Symmetric Searchable Encryption and Data Retrieval in Cloud Computing

R.Prabhu[1]
Assistant Professor
Department of Computer Science & Engineering
AAA College of Engineering and Technology, Sivakasi

Dr.S.Rajesh[1]
Associate Professor
Department of Information Technology
Mepco Schlenk Engineering College, Sivakasi

ABSTRACT- Cloud computing is a specialised form of distributed computing that introduces utilization models for remotely provisioning scalable and measured resources. The driving motivation behind the cloud computing is to provide IT resources as a service that encapsulates other IT resources. Data service outsourcing is one of the service that is economically enabled by the cloud computing. But in order to protect data privacy sensitive data has to be encrypted before outsourcing to the commercial public cloud. Data encryption protects data security to some extent, but this may lead to a compromise on the part of efficiency of storage and retrieval on the server. This paper analyzes various searchable encryption schemes where data owner itself is responsible for his data security. These searchable encryption schemes allow retrieval of encrypted data over the cloud thereby making data retrieval fast and efficient. These schemes also guarantee high security and efficiency.

Keywords*-Data outsourcing, data encryption, searchable encryption*

1. INTRODUCTION

Cloud [1] can be thought of as a distinct IT environment that is designed for the purpose of remotely provisioning scalable and measured IT resources. The term originated as a metaphor for the internet which is, actually, a network of networks providing remote access to a set of decentralized IT resources. It is a way to increase the capacity or add capabilities dynamically without investing in new infrastructure.

Adoption of cloud computing [2] is threatened by unresolved security issues that affect both the cloud provider as well as the cloud user. The main threat on data privacy lies in the cloud itself. Cloud users are concerned about the risks of the security of sensitive data and loss of direct control over the systems if not properly secured. When users outsource their private data onto the cloud, the cloud service providers are able to control and monitor the data and the communication between the users and the cloud. To ensure the data privacy [3], users generally encrypt the data before outsourcing it onto the cloud but it makes effective data utilization a very challenging task. Even if the data is encrypted before outsourcing to cloud users still need to communicate with the cloud and allow the cloud to operate on encrypted data which may cause leakage of sensitive information. Another issue that may be dealt with is if a user wants to retrieve a data file based on some keywords the system may be efficiently able to search over that encrypted data and only the data or files relevant to users should be sent to the users. This paper discusses the searchable encryption techniques and effective data retrieval from the cloud.

This paper is summarized as follows: section II describes about what is searchable encryption? Section III discusses the number of schemes or different techniques which are basis for searchable encryption i.e. which are required to develop an efficient searchable encryption algorithm. Section IV outlines the problem statement and its discussions. Section V finally gives a step by step analysis of how to create a symmetric searchable encryption algorithm. Finally section VI gives conclusions.

2. SEARCHABLE ENCRYPTION

Now a days data is encrypted by the users before outsourcing it on to the cloud.. They discussed two different approaches for the problem of searching on encrypted data. First one is to build up an index that for each keyword W to be searched, the index lists the documents that contain W. An alternative is to perform a sequential scan without an index. According to them the disadvantage of using an index is that the storing and updating the index has a substantial overhead but sequential scan is not efficient enough when the data size is very large. For, very large size databases a common technique to speed up the searching is to use a pre-computed inverted index that maps each keyword with the data file in which that keyword appears.

