Volume No:3, Issue No:5 (October-2017) ISSN No : 2454-423X (Online)

International Journal of Research in Advanced Computer Science Engineering

A Peer Reviewed Open Access International Journal

www.ijracse.com

Privacy-Preserving and Dynamic MultiKey Generation over Encrypted Cloud Data

Mannem Venkata Krishna
M.Tech Student
Department of Computer Science & Systems Engineering
AU College of Engineering (A), Andhra University, Visakhapatnam.

S.Jhansi Rani
Assistant Professor
Department of Computer Science & Systems Engineering
AU College of Engineering (A), Andhra University, Visakhapatnam.

ABSTRACT

Cloud computing has emerged as an important paradigm for managing and delivering services efficiently over the Internet. Convergence of cloud computing with technologies such as wireless sensor networking and mobile computing offers new applications of cloud services. An experiment deliberately imposes a treatment on a group of objects or subjects in the interest of observing the response.

This differs from an observational study, which involves collecting and analyzing data without changing existing conditions. Because the validity of an experiment is directly affected by its construction and execution, attention to experimental design is extremely important.

Due to the abundant characteristics of Cloud Computing; viz. on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service; it is becoming a buzz word these days. Many companies are providing cloud services. However confidential and sensitive data has to be secured. So the outsourced data has to be encrypted for security of the document retrieval. The existing multi-keyword ranked search scheme over encrypted cloud data supports dynamic update operations like deletion and insertion of documents. The data owner generates an exceptional tree-based catalog composition together with "Greedy Depth-first Search" criteria to make successful multi-keyword search. Achieving parallelism is the limitation of the existing system.

In this thesis we extend the multi-keyword ranked search scheme with secure Dynamic Key generation along with the vector space model and the widely-used TF_IDF model for the index construction and query generation. The dynamic key generation favors parallelism by allowing multiple users retrieve the same encrypted cloud data.

INTRODUCTION

Cloud computing technology is a service-based, Internet-centric, safe, convenient data storage and network computing service. It is an internet-based model for enabling a convenient and on-demand network access to a shared pool of configurable computing resources. Cloud computing is the use of resources that are delivered as a service over a network. The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. The cloud providers manage the infrastructure and platforms on which the applications run. End users access cloud-based applications through web browser or a lightweight desktop or mobile app while the business software and user's data are stored on servers at a remote location. Cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models. A general approach to protect the data confidentiality is to encrypt the data before outsourcing. However, this will cause a huge cost in terms of data usability. For example, the existing techniques on keyword-based information retrieval, which are widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally is obviously impractical. In order to address the above problem, researchers have designed some general-purpose solutions with fully-homomorphic encryption or oblivious RAMs. However, these methods are not practical due to their high computational overhead for both the cloud sever and user. On the contrary, more practical special purpose solutions, such as searchable encryption schemes have made specific contributions in terms of efficiency, functionality and security. Searchable encryption schemes enable the client to store the encrypted data to the cloud and execute keyword search over ciphertext domain. So far, abundant works have been proposed under different threat models to achieve various search functionality, such as single keyword search, similarity search, multi-keyword boolean search, ranked search, multi-keyword ranked search, etc. Among them, multikeyword ranked search achieves more and more attention for its practical applicability. Now some dynamic schemes have been proposed to support inserting and deleting operations on document collection. These are significant works as it is highly possible that the data owners need to update their data on the cloud server. But few of the dynamic schemes support efficient multikeyword ranked search.

This paper proposes a secure tree-based search scheme over the encrypted cloud data, which supports multikeyword ranked search and dynamic operation on the document collection. Specifically, the vector space model and the widely-used "term frequency (TF) × inverse document frequency (IDF)" model are combined in the index construction and query generation to provide multikeyword ranked search. In order to obtain high search efficiency, we construct a tree-based index structure and propose a "Greedy Depth-first Search" algorithm based on this index tree. Due to the special structure of our tree-based index, the proposed search scheme can flexibly achieve sub-linear search time and deal with the deletion and insertion of documents. The secure kNN algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors. To resist different attacks in different threat models, we construct two secure search schemes: the basic dynamic multi-keyword ranked search (BDMRS) scheme in the known cipher text model, and the enhanced dynamic multi-keyword ranked search (EDMRS) scheme in the known background model. Our contributions are summarized as follows:

1) We design a searchable encryption scheme that supports both the accurate multi-keyword ranked search and flexible dynamic operation on document collection.

