

# A SURVEY ON SECURED SEARCHING TECHNIQUES FOR EFFECTIVE DATA UTILIZATION IN PUBLIC CLOUD

<sup>1</sup>M.SHYAMALA DEVI,<sup>2</sup>C.ARUN

<sup>1</sup>Assistant Professor, Department of CSE, R.M.D. Engineering College, Chennai

<sup>2</sup>Professor, Department of ECE, R.M.K College of Engineering and Technology, Chennai

E-mail: [shyamalapmr@gmail.com](mailto:shyamalapmr@gmail.com), [carunece@gmail.com](mailto:carunece@gmail.com)

## ABSTRACT

Recent technology advances have made the success of cloud computing by acquiring the infrastructure, platform and software from the third party vendor. The promising benefit of cloud computing is data service outsourcing, by which the data owners stores their data in the public data centers by economically saving their capital investment towards data management. Cloud Storage provide users with abundant storage space and make user friendly for immediate acquiring of data, which is the foundation of all kinds of cloud applications. However, data outsourcing in the commercial public cloud also raise the problem for unauthorized data access and the cloud storage does not make sense if the outsourced data is not effectively utilized. The practical challenge is on how to make effective data access in the public cloud storage aiming at improvement of various searching techniques for increasing the data utilization. In this paper, an attempt is made to survey various searching techniques towards effective data utilization in cloud storage and is discussed in detail.

**Keywords:** *Cloud Storage, Data Outsourcing, Effective Data Utilization (EDU), Cloud Server (CS)*

## 1. INTRODUCTION

Cloud computing can have many definition from [2,3,6] as "A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet." Public clouds are run by third party service providers and applications from different customers are likely to be mixed together on cloud's servers, storage systems, and networks. Private clouds are built for exclusive use of one client and can be built and managed by organization's own administrator. Hybrid clouds combine both public and private cloud models which may be used to handle planned workload or storage clouds. Cloud storage [6] provides users with storage space and make user friendly and acquire data timely. Figure 1 shows the cloud storage architecture.

The rest of the paper is organized as follows. In Section 2, the literature survey is made about the data utilization in cloud storage. In Section 3, the existing searching strategies are discussed and the limitations in Section 4. The Performance analysis of all searching strategies is discussed in Section 5 and the research issues in Section 6. The performance

analysis of and the cloud data utilization service architecture is discussed in Appendix A at the end of the paper. In Section 7, we conclude our survey with further research directions.

## 2. LITERATURE SURVEY

Qin Liu [22] states that making CSP to manage the data leads security and privacy issues. So we can use cryptographic methods to disclose key only to authorized users to protect data from CSP. Ming [29] states that there must be search authorization to users to reduce privacy exposure to user or CS. Cong [20] states searchable encryption increases post processing overhead and network traffic. Kui [1] analyzed that users have various typing behaviors for keywords and are known as typos, representation inconsistencies and typing habits. To build Privacy assured cloud storage, Ming [9] satisfies the functional and Privacy requirements. Wei [15] created k-gram based fuzzy keyword set for W of encrypted files C and Jaccard coefficient to calculate keyword similarity. Jianfeng [14] discuss that the keyword contains file sensitive information, so keyword privacy must be protected EDU. Chai [16] developed a verifiable keyword search scheme in which the CS needs to prove that the search result is correct and complete.

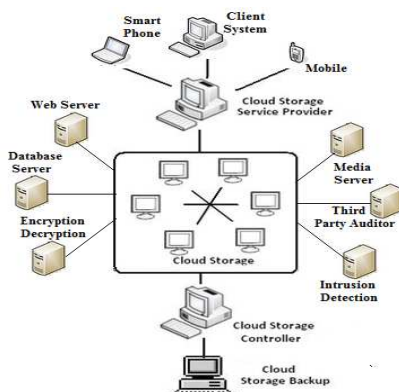


Figure 1 : Cloud Storage Architecture [6, 7]