Many schemes have been generated to make search on encrypted data possible. A number of algorithms based on symmetric searchable encryption schemes have been proposed in[6][7][8][9][10]. Using these earlier SSE schemes [5][7] enable users to retrieve the cipher text securely but they provide only Boolean keyword search, i.e., these schemes can only search for if a keyword exists in a file or not and do not consider to check for the relevance of the keyword with the file. When we apply these encryption schemes directly on outsourced data in cloud environment they prove to have two major drawbacks. First, for each search request, users have to incur huge post processing overhead as users do not have any knowledge of encrypted cloud data. Second, it causes unnecessary network traffic because server on each search request will search out and send all the files just based on presence or absence of keyword in the files which is undesirable for cloud's pay-per-use model. Afterwards an order preserving encryption scheme was proposed which is a deterministic encryption scheme where encryption function preserves numerical ordering of plaintexts. OPE allows the queries to process exactly and efficiently as for unencrypted data and also allows range queries. But Later it was analysed that only OPE scheme by default leaks data privacy. To overcome the drawbacks of OPE scheme further it was modified as OPM (Order Preserving Mapping) in which same plaintext is mapped to a different random value in a particular range size. J. Yu. et al [6] proposed a searchable symmetric encryption scheme TRSE (two-round searchable encryption) which fulfils the requirement for multi-keyword search and top-k relevant file retrieval over encrypted cloud data. Some others in [12] C.Wang et al [11] proposed a ranked searchable symmetric encryption (RSSE) scheme which integrates order- preserving symmetric encryption (OPSE).

3. SEARCHABLE ENRYPTION SCHEMES

3.1 Order Preserving Encryption Scheme (OPE)

Order Preserving Encryption (OPE) is a deterministic encryption scheme which uses an encryption function that preserves the numerical ordering of plaintext values. Boldyreva et al [gives the first cryptographic study of OPE scheme and provide a construction that is provably secure under the security framework of pseudorandom function. The reason for interest in such schemes is that they allow efficient range queries on encrypted data i.e a remote untrusted data base server is able to index the encrypted data it receives in a data structure that permits range queries.

Basically, OPE is a method of encrypting data so that it is possible to make inequality comparisons on encrypted data without decrypting it. It is a deterministic symmetric encryption scheme whose encryption algorithm preserves numerical ordering of plaintexts. Let M and N be finite ordered sets. We say that OPE is an order preserving encryption scheme with plaintext space M, ciphertext space N, and key space K. For any choice of keys k ϵ K any choice of inputs x1, x2 ϵ M, the following holds: If x1 < x2 then OPE (k, x1) < OPE (k, x2)

Let us have a random-order preserving injective function from M to N, where │M│< │N│. We can consider M the set {1, 2... M} and N {1, 2... N}. Now choose M elements of N randomly and put them in order. The injective function

f: M→N is simply this ordered set. To encrypt i ϵ M, just output the ith element of this list.

As OPE preserves the order of plaintexts therefore it is not a perfectly secure encryption scheme since ciphertexts leak the order information of plaintexts. There are various constructions of OPE schemes. Proposed an OPE algorithm which first generates a sequence of random numbers and then encrypts an integer x to the sum of first x random numbers. defines an OPE algorithm based on a sequence of strictly increasing polynomial functions. The encryption of an integer x is the output of iterative operations of those polynomial functions on x.

OPE has several caveats. The most problematic is the adversary's ability to guess approximately where the underlying plaintext of a ciphertext lives in the plaintext space. And it sometimes also reveals to certain attackers half the bits of plaintext given its ciphertext. The security of an encryption scheme depends on how precisely the adversary can predict the bits in the plaintext. Another problem against the OPE scheme based on IND-CPA is to reverse the order of chosen plaintext attack, i.e. the adversary is given the ciphertext and subsequently chooses the plaintexts.

3.2 Order Preserving Mapping (OPM)

In order to overcome the problems, the OPSE scheme has to be modified. To reduce the amount of information leakage one-to-many OPSE scheme [11] is required. As in OPSE scheme, cipher texts leak the order information of plaintexts, one-to-many order preserving encryption scheme uses the unique file identifier (ID) as an additional random seed value so that the same plaintext will be mapped to a random value within the randomly assigned interval in range R rather than the same cipher text. Because of the unique file identifier to be included in random selection of seed value the same plaintext will not be deterministically assigned to the same cipher text but it will be assigned in a random interval in range R. The mapping scheme should be as random as possible so that the score distribution for a specific keyword cannot be predicted. The range size R is as large as possible so that the specific characteristics are not preserved.