2) Due to the special structure of our tree-based index, the search complexity of the proposed scheme is fundamentally kept to logarithmic. And in practice, the proposed scheme can achieve higher search efficiency by executing our "Greedy Depth-first Search" algorithm. Moreover, parallel search can be flexibly performed to further reduce the time cost of search process.

Characteristics and Services Models:

The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below:

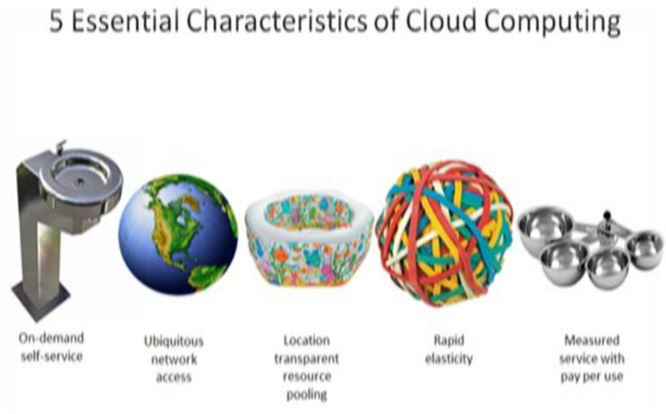
On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.

Broad network access: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).

Resource pooling: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.

Rapid elasticity: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

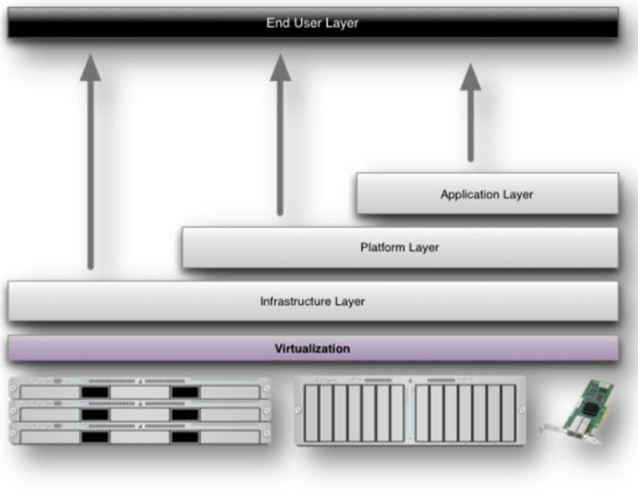
Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service.


5 Essential Characteristics of Cloud Computing
On-demand self-service
Ubiquitous network access
Location transparent resource pooling
Rapid elasticity
Measured service with pay per use

Characteristics of cloud computing

Services Models:

Cloud Computing comprises three different service models, namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The three service models or layer are completed by an end user layer that encapsulates the end user perspective on cloud services. The model is shown in figure below. If a cloud user accesses services on the infrastructure layer, for instance, she can run her own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications herself. If she accesses a service on the application layer, these tasks are normally taken care



Structure of service models

Benefits of cloud computing:

1. Achieve economies of scale – increase volume output or productivity with fewer people. Your cost per unit, project or product plummets.
2. Reduce spending on technology infrastructure. Maintain easy access to your information with minimal upfront spending. Pay as you go (weekly, quarterly or yearly), based on demand.
3. Globalize your workforce on the cheap. People worldwide can access the cloud, provided they have an Internet connection.
4. Streamline processes. Get more work done in less time with less people.
5. Reduce capital costs. There's no need to spend big money on hardware, software or licensing fees.
6. Improve accessibility. You have access anytime, anywhere, making your life so much easier!