Wenhai [21] states that EDU can be made by best search operation using Cosine measure. Jiadi [18] observes that server-side ranking based on order-preserving encryption leaks data privacy. He states that Ning [8], H.Hu [23] solve the problem of top-k multikeyword over encrypted cloud data but it suffers from Boolean representation. Shih Ting [23] discuss that PEKS and SCF-PEKS can be used in cloud storage to secure personal documents in cloud. H. S. Rhee [33] discusses the issues of SCF-PEKS and solves the problem for outside attacker for keyword. Yuanjie [26] states the issues of IND-PEKS and solves the problem for keyword outside attacker by analyzing frequency of occurrence of keyword trapdoor. Peng [19] found that third party could access files by knowing keyword search trapdoor. Ning [8] converse that Ranked search should protect the privacy by not revealing keyword anywhere by supporting multiple keyword searches [10, 11]. Wenhai [27] discuss that search result may have errors due to data corruption, software malfunction. Wenhai [31] discusses that owners have to make fine-grained search authorization to allow the users to search their data. Bing [32] point out that searching encrypted data involves not only information retrieval and also the search algorithms that run on the data structure.

The objective of our survey is to summarize the security and data utilization issues of various searching techniques in the encrypted cloud data.

### 3. KEYWORD SEARCHING STRATEGIES

#### 3.1 Secure and privacy preserving keyword search

Qin Liu [22] proposed this search that provides keyword privacy, data privacy and semantic secure by public key encryption. Here,

CSP is involved in partial decipherment by reducing the communication and computational overhead in decryption for users. The users submit the keyword trapdoor encrypted by users' private key to CS securely and retrieve the encrypted documents.

#### 3.2 Authorized Private keyword Search (APKS)

Ming Li [29] proposed APKS that provides keyword privacy, Index and Query Privacy, Fine-grained Search Authorization and Revocation, Multi-dimensional Keyword Search, Scalability and Efficiency. He proposed two methods APKS and APKS<sup>+</sup> using HPE. APKS method enhances the search efficiency using attribute hierarchy, and APKS<sup>+</sup> enhance the query privacy with the help of proxy servers that overcome dictionary attacks.

#### 3.3 Secure and Efficient Ranked Keyword Search

Cong Wang [20] proposed this search which solves processing overhead, data and keyword privacy, minimum communication and computation overhead. The owner build index along with the keyword frequency-based relevance scores for files. User request 'w' to CS with optional 'k' as  $T_w$  using the private key. The CS searches the index with scores and sends encrypted file based on ranked sequence.

#### 3.4 Secured fuzzy keyword Search

Kui Ren [1] proposed this search with symmetric searchable encryption (SSE). For keyword w, they form fuzzy keyword set  $T_w$  with edit distance 'd'. When user searches w to CS, it searches  $T_w$  and returns encrypted files matching  $T_w$ . They build the storage efficient fuzzy keyword set by using wild card and fuzzy searchable index by Multiway Tree which uses symbol-based trie – traverse searching.

#### 3.5 Privacy assured searchable cloud Storage

Ming Li [9] proposed this method using SSE, Scalar-Product-Preserving Encryption and Order-Preserving Symmetric Encryption that supports the privacy and functional requirements.

#### 3.6 K-gram based fuzzy keyword Ranked Search

Wei Zhou [15] proposed this search, in which owner create k-gram fuzzy keyword index I for files D and tuple  $\langle I, D \rangle$  is uploaded to search server(SS) which is inserted to bloom filter for size controlling. The encrypted file D is uploaded to

storage server. When user submits keyword  $K$ , the SS create  $k$ -gram fuzzy keyword set and calculate weight of each word in set and are searched with safe index  $I$ . Then the SS displays all files in sorted order matching the index based on rank.

### 3.7 Verifiable fuzzy keyword search (VFKS)

Jianfeng Wang [14] proposed this search using symbol-tree and index  $\tilde{G}_w$  has unique value "proof" for each node and the path is unique without key 'k'. When CS receives the keyword, it searches  $\tilde{G}_w$  and returns encrypted files. The verification is done by users by checking the proofset and IDset created from index.

