Order Preserving Encryption-Scheme to Preserve the Order of Encrypted Cloud Data Search

Punam S. Kharche[1] Roshan Thakur[2] Dinesh Gawande[3]
[1]M.Tech Student [2,3]Assistant Professor
[1,2,3]Department of Computer Science & Engineering
[1,2,3]DBACER/RTMNU, India

Abstract— The data on cloud computing is encrypted due to security concern or the factor of third party to excavate into it. As the consequence to this, the search over encrypted data becomes a complex task. The traditional approaches like to search in plain text cannot be apply over encrypted data. So the searchable encryption techniques are being used. In searchable encryption techniques the order of relevance must be consider as the concern because when it is large amount of data it becomes complex as relevant documents are more in number. We have discussed the probabilistic OPE technique which known as one-to-many OPE. The expected result is to be that cloud server cannot penetrate in actual user data and provide the search on encrypted data will be performed and results will appear in order of relevance score. Even though with good security of one-to-many OPE the cloud can get the information of the plain text if differential attacks occurred on the cipher text by calculating the differences between the cipher text.

Key words: Cloud Computing, Document Relevance, Privacy, Searchable Encryption

I. INTRODUCTION

Now a day almost everything is moving to Cloud Computing. Cloud has been the most scalable and cost effective way to our data No extra work required to store the data on Cloud as almost everything on Cloud is automatic. But the real concern of Cloud is that Cloud is managed by the Cloud service providers and everyone is now thing of store the data on the Cloud making Cloud too much available for public. Even if a person is having data securely in the laptop a person prefers to take the back up of data on the Cloud for the reason that if data is lost or get crash in his own laptop or system. Another reason for preferring Cloud is that data is increasing so fast that there is problem of storage on personal system and organization. Thus if hypersensitive data such as chemical formulae, combinations, patents, medical history of data, bank statements, password etc. are store in Cloud it may prone to attack by the attackers so it is very necessary to address the security of the system. In this paper we will discuss number of techniques to search the encrypted data over the Cloud so that others can search over the data and get the result in a relevant manner. This will eliminate the need to download the data and then decrypt it.

II. RELATED WORK

A. Survey

In this literature survey we are going to disscuss Security Analysis on One-to-Many Order Preserving Encryption Based Cloud Data Search: Below in literature we are Discussing some of them.

Many Techniques are developed so far now to search encrypted data over Cloud such as searchable encryption, PEKS, OPE etc. Searchable encryption is a technique to search encrypted Cloud over the cloud. There are two types of searchable encryption first one is searchable public key encryption abbreviated as SPE and other one is searchable symmetric encryption abbreviated as SSE.

SSE scheme was first introduced in [8] that successfully search encrypted data but it supports only single keyword, multi keyword search is not possible with SSE.[9] Proposed the searching techniques for multi keyword search.

[10] Introduced a technique to retrieve match files in the order of the relevance with the help of indexing technique. This will enable the quick search of documents that contain a given keyword.

[11] Introduced a technique for spelling error during the keyword search. The proposed scheme achieves fuzzy matching through algorithmic. This eliminates the need for predefined dictionary and also supports multi-keyword fuzzy search without increasing the complexity.

Boolean search is the traditional method to search which meets the effective data utilization is introduced by Cong Wang [12] which assures security guarantee. Relevance score is explored from retrieval to build a secure searchable index and develop a one-to-many order preserving mapping technique. Though the SSE helps in faster computation, but the security of the data is not sure since the shared key is used for both sender and receiver.

[1] Proposes a scheme called as Randomized Order Preserving Encryption (ROPE), a novel order preserving encryption scheme that leaks nothing beyond the order. SQL queries can be easily employed on encrypted data. Order-preserving symmetric encryption is proposed in [2] for permitting effective range queries upon encrypted data.

The first conventional cryptographic discussion of OPE appeared in the recent past in [3], where they formulated a security necessity for OPE and suggested a scheme that meets their security definition in an obvious and provable manner.

In [3], they include that OPE schemes cannot meet the standard notion of security called indistinguishability against chosen-plaintext attack (IND-CPA) as OPE scheme is not only deterministic, but also leak the order-relations among the plaintexts.