7. Monitor projects more effectively. Stay within budget and ahead of completion cycle times.

8. Less personnel training is needed. It takes fewer people to do more work on a cloud, with a minimal learning curve on hardware and software issues.

9. Minimize licensing new software. Stretch and grow without the need to buy expensive software licenses or programs.

10. Improve flexibility. You can change direction without serious "people" or "financial" issues at stake.

EXISTING SYSTEM

- Searchable encryption schemes enable the clients to store the encrypted data to the cloud and execute keyword search over ciphertext domain.
- Due to different cryptography primitives, searchable encryption schemes can be constructed using public key based cryptography or symmetric key based cryptography.
- Symmetric searchable encryption (SSE) scheme, and the search time of their scheme is linear to the size of the data collection.
- Some early works have realized the ranked search using order-preserving techniques, but they are designed only for single keyword search.
- Privacy-preserving multi-keyword ranked search scheme, in which documents and queries are represented as vectors of dictionary size. With the "coordinate matching", the documents are ranked according to the number of matched query keywords.
- Privacy-preserving multi-keyword ranked search scheme does not consider the importance of the different keywords, and thus is not accurate enough. In addition, the search efficiency of the scheme is linear with the cardinality of document collection.
- A secure multi-keyword search scheme that supports similarity-based ranking. The authors constructed a searchable index tree based on vector space model and adopted cosine measure together with TF×IDF to provide ranking results.

DISADVANTAGES OF EXISTING SYSTEM:

- Firstly, all the users usually keep the same secure key for trapdoor generation in a symmetric SE scheme wherein revocation of the user is big challenge. In order to revoke a user, we need to rebuild the index and distribute the new secure keys to all the authorized users.
- Secondly, generally symmetric SE schemes assume that all the data users are trustworthy. But, a dishonest data user leads to many security issues.
- For each search request, users without pre-knowledge of the encrypted cloud data have to go through every retrieved file in order to find ones most matching their interest, which demands possibly large amount of post processing overhead.
- Invariably sending back all files solely based on presence/absence of the keyword further incurs large unnecessary network traffic, which is absolutely undesirable in today's pay-as-you-use cloud paradigm.
- Cannot apply fully homomorphic encryption schemes because of their excessive computational complexity.

PROPOSED SYSTEM:

- The proposed architecture does not require modifications to the cloud database, and it is immediately applicable to existing cloud DBaaS.
- For our system, we choose the B-tree as indexing data structure to identify the match between search query and data documents.
- Specially, we use inner data correspondence, i.e., the number of query keywords appearing in document, to evaluate the similarity of that document to the search query.
- We have used Microsoft's Azure platform to emulate the proposed system and to study its performance.

ADVANTAGES OF PROPOSED SYSTEM:

- The proposed architecture does not require modifications to the cloud database, and it is immediately applicable to existing cloud DBaaS, such as the experimented PostgreSQL Plus Cloud Database, Windows Azure.
- We provide not only multi-keyword query and accurate result ranking, but also dynamic update on document collections.
- Improved searching efficiency with privacy preserving.
- Parallel search process is also possible to reduce the time cost
- It guarantees data confidentiality by allowing a cloud database server to execute concurrent SQL operations (not only read/write, but also modifications to the database structure) over encrypted data.
- It provides the same availability, elasticity, and scalability of the original cloud DBaaS because it does not require any intermediate server.

METHODOLOGY

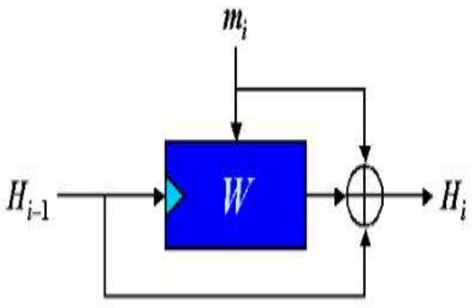
For our system, we choose the B-tree as indexing data structure to identify the match between search query and data documents. Specially, we use inner data correspondence, i.e., the number of query keywords appearing in document, to evaluate the similarity of that document to the search query. Whenever user wants to search, he/she creates a trapdoor for the keywords. Our aim is to design and analyse the performance of multiple keywords ranked search scheme using Commutative Further, we analysed its performance over B-tree based searchable index tree. We have used Microsoft's Azure platform to emulate the proposed system and to study its performance.