### 3.8 Privacy-preserving Multi-keyword Text Search

Wenhai Sun [21] proposed this search that provides similarity based search result ranking, keyword privacy, Index and Query confidentiality and Query Unlinkability. The encrypted file is built by vector space model supporting conjunctive and disjunctive file search. The searchable index is built using Multidimensional B tree. Owner creates encrypted query vector  $\tilde{Q}$  for file keyword set. User gets encrypted query vector of  $W$  from owner which is given to CS. Now CS searches index by MD algorithm and compares cosine measure of file and query vector and returns top  $k$  encrypted files to user.

### 3.9 Secure Multikeyword Top-k Retrieval Search

Jiadi [18] proposed this search using Two-round searchable encryption (TRSE). In 1<sup>st</sup> round, users submits multiple keyword REQ  $W'$  as encrypted query for achieving data, keyword privacy and create trapdoor(REQ, PK) as  $T_w$  and sends to CS. Then CS calculates the scores from encrypted index for files and returns the encrypted score result vector to user. In 2<sup>nd</sup> round, user decrypts  $N$  with secret key and calculates the file ranking and then requests files with top- $k$  scores. The ranking of files is done on user side and scoring is done on server side.

### 3.10 Public-Key Encryption with Keyword Search (PEKS)

D. Boneh [24] proposed this search, in which CS contains encrypted files and keyword. User creates keyword trapdoor  $T_w$  using its private key to search  $W$ . The CS checks  $T_w$  with existing encrypted keyword and sends encrypted file that matches it. Since sender makes file encryption and

server makes the user authentication, there exists a secure channel between the sender, server and user.

### 3.11 Secure Channel Free (SCF-PEKS) Public-Key Encryption with Keyword Search

J. Baek [25] proposed this method, in which CS creates its own public and private key pair. Sender encrypts all files, keyword using servers' and users' public key before outsourcing [23]. User requests keyword trapdoor  $T_w$  to CS using its private key. CS checks the  $T_w$  using servers' private key and returns encrypted file.

### 3.12 Trapdoor Indistinguishability(IND-PEKS) Public-Key Encryption with Keyword Search

H. S. Rhee [33] proposed this search in which the outsourcing is done as SCF-PEKS. User requests  $T_w$  to CS encrypted with servers' public key and users' private key. CS checks  $T_w$  using servers' private key and returns encrypted file matching the keyword. Here the outsider cannot perform KGA without server's private key.

### 3.13 New Trapdoor Indistinguishability Public-Key Encryption with Keyword Search

Yuanjie[26] proposed this search, in which the outsourcing is same as SCF-PEKS. User creates two trapdoor for single keyword  $W$   $T_w = (T_{w1}, T_{w2})$  encrypted with servers' public key and users' private key which is sent to CS. Now CS checks the  $T_w$  using servers' private key and returns encrypted file. Outsider cannot perform KGA by analyzing the keyword occurrence frequency without server's private key and cannot distinguish two trapdoors.

### 3.14 Public-Key Encryption PEFKS with Fuzzy Keyword Search

Peng Xu [19] proposed this search, in which user creates fuzzy keyword trapdoor  $T_w$  and exact keyword trapdoor  $K_w$  for  $W$ . User requests  $T_w$  to CS. Then CS checks  $T_w$  with fuzzy keyword index and sends superset of matching cipher texts by FuzzTest algorithm that is executed by CS. The user process ExactTest algorithm for verifying ciphertexts with  $K_w$  and retrieve the encrypted files.

### 3.15 Privacy Preserving Multi Keyword Ranked Search (MRSE)

Ning [8, 12] proposed this search for known cipher text model and background model over encrypted data providing low computation and communication overhead. The coordinate matching is chosen for multi-keyword search. They used inner product similarity to quantitatively evaluate similarity for ranking files. The drawback is that

MRSE have small standard deviation  $\sigma$  which weakens keyword privacy.

