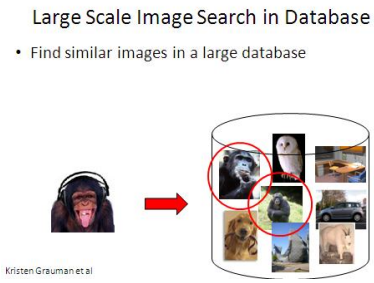


Fig. 7 Image similarity Identification

3) Digital figure printing

Video or digital figure printing is a technique in which software identifiers, extracts and then compresses characteristics of video, enabling that video to be uniquely identified by resultant "fingerprint".
Digital fingerprinting technology that has proven itself to be effective at identifying and comparing digital video data.

4) Audio Fingerprinting

Audio or acoustic fingerprinting is a condensed digital summary, deterministically generated from an audio signal, that can used for identify audio sample or kindly locate similar items in an audio database.

Practically use of audio fingerprinting includes identifying songs, tunes or advertisement and sound effect library management.

Media identification using video fingerprints can used to monitor the use of specific musical work performance on radio broadcast, records, CD's and peer to peer networks. This identification used for copyright compliance, licensing.

Secure Search Scheme

In this we describe the basic protocol for similarity search

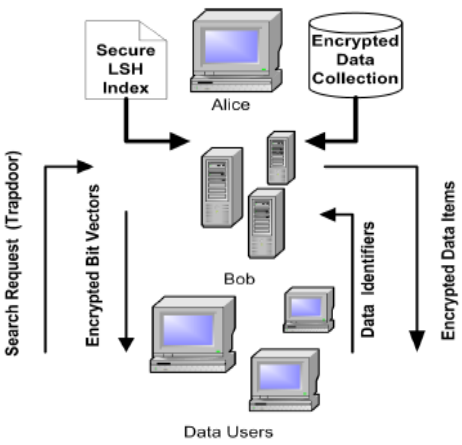


Fig. 8 Basic Secure Search Protocol

1. Key generation: In key generation, we generate key from authorized file security. The data owner who know what kind of file it is for make a key it becomes encrypt with file name for the purpose of security.

2. Index construction: The key of index can be generated by the word index which is given by data owner and file security. The search scheme can be used to perform fast similarity search. But in such situation there is problem not to sacrifice confidentiality of the essential data while providing the functionality schema.

3. Data Encryption: In data encryption, lent sends the encrypted data with secure index to server. Once the data is outsourced, then users are able to retrieve the selected data from server. To perform this, following information client shares with users.
Secrete key of data collection encryption
Secrete key of index construction
Metric key translation function of index construction
Locality sensitive hash functions of index constructions shared information.

4. Data Decryption: Once encrypted data are retrieved from corresponding search request, then user decrypts them with plain data. Client generates trapdoor for the bucket and requests encrypted bit vectors from the server. Once the bit vectors are updated and encrypted by client, they can send back to server. If some buckets may not exit on the server, then client can ask the server to replace fake records with real records. For this, lent should keep identifiers for fake records and generate trap doors for replacement. So after updating if contents of the some buckets become empty, then client asks for the replacement of corresponding records with the fake records using replacement request.

5. Search: In the search, we specify the similarity search schemes to clarify its mechanism. In this, server does operations on the index and sends bit vectors to the client. In this way, search scheme shows the proposed system analysis on real data.

Homomorphic encryption

Homomorphic encryption is a form of encryption where computations are carried out on cipher text which generates encrypted result, when we decrypt the result then we get original plaintext.

In homomorphic algorithm we encrypt data before we send to client. In this encryption we provide private key to the server to decrypt data before execute calculation required.

Homomorphic encryption encrypts data, but it does not know the private key. In this client is only one holder. When we decrypt the result it is same as that of the calculations carried on the cipher text.

It performs chaining of different services without exposing the data.