Origins of Whirlpool

• Created by Vincent Rijmen and Paulo S. L. M. Barreto
• Hashes messages of plaintext length 2^256 • Result is a 512 bit message
• Three versions have been released
– WHIRLPOOL-0
– WHIRLPOOL-T
– WHIRLPOOL

Structure of WHIRLPOOL

• Merkle-Damgård strengthening
• Miyaguchi-Preneelhashing scheme
• "W" is a 512-bit block cipher
• "m" is the plaintext, split into 512 bit blocks
• "H" is the blocks formed from the hashes



W Explained

• The block cipher W is the core element of the Whirlpool hash function
• It is comprised of 4 steps.
– Add Round Key
– Shift Columns
– Mix Rows
– Substitute bytes

Add Round Key

• During the Add Round Key step, the message is XOR'd with the key
• If this is the first message block being run through, the key is a block of all zeros
• If this is any block except the first, the key is the digest of the previous block

Shift Columns

• Starting from left to right, each column gets rotated vertically a number of bytes equal to which number column it is, from top to bottom
– Ex:
• [0,0][0,1][0,2]   [0,0][2,1][1,2]

- [1,0][1,1][1,2] ------> [1,0][0,1][2,2]
- [2,0][2,1][2,2] [2,0][1,1][0,2]

Mix Rows

- Each row gets shifted horizontally by the number of row it is. Similar to the shift column function, but rotated left to right

– Ex:

- [0,0][0,1][0,2] [0,0][0,1][0,2]
- [1,0][1,1][1,2] ------> [1,2][1,0][1,2]
- [2,0][2,1][2,2] [2,1][2,2][0,2]

Substitute bytes

- Each byte in the message is passed through a set of s-boxes
- The output of this is then set to be the key for the next round

B- Tree:

A B-tree is a data structure. The tree contains index nodes and leaf nodes. All leaf nodes are at the same level (same depth). Each index nodes contain keywords and pointers. Each node except root node in a B-tree with order n must contain keys between n to 2n keys. Each node also contains (number of keys + 1) pointers to its child nodes. If the root node is an index node then it must have at least 2 children. The insertion, deletion, search operations takes only logarithmic time. To design an efficient multi-keyword searchable encryption scheme(SSE) based on public key cryptography, we included the following modules.

Index Module:

Index structures for huge datasets cannot be stored in main memory. Disk is a possible alternative. Storing it on disk requires different approach. The solution is to use more branches to reduce the height of the tree. For this we used B-tree data structure for each document. B-tree is a data structure of order n. The nodes are filled from n to 2n keys. Nodes are always at least half full of keys. The keys are within each node. A list of pointers is inserted between keys. These pointers help to navigate through tree. In general, a node with k keys has (k+1) pointers.

Ranking Module:

In large databases, it is quite likely that the keyword might be matching with more number of documents. It is cumbersome for a user to decrypt and go through all the documents. Therefore there is a need for ranking the documents based on their relevance to the keywords. In our scheme we used (TF * IDF) to rank the documents. TF is the term frequency i.e. occurrence of keywords in a document and IDF is inverse document frequency i.e. total number of documents divided by number of documents containing the keyword. Similarity measure is used to find the rank based on relevance. For this, we maintain two vectors one for storing TF weight and other to store IDF weight.

Platform Used:

Microsoft Azure is a cloud service provider. It provides storage as a service to the customers. Azure architecture contains roles, i.e. the worker role and the web role . The web role is used for designing UI, whereas worker role is used to run background asynchronous applications. The workers in the B-tree provide search encryption services which support the multi-keyword search application. The encrypted index tree is created by tree builder function using encrypted keyword contents (worker A). Cloud users (web role) enter the keywords for search. The B-tree based tree search algorithm i.e. searches for the encrypted keywords in index tree. The search results are obtained using query on index tree and using tree search algorithm. Relevance score for ranking the search results is calculated using search algorithm, the index tree and database (worker B) as explained above in rank module.