### 3.16 Verifiable Privacy-Preserving Multi-keyword Text Search

Wenhai Sun [27] proposed Verifiable Privacy-Preserving Multi-keyword Text Search search that provides multi-keyword search by similarity search based result ranking. Owner outsources encrypted document  $\check{D}$  using vector space model and authenticated secure index tree built using Multidimensional B- tree encrypted using RSA and SHA-1 as [21]. User submits  $W$  to owner and receives encrypted query vector  $\check{Q}$  for  $W$ . The query  $\check{Q}$  along with search parameter  $k$  is given to CS. Now CS searches  $\check{Q}$  using MD algorithm and compares cosine measure of  $\check{D}$  and  $\check{Q}$  and returns top  $k$  encrypted files to user. Then user searches this minimum tree using the same search algorithm as CS and verifies the query results.

### 3.17 Attribute-based Keyword Search

Wenhai Sun [31] proposed Attribute-based Keyword Search search that provides conjunctive keyword search, keyword semantic security and Trapdoor unlinkability. The owners creates index with all keywords and access list with policy attributes which specifies the users list authorized for searching. Now owners encrypt the document, index with access list using cipher-text policy attribute based encryption. To have user membership management, they used proxy re-encryption [35] and lazy re-encryption techniques [36] to share the workload to CS. The user requests the  $T_w$  to CS using its private key. Now CS retrieves  $T_w$  and searches the encrypted indexes and return files only if the user's attributes in  $T_w$  satisfies access policies in indexes which makes coarse-grained dataset search authorization.

### 3.18 Privacy-Preserving Multi-Keyword Fuzzy Search

Bing Wang [32] proposed this Privacy-Preserving Multi-Keyword Fuzzy Search that provides keyword privacy, data privacy, Index privacy, user query privacy, multi keyword search and Trapdoor unlinkability. The encrypted file  $C$  and searchable Fuzzy keyword index  $I$  is outsourced to CS. The fuzzy keyword set is created by dividing the keyword into Bigram vector. It uses locality sensitive hashing to build the fuzzy searchable index vector and is inserted into bloom filter. The users query vector is also encrypted into query bloom filter. The server process the inner product of index vector and query vector and return the matched encrypted file.

## 4. LIMITATION OF EXISTING KEYWORD SEARCHING STRATEGIES

### 4.1. Secure and privacy preserving keyword search

- The communication and computational cost for encryption and decryption is more

### 4.2. Authorized Private keyword Search

- In practice not all the attributes are hierarchical.

### 4.3. Secure and Efficient Ranked Keyword Search

- It does not perform multiple keyword searches.
- Little overhead in index building.

### 4.4. Secured fuzzy keyword Search

- Does not support fuzzy search with public-key-based searchable encryption
- Could not perform multiple keywords semantic search
- The updates for fuzzy searchable index are not efficiently done.

### 4.5. Privacy assured searchable cloud Storage

- This scheme reveals the access pattern to the cloud server
- It does not hide the sum of multiple keywords scores from the cloud server, which leads to accessing the statistical information for re-identifying the search keywords.
- It does not support public key based searchable encryption

### 4.6. K-gram based fuzzy keyword Ranked Search

- The size of the k-gram based fuzzy keyword set depends on the jaccard coefficient value.

### 4.7. Verifiable fuzzy keyword search

- Verifiable fuzzy keyword search requires more space for storing the symbol tree fuzzy searchable index  $\check{G}_w$
- The updates for fuzzy searchable index is not efficient

### 4.8. Privacy-preserving Multi-keyword Text Search

- The similarity rank score of the document vector fully depends on the type of the document



**4.9. Secure Multikeyword Top-k Retrieval Search**

- Though the reduction and compression is used to reduce ciphertext size, the key size is still too large
- The communication overhead will be very high if the encrypted trapdoor size is too large.
- It does not make effective searchable index update

**4.10. Public-Key Encryption with Keyword Search**

- Creating a secure channel is more costly and inefficient
- The trapdoor have to be created for each keyword by the user
- It does not support search for multiple keyword
- Keywords may be hacked by Keyword Guessing Attack (KGA).