3.3 Homomorphic Encryption

Homomorphic encryption systems were developed to perform calculations on encrypted data without decrypting it. It allows specific types of computations to be carried out on cipher texts and the result is the ciphertext of the result of the same operations performed on the plaintext. That is, Homomorphic encryption [6] allows computation of ciphertext without knowing anything about the plaintext to get to the encrypted result. An encryption is homomorphic if from Enc(a) and Enc(b) it is possible to compute Enc(f(a, b)). Where f can be:

+, *, ⊕ operations, without using private key. Discussed

Homomorphic encryption scheme which requires a key generation algorithm that produces public key and secret key, an encryption algorithm which takes as input the plain text and encrypts the plaintext to get cipher text.

There are two types of Homomorphic encryption: Somewhat Homomorphic Encryption (SHE) and Fully Homomorphic Encryption (FHE). Each type differs in the number of operations that can be performed on encrypted data. FHE allows for an unlimited, arbitrary number of computations (both addition and multiplication) to be performed on encrypted data. SHE cryptosystems support a limited number of operations and are faster and more compact then FHE cryptosystem. According to the operations that allow assessing on raw data, homomorphic encryption can be differentiated into Additive Homomorphic Encryption and Multiplicative Homomorphic Encryption.

An additively homomorphic scheme is one with a ciphertext operation that results in the sum of the plaintexts. That is, Encrypt(a) +Encrypt(b) = Encrypt(a+b) where the decryption of both sides yields the sum of the plaintexts. A multiplicatively homomorphic scheme is one that has an operation on two cipher texts that results in the product of the plaintexts. That is, Encrypt(a) * Encrypt(b) = Encrypt(a*b) where the decryption of both sides yields the product of the plaintexts. The most famous multiplicatively homomorphic scheme is RSA encryption. A fully homomorphic encryption (FHE) is one which supports both multiplicative and additive operations and is far more powerful. Using such a scheme, any circuit can be evaluated homomorphically, and effectively allowing the construction of programs which may be run on encryptions of their inputs to produce an encryption of theiroutput. Since such a program never decrypts its input, it can be run by any untrusted body and its inputs and internal states need not be revealed. In context of cloud computing, while outsourcing an efficient and fully homomorphic cryptosystem will have great practical impacts.

3.4 Two Round Searchable Encryption (TRSE)

Searchable symmetric encryption schemes

[4][11] based on OPE schemes employ a server- side ranking to improve the efficiency of retrieval over encrypted cloud data. But this server-side ranking based on OPE violates the privacy of sensitive information, as OPE scheme leaks the information about order of bits in the plaintext. But in an security oriented third party cloud computing scenario security can never be the tradeoff for efficiency. Therefore ranking of files based on the keywords entered by user should be done at the user side.

4. PROBLEM STATEMENT

[6] Proposed a new searchable encryption scheme where the data owner encrypts the searchable index with Homomorphic encryption. When the cloud server receives a query consisting of multi keywords , it computes the scores from the encrypted index stored on the cloud and then returns the encrypted scores of the files to the data user. Now, the data user decrypts the score and picks the top-k highest scoring files' identifiers to request the cloud server. The retrieval takes a two-round communication between the cloud server and the data user hence the name two round searchable encryption.