RESULTS

The privacy-preserving multi-key generation over the encrypted cloud data has been designed. The system model presented has been developed on JAVA. The overall system has been developed and implemented with Microsoft Azure cloud platform. We build a special keyword balanced binary tree as the index. In addition, the search process may be performed in parallel to reduce the time, cost. The security of the system is protected against two threat models through secure top-k retrieval algorithm. In the existing system, all the users usually keep the same secure key for trapdoor generation in a symmetric SE scheme but in our proposed scheme dynamic keys are generated for trapdoor generation in a symmetric SE scheme to all the authorized users. In addition, the parallel search process is carried out to reduce the time cost in this proposed scheme. There are still many challenges problems in symmetric SE systems. In the proposed scheme the data owner is responsible for generating update information and sends it to the cloud server. The experimental results shows the effectiveness of our proposed scheme and are shown in below figures 3 and 4.

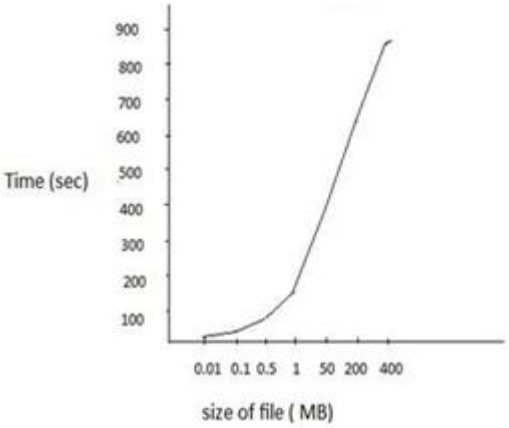


Fig. 3 Proposed System Using Parallel Search Process

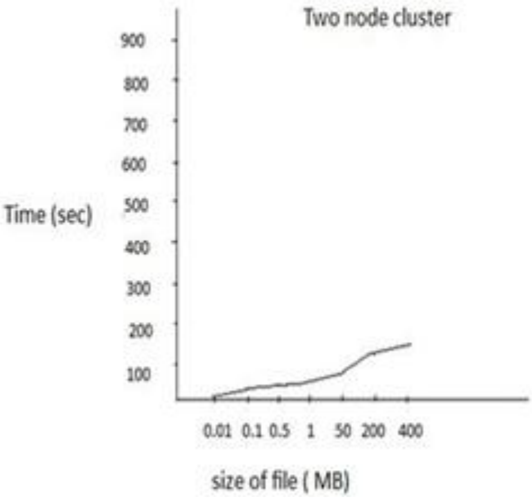


Fig. 4 Existing System without Using Parallel Search Process

CONCLUSION

In this paper, multi-keyword ranked search scheme with dynamic deletion and insertion of documents is proposed. By which privacy-preserving is possible and at the same time it is efficient. A special keyword balanced binary tree is constructed and a “Greedy Depth-first Search” algorithm is used to obtain better efficiency than linear search. The kNN algorithm is used to provide security against two threat models. Firstly, all the users usually keep the same secure key for trapdoor generation in a symmetric SE scheme wherein revocation of the user is big challenge. In order to revoke a user, we need to rebuild the index and distribute the new secure keys to all the authorized users. Secondly, generally symmetric SE schemes assume that all the data users are trustworthy. But, a dishonest data user leads to many security issues.

For example, a dishonest data user may search the documents and distribute the decrypted documents to the unauthorized ones. Also, a dishonest data user may distribute his/her secure keys to the unauthorized ones. In the future works, we will try to improve the SE scheme to handle these challenge problems.