**4.11. Secure Channel Free Public-Key Encryption with Keyword Search**

- It suffers from outside attacker by KGA

**4.12. Trapdoor Indistinguishability(IND-PEKS) Public-Key Encryption with Keyword Search**

- It suffers from outside attacker using KGA and analyzing the frequency of occurrence of keyword trapdoor

**4.13. New Trapdoor Indistinguishability Public-Key Encryption with Keyword Search**

- Creating two trapdoors for single keyword and its maintenance is complicated

**4.14. Public-Key Encryption with Fuzzy Keyword Search**

- The process of creating fuzzy keyword index and exact keyword index is too difficult if the database is very large

**4.15. Privacy Preserving Multi Keyword Ranked Search**

- Multi-keyword ranked search (MRSE) for known cipher text model may produce two different trapdoor which vague the privacy leakage problem of trapdoor unlinkability which may weaken the keyword privacy.
- MRSE have small standard deviation  $\sigma$  which in turn weaken the keyword privacy.
- The integrity of the rank order is not checked in MRSE.

**4.16. Verifiable Privacy-Preserving Multi-keyword Text Search**

- Verifiable Privacy-Preserving Multi-keyword Text Search does not support typing inconsistencies

**4.17. Attribute-based Keyword Search**

- Trapdoor generation will need more time with the increased number of attributes

**4.18. Privacy-Preserving Multi-Keyword Fuzzy Search**

- Users need to add documents with relatively high score into the result to reduce the false negative rate.

**5. PERFORMANCE ANALYSIS**

The performance analysis for all searching methods is given in appendix. Qin Liu [22] implemented Secure and privacy preserving keyword search by comparing it with PEKS in encryption and decryption. Ming [29] implemented APKS and APKS<sup>+</sup> using Pairing-Based Cryptography Library on Linux with 32-bit, 3.4GHz Pentium D CPU. Cong [20] implemented Secure and Efficient Ranked Keyword Search in C language on linux machine with dual Intel Xeon CPU running at 3.0 GHz for Request for comments (RFC) DB. Kui [1] implemented C code on Amazon EC2 cloud with RFC DB. Wei Zhou [15] simulates for building k-gram fuzzy keyword set for search request and time to build query request is growing linearly as the number of query words increases. Jianfeng [14] implemented C code on LINUX with Intel Pentium E5800 3.2GHz and 2G memory for constructing symbol tree fuzzy searchable index  $\tilde{G}_w$ . Wenhai [21] implemented Privacy-preserving Multi-keyword Text Search using JAVA on Linux Server with Intel i3 Processor 3.3GHz and used ten years' IEEE INFOCOM publications and evaluated by randomly selecting keywords. Jiadi [18] implemented C code on Windows 7 machine with Core 2 Duo CPU at 2.0 GHz for user and server uses C code on Linux with Xeon E5620 CPU at 2.4 GHz. They implement TRSE scheme on file set of National Science Foundation Research Awards. Peng [19] implemented PEFKS from bilinear pairings using bilinear map generator and gave analysis for PEKS and PEFKS. Ning [8] implemented C code on Linux with Xeon Processor 2.93 GHz for Multi-keyword ranked search on Enron Email Data Set [13]. Wenhai [27] implemented verifiable Privacy-Preserving Multi-Keyword Text Search using JAVA on Linux Server with Intel Core i3 Processor

3.3GHz with ten years' IEEE INFOCOM publications with 3600 publications by extracting 9000 keywords. Wenhai [31] developed Attribute-based Keyword Search using C on Linux with Intel Core i3 Processor 3.3GHz with Enron Email Dataset. Bing [32] implemented Privacy-Preserving Multi-Keyword Fuzzy Search on PC with Intel Core i3 processor at 3.3 GHz and 4 Gb RAM, with 10 years' IEEE INFOCOM publication.