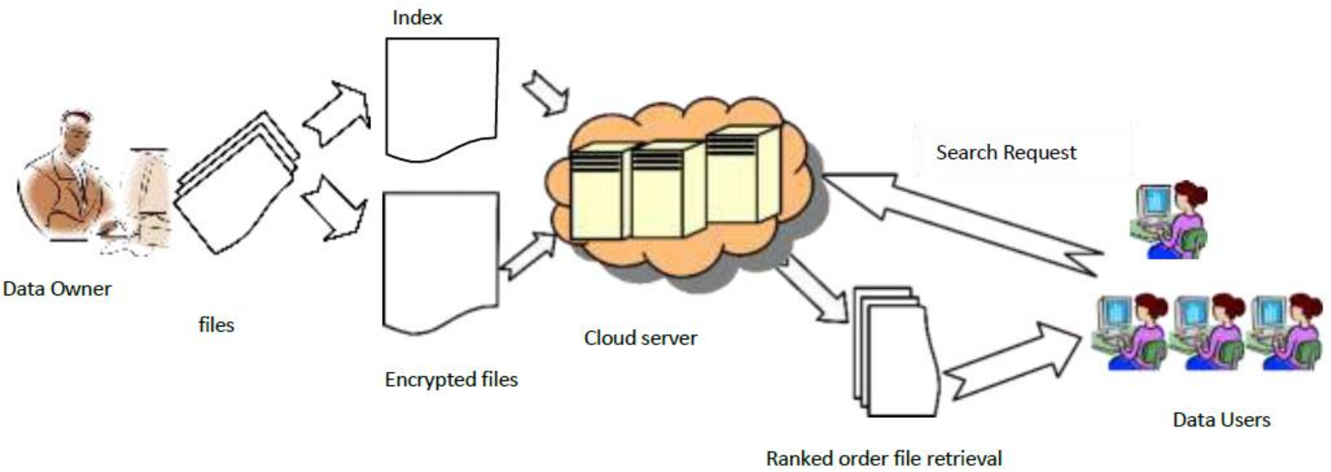

Fig. 1 Architecture for symmetric searchable encryption and Data Retrieval over cloud data

4.1 Architecture for Search and Retrieval over Encrypted Cloud Data

In [6][[11][4] the authors proposed an architecture for search over encrypted data and its retrieval which involves three different entities in a cloud computing system that hosts data services : Data owner, Data User and Cloud server. Cloud server hosts third party data storage and retrieval services. Since, data may contain sensitive information, cloud servers cannot be fully trusted in protecting data. For this reason, outsourced files must be encrypted.

Data owner has a collection of n data files C=(F1,F2,.....,Fn) that he wants to outsource on the cloud server in encrypted form and the data search and retrieval service based on certain encrypted keywords is kept with data owner and authorized data users. Before outsourcing, the data owner first builds a secure searchable index I from a set of m distinct keywords W= {w1, w2, w3,.........., wm) taken from file collection C, and store both the encrypted file collection C and the Index I on the cloud server.

This architecture assumes the proper authorization between the data owner and the data user has been already done. When a user wants to search or retrieve a file from the collection of files on the cloud server based on the keywords w, the user submits the search request to the cloud server. Now, the cloud server is responsible for searching the files and retrieving for the user based on the relevance of the keyword submitted by the user. After getting the file which is in encrypted form the data user decrypts the file and gets the original needed file. Hence the entire process requires five steps to search over the encrypted text and retrieve the needed data. These steps are as follows:

KeyGeneration_algorithm() is run by data owner to generate the random keys for encryption and decryption of the data. It takes user details as input and generates keys as output. Also encrypts the files to be outsourced to the cloud server using the the public key generated by the algorithm.

CreateIndex_algorithm() is run by data owner to create indexes of all the unique keywords occurring in the files. It will also generate a mapping table called inverted index table that will store the mapping of a keyword to a particular file being outsourced that contains the keyword. It takes unique keywords as input and returns an index as output.

Create_SearchRequest() is run by a data user. It takes a user's secret key and a keyword as input and returns a search request as output.

Search() is run by cloud server. It takes search request as an input and returns search result as an output. That is extended by giving a value n to have top n data files as output.

Decrypt() is run by a data user. It takes encrypted file and secret key as inputs and returns original file as output.

4.2 Relevance Scoring and Ranking Function

To search for a given keyword w in the file collection, an authorized user generates and submits a search request in a secret form to the cloud server. Upon receiving the request, the cloud server is responsible to search the index and return certain set of files to the user. The search results are returned according to certain ranked relevance criteria to improve retrieval accuracy for users who do not have any prior knowledge on file collection [4][6].