In [4] a scheme called mutable order-preserving encoding (MOPE) is proposed, to attain perfect IND-OCPA security. In [4], it is shown that IND-OCPA in reality is accomplishable with mutable cipher texts with respect to encodings.

In [5] a new scheme called DOPE is proposed, which adopts MOPE scheme with few changes in the security model. In [5], the performance of DOPE scheme is

All rights reserved by www.ijsrd.com

compared with querying on plain text database and observed that there is a time overhead.

B. Motivation

Data or file sent to Cloud is in encrypted format and downloaded in the encrypted format only and then it is decrypted by the owner. But what if user or a particular system wants to search something on cloud? Manually or traditionally we can say that download all the files that are relevant for the search and then decrypt it. If the item that is to be searched in not found then again the same process will be repeated. But this eliminated the security and privacy of the data and there is more irrelevancy of document. Thus we need a technique where user can search an item in the Cloud when the file is in encrypted format only that is there is no need to download each and every file and then decrypt it. Also searching should be in such way that it should return the user that the most relevant search first same as Google does. Traditional encryption search techniques such as Searchable Encryption, PEKS, OPE is preserves security of the document but have limitation with ranked search. For this reason we proposed a system that will effectively returns search result according to the most relevant documents. For ranked search in encrypted Cloud data, order preserving encryption (OPE) is most efficient tool to encrypt relevance scores of the inverted index. When we using deterministic OPE, the cipher texts will reveals the distribution of relevance scores. We will implement an advanced form of OPE to eliminate above limitation.

III. Proposed Approach Framework and Design

A. Problem Definition:

Now a day's users connected to the Internet may store their data on Cloud servers and let the servers manage or process their data. They can enjoy convenient and efficient service without paying too much money and energy However, no matter how advantageous Cloud computing may sound, large number of people still worry about the safety of this technology. If Cloud server gets direct access to all these users' data, it may try to analyses the documents to get private information

Theoretically, the server is not supposed to have access to sensitive data at all; therefore we should ensure the server has no access to leaking these data to an untrusted third party. Thus, sensitive data have to be encrypted before being outsourced to a commercial public Cloud However, encryption on sensitive data presents obstacles to the processing of the data. Information retrieval becomes difficult in the encrypted domain because the amount of outsourced files can be very large and traditional search patterns cannot be deployed to cipher text retrieval directly Users need to download all the data, decrypt it all, and then search keywords like plaintext retrieval

To overcome this, Searchable Encryption (SE) was proposed to make query in the encrypted domain possible while still preserving user's privacy.

B. Proposed System

We are proposing the scheme of searchable encryption based on one to many OPE. The modules are divided as below

1) Modules 1: General UI and framework Building

In this module, we will generate the basic framework required to form the application and we will generate the authentication and dashboard are for different entities.

2) Module 2: Encryption and Hashing

In this module the data owner files will be encrypted using symmetric algorithm like AES. The index for plain text files will be built in this phase and the keywords values will be wrapped in hash and will be uploaded over cloud. The batch inverted index table which is created for file is introduced with some garbage files IDs. This will create noise in inverted index tree and can lead to resistance to differential attack.

3) Module 3: Decryption and Searching

In this module the end user will execute the search query on the encrypted data over cloud. The documents will be extracted based on the one to many OPE. In this module user will get documents based on relevance score.

4) Module 4: Testing and Analysis

In this module the analysis will be performed and results matrix will be developed.