## 6. RESEARCH ISSUES

The primary concern of moving the private workload to cloud is for protecting data and secured data utilization. So the searching techniques can be improved through fuzzy search with owner controlled cloud data, privacy assured, high data sharing and supporting data dynamics

## 7. CONCLUSION

This paper summarizes various searching techniques in the encrypted cloud data. We have done a systematic study on the security and data utilization issues in the cloud storage for all the available searching techniques. We have identified the main issues that are to be satisfied for secured data utilization are keyword privacy, Data privacy, Index privacy, Query Privacy, Fine-grained Search, Scalability, Efficiency, Result ranking, Index confidentiality, Query confidentiality, Query Unlinkability, semantic security and Trapdoor Unlinkability. Some of the searching techniques mainly focus on security and some on data utilization. The limitations of all the searching techniques are discussed as well. By the above survey, security can be provided by Public-Key Encryption and effective data utilization by fuzzy keyword search. We believe that this survey will make the researchers to shape their problem in the area of data utilization in cloud storage.

## REFERENCES:

- [1] Kui Ren et al., "Towards Secure And Effective Data utilization in Public Cloud", IEEE Transactions on Network, volume 26, Issue 6, November / December 2012
- [2] J. Geelan. "Twenty one experts define cloud computing," Virtualization, August 2008. Electronic Mag., article available at <http://virtualization.sys-con.com/node/612375>.
- [3] I. Foster et al., "Cloud computing and grid computing 360-degree compared," Grid Computing Environments Workshop, 2008. GCE'08, 2009, pp. 1-10.
- [4] R. Curtmola et al., "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," ACM CCS, 2006, pp. 79-88.
- [5] J. Li et al., "Fuzzy Keyword Search Over Encrypted Data in Cloud Computing," Proc. IEEE INFOCOM, 2010, pp. 441-45.
- [6] Libor Sarga, "Cloud Computing: An Overview", journal of systems integration 2012
- [7] <http://searchsmbstorage.techtarget.com/feature/Understanding-cloudstorage-services-A-guide-for-beginners>
- [8] Ning Cao et al., "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data", IEEE Transactions on parallel and distributed systems, vol. 25, no. 1, jan 2014
- [9] Ming Li et al., "Toward Privacy-Assured and Searchable Cloud Data Storage Services", IEEE Transactions on Network, volume 27, Issue 4, July/August 2013
- [10] A. Singhal, "Modern Information Retrieval: A Brief Overview," IEEE Data Eng. Bull., vol. 24, no. 4, pp. 35-43, Mar. 2001.
- [11] D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," Proc. IEEE Symp. Security and Privacy, 2000
- [12] S. Yu et al., "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing," Proc. IEEE INFOCOM, 2010.
- [13] W.W.Cohen, "Enron Email Data Set," <http://www.cs.cmu.edu/~enron/>, 2013
- [14] Jianfeng Wang et al., "Efficient Verifiable Fuzzy Keyword Search over Encrypted Data in Cloud Computing" , Journal of Computer Science and Information system, volume 10, Issue 2, April 2013
- [15] Wei Zhou et al., "K-Gram Based Fuzzy Keyword Search over Encrypted Cloud Computing "Journal of Software Engineering and Applications, Scientific Research , Issue 6, Volume 29-32, January 2013
- [16] Chai.Q et al., "Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers," <http://www.cacr.math.uwaterloo.ca/techreports/2011/cacr2011-22.pdf>
- [17] T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In M. Matsui, editor, Advances in Cryptology - ASIACRYPT 2009, volume 5912 of LNCS, pages 214-231. 2009.
- [18] Jiadi Yu, Peng Lu, Yanmin Zhu, Guangtao Xue, Member, IEEE Computer Society, and Minglu Li, "Toward Secure Multikeyword Top-k Retrieval over Encrypted Cloud Data", IEEE