Let C denotes the collection of files to be outsourced, denoted as set of n data files. W denotes the distinct keywords extracted from file collection C, denoted as set of m words W= (w1, w2,…,wm). Let Id(F_j) be the identifier of file F_j that can help uniquely locate the actual file. Let I be the index built from the file collection and T_{w_i} be the trapdoor generated by a user as a search request of keyword w_i. F(w_i) be the set of identifiers of files in C that contain keyword w_i and N_i the number of files containing the keyword w_i and $N_i = |F(w_i)|$.

For information retrieval an inverted index structure is used for indexing the keywords and stores the list of mappings from the keywords to the corresponding files in which those keywords are encountered. For searching a text, the task of relevance of a file to a keyword is done by using some numerical score precalculated on the basis of ranking function. A ranking function is used to calculate relevance scores of matching files to a given search request. Several ranking models have been proposed to score and rank files in Information Retrieval (IR) community. The most widely used among them uses TF x IDF rule, where TF (term frequency) is simply the number of times a given term or keyword appears with in a file and IDF (Inverse Document Frequency) is obtained by dividing the number of files in the whole collection by the number of files containing the term. A formula to calculate the relevance scores that is commonly used and widely seen in the literature[4] is defined as follows:

$$\mathbf{Score(k,F_d)=(1/|F_d|)*(1+\ln(TF_{d,t}))*\ln(1+N/Ft)}$$

Where k denotes the keyword to be searched, $TF_{d,t}$ denotes the term frequency of term t in file F_d, Ft denotes number of files containing term t and N denotes total number of files. $|F_d|$ denotes the length of the file F_d.

4.3 Semantic Relationship

Authors in [26] proposed the association ratio for measuring word association norms, based on the concept of mutual information. This could be used to calculate the semantic relationship of the keyword in the search request with the files in which that keyword exists. The degree of such semantic relationship between the terms in the collection could be calculated effectively using data mining. Church et al in calculated the mutual information for two terms, x and y as:

$$\mathbf{l(x,y) = \log_2 (P(x,y)/P(x)P(y))}$$

Here, P(x,y) is the probability of observing x and y together. P(x) and P(y) are the probabilities of observing x and y independently in the collection. The higher the semantic relationship between x and y the larger the mutual information l(x,y). This value of the mutual information is normalized into a value of relationship in the interval [0, 1]. Then semantic relationship library will be constructed as a weighted graph structure.

5. METHODOLOGY

There are many traditional searchable encryption algorithms available that allow the authorized users to securely search over encrypted outsourced data without decrypting it. But they did not take into consideration the multiple keywords and the relevance of the files to the keywords. In order to have an efficient ranked keyword search based on the relevance of the keyword to be searched a number of schemes exist. The order preserving encryption scheme preserves the order of encryption based on certain criteria but could leak the order information of the plaintext. To avoid the drawbacks of the deterministic properties of order preserving encryption a modification in the form of one-to-many order preserving encryption scheme can be used which always maps a plaintext to a different value in some range R. The efficiency of this one-to-many order preserving scheme depends on the range size R and the domain of the score of plaintext. Homomorphic encryption was designed to perform calculations on encrypted data without decrypting it. It also allows computation of ciphertext without knowing anything about the plaintext to get to the encrypted result. And a two round searchable encryption scheme. In a TRSE scheme the cloud server when receive a search request and calculates the relevance score. Server sends encrypted scores of files to the data user to pick top-k scoring files and requests the server to send. The retrieval takes a two-round communication between the cloud server and the data user hence the name two round searchable encryption. This entire procedure as analysed for secure searchable encryption and top-k retrieval of encrypted data files could be expressed as follows:

STEP1: KeyGeneration_algorithm() should be run to generate public/private key pair. This algorithm should be able to generate large random keys as the strength of algorithm lies in the strength of keys. RSA could be used to generate the key pair which is also multiplicatively Homomorphic. This architecture assumes that a prior authorization between the data owner and data user has been already done. There should be no issues like who can access what and how keys are to be distributed between the data owner and data user.