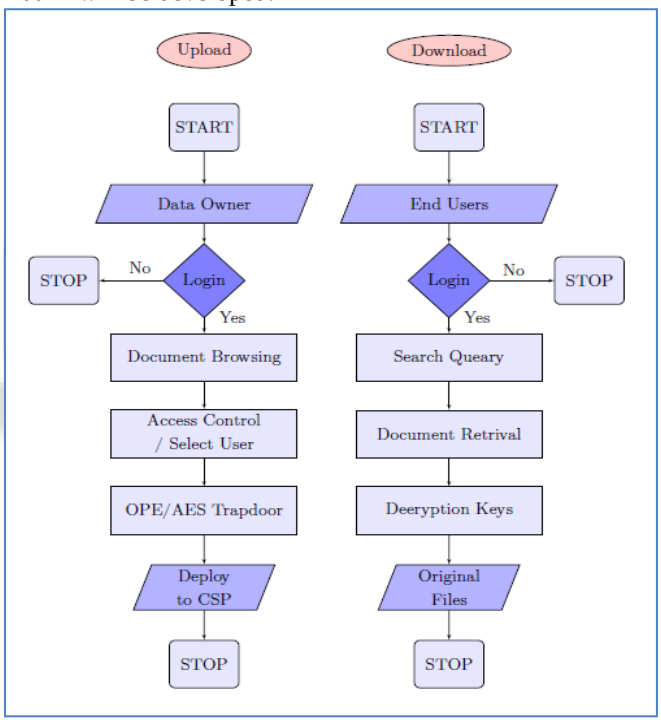

Fig. 1: Flowchart of System Architecture

C. Searchable Encryption and PEKS Scheme

Searchable encryption is a broad concept that deals with searches in encrypted data. Data stored at a server in an encrypted form, so whenever the user will search for a data searchable Encryption will allow others to search data without having access to plaintext. This technique avoids the downloading and decrypting of data from the server for searching purpose. The key factors and the role of Searchable Encryption is to protect the retrieved data, search query and search query outcome. One of the applications of searchable encryption is the email gateway. Let's take an example of email gateway. Say B sends a message to A. Now A's email gateway want to test if the email contains the word "urgent", so that it could route the email accordingly. . But off course it is expected that no other party other than A can decrypt and view the message. Searchable Encryption

All rights reserved by www.ijsrd.com

will allow gateway to test whether a given keyword is present in the email without learning anything else about the email. Searchable encryption is accomplished by Public Key Encryption with keyword Search (PEKS) scheme. Now if B is sending a message to A with keywords $W_1 \ldots W_k$. Keywords can be words in the subject line or the sender's address could be used as keywords. So B will send the message in encrypted form as shown below

$$E_{Apub} \mid [msg], PEKS(A_{pub}, W_1), \ldots, PEKS(A_{pub}, W_k)$$

Where A_{pub} is A's public key i.e. receiver's public key, msg is the email body. PEKS is an algorithm as Public Key Encryption with keyword Search. i.e. we can say the PEKS value is appended to the encrypted message. The PEKS values do not reveal any information about the message, but enable searching for specific keywords. Where A_{pub} is A's public key i.e. receiver's public key, msg is the email body. PEKS is an algorithm as Public Key Encryption with keyword Search. I.e. we can say the PEKS value is appended to the encrypted message.The PEKS values do not reveal any information about the message, but enable searching for specific keywords.

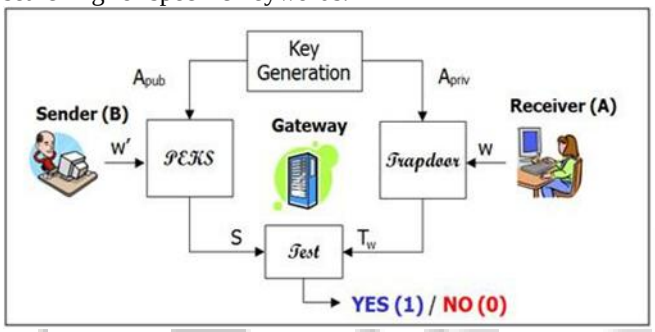

Fig. 2: Email Gateway Keyword Search

Below are the steps of Searchable Encryption:-

1) KeyGen(s) : This algorithm takes input a security parameter say 's' and generates a public/private key pair A pub and A priv.
2) PEKS(A_{pub} ,W) pub : This algorithm takes input as a public key A_{pub} and a word W (keywords), and produces a searchable encryption of W.
3) Trapdoor(A_{priv} ,W) priv : This algorithm takes input as A's i.e. receiver's private key and a word W and produces a trapdoor T_w .
4) Test (A_{pub}, S, T_W): This algorithm takes input as A's public key, a searchable encryption S = PEKS(A_{pub}, W') and a trapdoor TW = Trapdoor(A_{priv} , W) outputs 'yes' if $W=W_0$ and 'no' otherwise.