- Transactions on dependable and secure computing, vol. 10, no. 4, July/August 2013
- [19] Peng Xu et al., "Public-Key Encryption with Fuzzy Keyword Search: A Provably Secure Scheme under Keyword Guessing Attack", IEEE Transactions on computers, vol. 62, no. 11, November 2013
- [20] Cong Wang et al., "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data", IEEE Transactions on parallel and distributed systems, vol. 23, no. 8, August 2012
- [21] Wenhai Sun et al., "Privacy-Preserving Multi-keyword Text Search in the Cloud Supporting Similarity-based Ranking", *the 8th ACM Symposium on Information, Computer and Communications Security*, Hangzhou, China, May 2013.
- [22] Qin Liuy, Guojun Wangyz, and Jie Wuz, "Secure and privacy preserving keyword searching for cloud storage services", ELSEVIER Journal of Network and computer Applications, March 2011
- [23] Shih-Ting Hsu et al., "A Study of Public Key Encryption with Keyword Search", International Journal of Network Security, Vol.15, No.2, PP.71-79, Mar. 2013
- [24] D. Boneh et al., "Public key encryption with keyword search," in *Advances in Cryptology – EUROCRYPT 2004*, Lecture Notes in Computer Science, vol. 3027, pp. 506-522, Interlaken, Switzerland, 2004. Springer
- [25] J. Baek et al., "Public key encryption with keyword search revisited", in *ICCSA 2008*, vol. 5072 of Lecture Notes in Computer Science, pp. 1249 - 1259, Perugia, Italy, 2008. Springer Berlin/Heidelberg.
- [26] Yuanjie Zhao et al., "A New Trapdoor-indistinguishable Public Key Encryption with Keyword Search, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 3 (1/2), 72-81, 2012.
- [27] Wenhai Sun et al., "Verifiable Privacy-Preserving Multi-keyword Text Search in the Cloud Supporting Similarity-based Ranking", Accepted for IEEE Transactions on Parallel and Distributed Systems (TPDS)
- [28] <http://aws.amazon.com/ec2>, 2008, "Amazon Elastic Compute Cloud (Amazon EC2)"
- [29] Ming Li et al., "Authorized Private Keyword Search over Encrypted Data in Cloud Computing, IEEE proc. international conference on distributed computing systems, June 2011, pages 383-392
- [30] Boyang Wang et al., "Privacy-Preserving Multi-Dimensional Range Search over Encrypted Cloud Data within Sublinear Time", *Technical report*, 2013.
- [31] Wenhai Sun et al., "Protecting Your Right: Attribute-based Keyword Search with Fine-grained Owner-enforced Search Authorization in the Cloud", IEEE INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014
- [32] Bing Wang et al., "Privacy-Preserving Multi-Keyword Fuzzy Search over Encrypted Data in the Cloud", Accepted for IEEE INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014
- [33] H. S. Rhee et al., "Trapdoor security in a searchable public-key encryption scheme with a designated tester," *The Journal of Systems and Software*, vol. 83, no. 5, pp. 763-771, 2010.

#### AUTHOR PROFILES:



**M. Shyamala Devi** received B.E (CSE) in 2005, M.E (CSE) in 2009 from P.S.N.A CET, Dindigul and M.B.A from Madurai kamaraj University, Madurai in 2009. She is pursuing Ph.D from Anna University, Chennai. Currently, she is an Assistant Professor at