STEP2: Data Owner encrypts the data files using an encryption algorithm and the keys generated in step 1.

STEP3:An index file containing the keywords, of all the files being outsourced, is created using CreateIndex_algorithm(). Then this index file is encrypted using the same encryption algorithm. Basically an inverted index structure is created which includes list of mappings from keywords to set of files that contain the keyword. Further a relevance score is calculated for each file with respect to a specific keyword. In order to achieve a higher degree of security this index construction is done using order-preserving mappings where original scores are replaced with one that is calculated using the OPM scheme. Based on these relevance scores top-k retrieval is done.

STEP 4: The files as well as their inverted index are outsourced to the cloud server.

STEP 5: A search request can be generated by an authorized user to search for top-k retrieval from the cloud server. This is done using Create_SearchRequest() algorithm which takes the keyword to be searched form the user and generates a search request query to search out the required file from the encrypted files on the cloud server.

STEP 6: Cloud server when receives a search request runs the search() algorithm to find out the relevance scores of all the files containing the keyword for which search request has been generated.

STEP 7: In a TRSE scheme the relevance scores in the first round are sent to the user by the cloud server.

STEP 8: Based on these scores the user tells the cloud server which files are required?

STEP 9: In the second round only the relevant files as required are sent to the user. This overall communication between the cloud server and user takes two rounds and hence the name Two-Round Searchable Encryption.

STEP 10: The user decrypts these encrypted files on his own system using a corresponding decryption algorithm and the key.

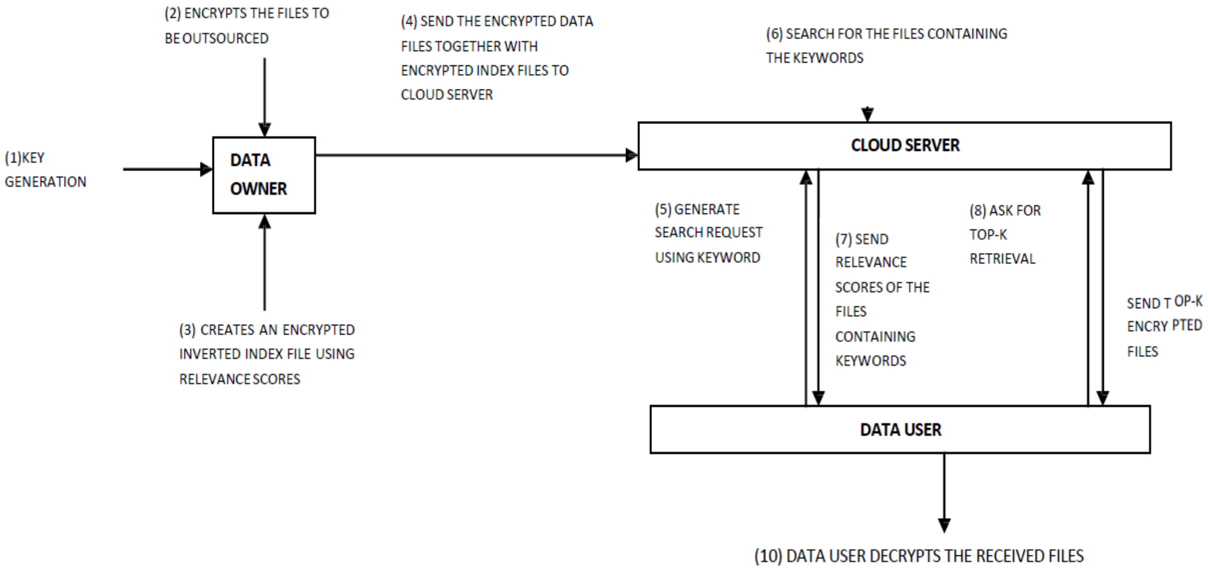
(2) ENCRYPTS THE FILES TO BE OUTSOURCED
(4) SEND THE ENCRYPTED DATA FILES TOGETHER WITH ENCRYPTED INDEX FILES TO CLOUD SERVER
(6) SEARCH FOR THE FILES CONTAINING THE KEYWORDS
(1)KEY GENERATION
DATA OWNER
CLOUD SERVER
(5) GENERATE SEARCH REQUEST USING KEYWORD
(7) SEND RELEVANCE SCORES OF THE FILES CONTAINING KEYWORDS
(8) ASK FOR TOP-K RETRIEVAL
SEND T OP-K ENCRY PTED FILES
(3) CREATES AN ENCRYPTED INVERTED INDEX FILE USING RELEVANCE SCORES
DATA USER
(10) DATA USER DECRYPTS THE RECEIVED FILES