Searchable encryption is having some certain limitations that they are more prone to leaks information and have inadequate security definition. Some of the problem with Searchable encryption is fuzzy search, ranked search, multi-keyword search. Fuzzy search is a text retrieval technique based on fuzzy logic, it finds matches even where the keywords/search words are misspelled or in the case when only hint is provided for a search. Searchable encryption techniques create index of keywords and appends that index with the file. Searchable encryption techniques will work only for exact keywords.A small minor typo error will not allow retrieving the correct result, and it's common that user's input for search might not match the pre-defined keywords. Although traditional searchable encryption schemes allow users to securely search over encrypted data through keywords, these techniques support only Boolean search, without capturing any relevance of data files. Thus a multi-keyword search is not possible with the searchable encryption schemes. This approach is suffering from two main drawbacks when directly applied in the context of Cloud Computing. On the one hand, users, who do not necessarily have pre-knowledge of the encrypted Cloud data, have to post process every retrieved file in order to find ones most matching their interest, On the other hand, invariably retrieving all files containing the queried keyword further incurs unnecessary network traffic, which is absolutely undesirable in today's pay-as-you-use Cloud paradigm.

D. Order preserving encryption OPE

Fast ranked Search is most important feature of searching in the Cloud environment. Order preserving encryption (OPE) is one practical way of supporting fast ranked search. OPE is a symmetric cryptosystem, therefore it is also called order preserving symmetric encryption (OPSE). The order preserving property means that if the plaintexts x1 < x2, then the corresponding cipher texts E(x1) and E(x2) satisfy E(x1) < E(x2) where E is an order-preserving encryption function. Order preserving encryption function preserves numerical ordering of the plaintexts. OPE has a long history in the form of one-part codes, which are lists of plaintexts and the corresponding cipher texts, both arranged in alphabetical or numerical order so only a single copy is required for efficient encryption and decryption. One-part codes is a code where the plain text is arranged in alphabetical or numeric order with their code group(encrypted data) in alphabetical or numerical and other systematic order. Such Scheme provides an efficient range queries on encrypted data. That is, a remote untrusted database server is able to index the (sensitive) data it receives, in encrypted form, in a data structure that permits efficient range queries i.e. asking the server to return cipher texts in the database whose decryptions fall within a given range, say [a, b]. Range queries are more efficient than the linear as it act on a particular range of data rather than the whole data. However, the security definition and the constructions of OPE in [3] and [7] are based on the assumption that OPE is a deterministic encryption scheme which means that a given plaintext will always be encrypted as a fixed cipher text. However, deterministic encryption leaks the distribution of the plaintexts, so it cannot ensure data privacy in most applications. In the OPE scheme, binary search based on a random HGD sampler is used to preserve the order of encryption. It uses a Number equations consecutively with strategy of recursively mapping "range gaps" to "domain gaps" in binary search way which is used to create a virtual barrier between between two consecutive points in the range or domain.

Below are the steps for OPE algorithm:-

1) The Binary Search based OPE algorithm takes input as secret key K which is generated by using randomized key generation algorithm, plain text space as $D = \{0,1,2..., M\}$ and cipher-text space as $R = \{1,2,..., N\}$ and m as plain text.
2) The algorithm at first maps the middle range gap "y" to a domain gap. y is the gap between the middle two range point.

All rights reserved by *www.ijsrd.com*

3) Random coins are generated by using random coin generator i.e. *TapeGen*() algorithm that takes the input as K, D, R and y.
4) Hyper geometric probability distribution function is used to generate x.
5) If the input domain point m that represents the plain text is less than or equal to domain gap x that represents the cipher text that is generated from above steps then the algorithm recurses on the lower i.e. respective upper half of the range and the lower i.e. respective upper part of the domain.

E. Probabilistic OPE

The purpose of both OPE and One-to-Many OPE is to prevent information leakage to the Cloud server. If a deterministic OPE is used to encrypt relevance scores, the cipher texts will share exactly the same distribution as its plain counterpart, by which the server can specify the keywords. To address this limitation of deterministic OPE, the above OPE deterministic algorithm is modified as One-to-Many OPE. For m i.e. plaintext relevance score, the "One-to-Many OPE" first executes above deterministic OPE algorithm to select a bucket for "m", and then randomly chooses a value in the bucket as the cipher text. The randomly choosing procedure in the bucket is seeded by the unique file IDs together with the plaintext m, and thus the same relevance score in the inverted Index will be encrypted as different cipher texts. Below diagram shows the difference between deterministic OPE and One-to-Many OPE.