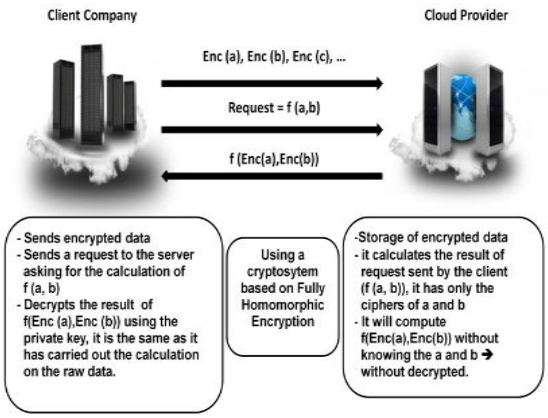


Fig. 9 Homomorphic Encryption applied to the Cloud Computing

In above figure there are two servers, client company and client provider.
1. Client company sends encrypted data to cloud provider.
2. Then client provider stores that encrypted data.
3. Client sends request to server for the calculation of function (a, b).
4. Server calculates the result whose request sent by client function (a, b).in this we see only the ciphers of a and b.
5. Hence client provider calculates the function of encryption of a and b without knowing the values of a and b without decrypted.
6. Then client provider decrypt result using private key. so we will find that it displays same calculations which are carried on plaintext.

Implementation of homomorphic algorithm

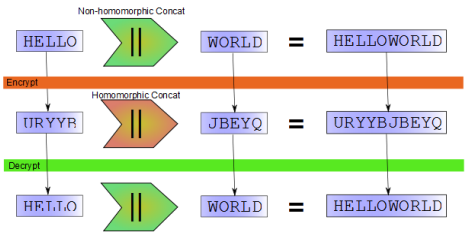


Fig.10 A string concat example of Homomorphic Encryption

In the above example, there are two variables hello and world. Then we encrypt these two variables. After doing encryption we concat the string. Then we decrypt the concated two variables using homomorphic algorithm we get our original string.
Homomorphic encryption schemes

Homomorphic encryption has numerous applications such as searching on the encrypted data, also it improves the security of efficient computation, making encrypted queries on search engine.

Definitions of homomorphic algorithm

1. Key generation: It generates the random public/secretes key pair.
2. Encryption: Encrypt a message with public key.
3. Decryption: Decrypt a cipher text with the secrete key.

Conclusion

We proposed LSH based secure index and search scheme to enable fast similarity search over encrypted data.

We provided a rigorous security definition and proved the security of the scheme to ensure confidentiality of the sensitive data.

References

M. Bellare, A. Boldyreva, and A. ONeill (2007), Deterministicand efficiently searchable encryption, in *Proc. of Crypto'07*, pp. 535–552.

R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky (2006), Searchable Symmetric encryption: Improved definitions and efficient constructions, in *Cryptology ePrint Archive, Report 2006/210*

J. Feigenbaum, Y. Ishai, K. Nissim, M. Strauss, and R. Wright (2006), Secure multiparty computation of approximations, *ACM Transactions on Algorithms*, vol. 2, pp. 435–472.

M. Atallah, F. Kerschbaum, and W. Du, Secure and private sequence comparisons, in *Proc. of the WPES'03*, 2003, pp. 39–44.

P. Indyk and R. Motwani (1998), Approximate nearest neighbors: towards removing the curse dimensionality, in *30th STOC, ,* pp. 604–613.

Maha TEBAA, Saïd EL HAJJI, Abdellatif EL GHAZ (2012) Homomorphic Encryption Applied to the Clo Computing Security, *Proceedings of the World Congress on Engineering Vol I WCE 2012*, Jul 4 - 6, 2012, London, U.K.

Sombir Singh Sunil K. Maakar Dr.Sudesh Kumar (June 2013) Enhancing the Security of DES Algorithm Us Transposition Cryptography Technique Volume 3, Issue 6, ISSN: 2277 128X

Shaymaa Mohammed Jawad Kadhim Manjusha Joshi, Dr.Shashank Joshi (October 2013), Provide the Security to a Web Service by using DES Cryptography Algorit Volume 3, Issue 10, ISSN: 2277 128X