Fig 2. Procedure for symmetric searchable encryption and data retrieval over cloud server

6. CONCLUSIONS

In this paper we have studied the searchable encryption technique for top-k relevant files retrieval. Various schemes that are used for the searchable encryption have been discussed in detail. The scheme to search on the encrypted data without decrypting it provides a secure mechanism to outsource the data on the public clouds. Clouds these days are known for the advantages they provide but the security risks prevent many users to take these advantages. This searchable encryption scheme gives the data owners a benefit that he himself encrypts the data making it secure and cloud server is no where going to decrypt it. There is no information leakage in the scheme and provides the privilege to have the original data to authorized user only.

7. REFERENCES

[1] P. Mell, T. Grance,"The NIST Definition of Cloud Computing", Version 15,10-7-09

[2] B.Reddy Kandukuri, V. Ramakrishna, A. Rakshit, "Cloud Security Issues", 2009 IEEE International Conference on service computing(SSC 2009), September 2009, Bangalore. pp. 517-520.

[3] S. Subashini S, V. Kavitha, "A survey on security issues in service delivery models of cloud computing", Journal of Network and Computer Applications Volume 34 issue 1 January 2011, pp.1-11.

[4] C. Wang, N. Cao, J. Li, K. Ren, W. Lou, " Secure Ranked Keyword Search over Encrypted Cloud Data", proc. IEEE 30th International Conference on Distributed Computing Systems(ICDCS '10), 2010.

[5] D.Song, D. Wanger, and A. Perrig, "Practical Techniques for Searches on Encrypted Data" Proc. IEEE symp. Security and Privacy, 2000.

[6] .J. Yu, P. Lu, Y. Zhu, G. Xue, M. Li, " Toward Secure Data", IEEE Transactions on Dependable and Secure Computing, vol. 10, No. 4, July 2013.

[7] D. Boneh, G. Crescenzo, R. Ostrovsky, G. Persiano, " Public Key Encryption with Keyword Search", Proc. International Conference on Theory and Applications of Cryptographic Techniques (Eurocrypt), 2004.

[8] R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," Proc. ACM 13th Conf. Computer and Comm. Security (CCS), 2006.

[9] O S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+r: Top-k Retrieval from a Confidential Index," Proc. 12th International Conference on Extending Database Technology: Advances in Database Technology (EDBT), 2009.

[10] A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A.L. Varna, S. He, M. Wu, and D.W. Oard, "Confidentiality-Preserving Rank-Ordered Search," Proc. Workshop Storage Security and Survivability, 2007.

[11] C. Wang, N. Cao, K. Ren, W. Lou, " Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data", IEEE Transactions On Parallel and Distributed Systems, Vol. 23, No. 8, August 2012.

[12] P.S. Priya, D. Preethi, J. Priya, B. Shanthini, " Retrieval of Encrypted Data Using Multi Keyword Top-k Algorithm", International Journal of Scientific and Research Publications, Vol. 4, Issue 4, April 2014.