In [6], comparison between plaintext distribution and cipher text distribution obtained by two kinds of OPE on a particular keyword. And it is observed in [6] that deterministic OPE makes the plaintexts and the cipher texts share the same distribution, which would make it too easy for an attacker to get the exact keyword's information. And in one-to-many OPE the size of the cipher text domain is large so One-to-Many OPE can flatten the distribution of plaintexts.

F. Attacks on OPE

In OPE it has been observed that the distribution of plain text has been successfully hidden but the one-to-many OPE does not ensures strict cryptanalysis. In [SECURITY ANALYSIS ON ONE-TO-MANY OPE-BASED CLOUD DATA SEARCH] it has been observed that by doing analysis on the differences between the cipher texts, the Cloud server can get idea on the distribution of the plaintexts. As discussed in one-to-many order preserving encryption (OPE) scheme plain text m is mapped into many possible cipher texts belonging to a fixed bucket and the cipher text is randomly selected in the bucket. So the plaintext value having high frequency will contain more cipher texts in the bucket and those having low frequency will contain less cipher texts in the bucket. Thus the plaintext profile can be easily determined and portrayed by the denseness of the cipher text in the bucket. Differential Cryptanalysis is one of the techniques that attempts to find the key used for encryption. Density of a particular cipher text can be easily revealed by taking the difference between neighboring cipher texts. Thus the Cloud server can easily determine the distribution of plain text from the differential cipher text. This attack is called as differential attack. The most important thing in differential attack is to find the change point between the neighboring sequences of cipher text. Analysis to find the change point in called as Change Point Analysis (CPA).

Below steps describe the differential attacks:-

1) If the original cipher text sequence as $c_1, c_2, \ldots, c_L$. Sort the cipher text sequence in ascending order based on the index as $c_{i1} \leq c_{i2} \leq \ldots \leq c_{iL}$.
2) A Differential Sequence is calculated as:

$$d_1 = c_{i2} - c_{i1},\ d_2 = c_{i3} - c_{i2},\ \ldots,\ d_{L-1} = c_{iL} - c_{iL-1} \quad (1.1)$$

3) For each d_i the CUSMU (cumulative sum) Sequence is calculated. First the average value is calculated as below:

$$\bar{d} = \frac{1}{L-1} \sum_{i=1}^{L-1} d_i \quad (1.2)$$

The initial value of cumulative sum as $S_0 = 0$. The other Cumulative sum values are calculated in a recursion way as shown by below equation:

$$S_i = S_{i-1} + (d_i - \bar{d}) \quad (1.3)$$

Where i = 1, 2,..., L - 1

If we can locate the change points of the distribution of the differential cipher texts, we can determine the boundaries of the buckets in the cipher text range R = {1, 2, .., N}. With these boundaries, the histogram of the plaintexts can be easily estimated by counting the number of cipher texts belonging to each bucket. Therefore, the Cloud server may reconstruct the distribution of plaintexts from the differential cipher texts, which we call "differential attack". The key issue in "differential attack" is locating the change.

Change points are located by using the bootstrap sample and the difference between the sequences.

Changes are sorted in ascending order and the histogram of plain text is generated that will count the number of cipher text drop in each interval.

IV. RESULT AND ANALYSIS

A. Result

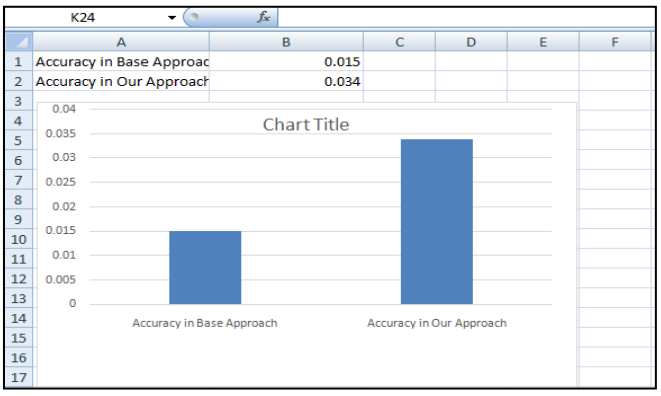

Fig. 3: Final Result of System

B. Analysis

The experiment evaluation is done on windows 7 operating system with 4 GB RAM with i5 processor.