FUTURE WORK:

In the future work, we will try to improve the SE scheme to handle these challenges. There are still many challenges in symmetric SE schemes. In the proposed scheme, the data owner is responsible to update information and transfer them to the cloud server. Thus, the data owner needs to store the unencrypted index tree and the information that is necessary to recalculate the IDF values. Such an active data owner may not be very suitable for the cloud computing model. It is a meaningful yet difficult task to design a dynamic searchable encryption scheme where the update operation can be performed by server at the cloud only.

As well, our scheme mainly considers the challenge from the cloud server. Actually, there are many secure challenges in a multi-user scheme.

REFERENCES:

[1] Xia, Zhihua, et al. "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data." IEEE Transactions on Parallel and Distributed Systems 27.2 (2016): 340-352.

[2] Prasanna B.T, C.B. Akki, 'A Survey on Homomorphic and Searchable Encryption Security Algorithms for Cloud Computing,' Communicated to International Journal of Information Technology and Computer Science, November, 2014.

[3]Prasanna B.T, C.B. Akki, 'A Comparative Study of Homomorphic and Searchable Encryption Schemes for Cloud Computing,' Communicated to International Journal of Communication Networks and Distributed Systems, November, 2014.

[4] Prasanna B.T, C.B. Akki, 'A Survey on Challenges and Security Issues in Cloud,' Presented in conference presented in Conference on Evolutionary Trends in Information Technology, May 20-22 2011, at Visvesvaraya Technological University, Belgaum, Karnataka.

[5] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," Parallel and Distributed Systems, IEEE Transactions on, vol. 23, no. 8, pp. 1467–1479, 2012.

[6] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in IEEE INFOCOM, April 2011, pp. 829–837.

[7] S. Kamara and K. Lauter, "Cryptographic cloud storage," in Financial Cryptography and Data Security. Springer, 2010, pp. 136–149.

[8] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.

[9] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," Journal of the ACM (JACM), vol. 43, no.3, pp. 431–473, 1996.

[10] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, 'Public key encryption with keyword search,' in Proc. of EUROCRYPT, 2004.

[11] C. Wang et al., 'Secure Ranked Keyword Search Over Encrypted Cloud Data,' Proc. ICDCS '10, 2010

[12] Wenjun Lu; Varna, A.L.; Min Wu, 'Confidentiality-Preserving Image Search: A Comparative Study Between Homomorphic Encryption and Distance-Preserving Randomization,' Access, IEEE, vol.2, no., pp.125,141, 2014

[13] W. K. Wong, D. W. Cheung, B. Kao, and N. Mamoulis, 'Secure knn computation on encrypted databases,' in Proc. of SIGMOD, 2009.

[14] K. Ren, C. Wang, and Q. Wang, 'Security Challenges for the Public Cloud,' IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.

[15] Zhangjie Fu et al, 'Multikeyword Ranked Search Supporting Synonym Query over Encrypted Data in Cloud Computing', IEEE Conference, 2013.

[16] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, 'Searchable symmetric encryption: improved definitions and efficient constructions,' in ACM CCS, 2006.

[17] P. Naresh, K. Pavan kumar, and D. K. Shareef, 'Implementation of Secure Ranked Keyword Search by Using RSSE,' International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 2 Issue 3, March – 2013.

[18] S.Buyrukbilen and S.Bairas, 'Privacy preserving ranked search on public key encrypted data,' in Proc. IEEE International Conference on High Performance

Computing and Communications (HPCC), November 2013.

[19] B. H. Bloom, ‘Space/time trade-offs in hash coding with allowable errors,’ Communications of the ACM, vol. 13, no. 7, 1970, pp. 422– 426.

[20] C. Gentry and Z. Ramzan, ‘Single-database private information retrieval with constant communication rate,’ in ICALP, pp. 803– 815.2005.

[21] Sun, W., Wang, B., Cao, N., Li, M., Lou, W., Hou, Y.T., Li, H., ‘Privacy-preserving multi keyword text search in the cloud supporting similarity-based ranking,’ Proceedings of the 8th ACMSIGSAC symposium on Information, computer and communications security, ACM, pp. 71–82.2013.