The total documents uploaded to cloud are 28 and total keywords of interest that is Nw = 123.

The total representative keywords considered for estimating identifying accuracy that is Nt = 71

Background strength that is alpha (a) = 0.66

so

$$I\ A(w) \text{ is } (Nw - 1)/Nw \quad (2.1)$$

All rights reserved by www.ijsrd.com

$$I\,A(W) = (123 - 1) / 71 = 1.71$$
$$AIA = I\,A(W) / Nt * alpha \quad (2.2)$$
$$AIA = 1.71 / 71 * 0.66 = 0.034$$

So the accuracy for our approach is0.034 which is higher as compare to base paper approach that is 0.015.

V. CONCLUSION

In this paper we investigate efficient data retrieval from the cloud database. The demand for cloud computing rises day by day, consumers can store their files and can retrieve it since it is valuable and soothing process. As the demand increases it is required for the search service to allow multi keyword query and provide results based on similarity ranking to meet effective data retrieval. Here we proposed one to many order preserving encryption (OPE), for applications of searchable encryption, which flattens the distribution of the plain text to solve the problem of secured multi keyword search over encrypted cloud data (ECD). The result shows that data can be retrieved faster by efficient multi keyword search from remotely stored encrypted data. In future work, at extending these ideas to design secure methods of probabilistic OPE and schemes for search in encrypted data.

REFERENCES

[1] A. Boldyreva, N. Chenette, and A. O'Neill, "Order Preserving Encryption revisited: Improved security analysis and alternative solutions," In Advances in Cryptology. Berlin, Germany, Springer-Verlag, 2011, pp.578-595.
[2] B. Wang, S. Yu, W. Lou and Y.T.Hou, "privacy-preserving multi-keywords fuzzy search over encrypted data in the cloud," in proc. IEEE INFOCOM Apr/May 2014 pp.2112-20120.
[3] B. Wang, S. Yu, W. Lou and Y.T.Hou, "privacy-preserving multi-keywords fuzzy search over encrypted data in the cloud," in proc. IEEE INFOCOM Apr/May 2014 pp.2112-20120.
[4] C.Wang, N. Cao, K. Ren and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced Cloud data," IEEE Trans.Parallel Distrib. System Vol. 23 No. 8 pp. 1467-1479 Aug. 2012.
[5] D. X song, D. Wagner, and A. Perring, "Practical techniques for searches on encrypted data, "in proc. IEEE symp. Secure Privacy May 2000, pp.44-55.
[6] J. Yu, P. Lu, Y. Zhu, G. Xue, M. Li, " Toward Secure Multikeyword Top-k Retrieval over Encrypted Cloud Data", IEEE Transactions on Dependable and Secure Computing, vol. 10, No. 4, July 2013.
[7] K. Srinivasa Reddy and S. Ramachandram "A New Randomized Order Preserving Encryption Scheme" In International Journal of Computer Applications (0975 – 8887) Volume 108 – No 12, December 2014.
[8] K. Srinivasa Reddy and S. Ramachandram "A Novel Dynamic Order Preserving Encryption Scheme" In Networks & Soft Computing (ICNSC), 2014 First International Conference on 19-20 Aug. 2014.
[9] K. Srinivasa Reddy and S. Ramachandram "A Novel Dynamic Order Preserving Encryption Scheme" In Networks & Soft Computing (ICNSC), 2014 First International Conference on 19-20 Aug. 2014.
[10] M. Abdalla et al., "Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions" in Advances in Cryptology. Berlin, Germany: Springer-Verlag, 2005, pp. 205–222.
[11] Raluca Ada Popa, Frank H. Li, Nickolai Zeklovich "An Ideal-Security Protocol for Order Preserving Encoding" In proc. Of the 34th IEEE Symposium on Security and Privacy.
[12] R. Agrawal, J. Kieman, R. Srikant, and Y. Xu, '' Order Preserving Encryption for Numeric Data" In Proc. ACM SIGMOD Int. Conf. Manage Data, 2004 pp.563-574.

All rights reserved by *www.ijsrd.com*