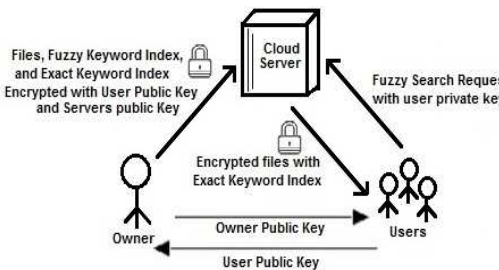
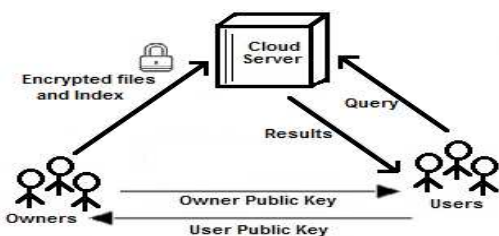
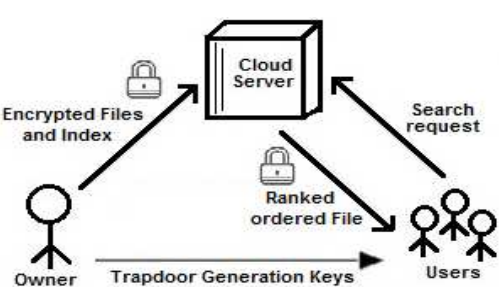
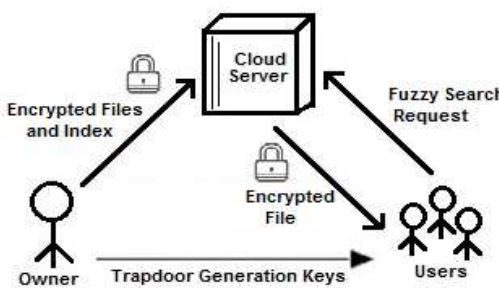
R.M.D Engineering College, Chennai. Her Research interests include Cloud Computing. She has authored 7 engineering books titled Theory of Computation, Principles of Compiler Design, Data Structures, Graphics and Multimedia, FOC Programming, Digital Computer Fundamentals and Visual Programming, by Shri Krishna HiTech Publishing Pvt Ltd, Chennai, TN, India



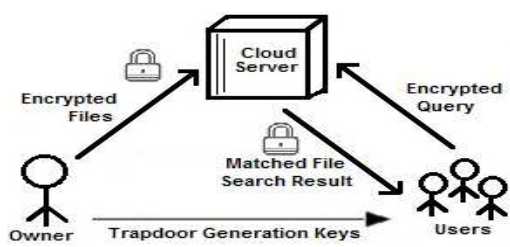
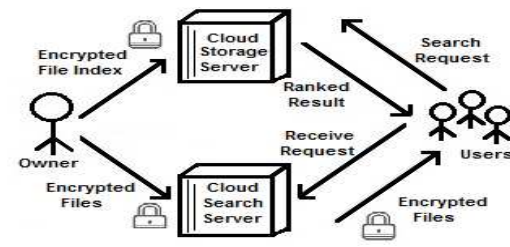
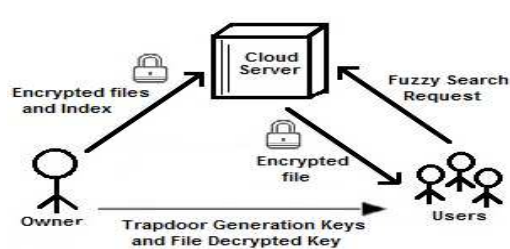
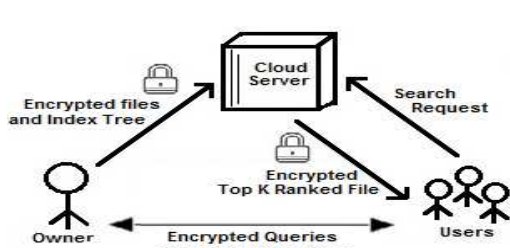
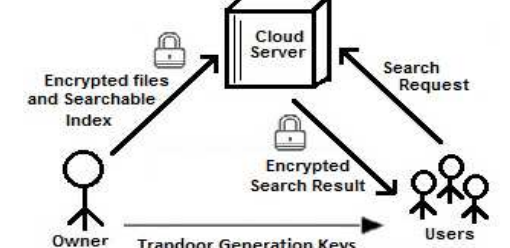
**Dr. C. Arun** received B.E in ECE from Bharathidasan University, Trichy, M.E Applied Electronics from Anna University, Chennai. He received Ph.D. degree from Anna University, Chennai in

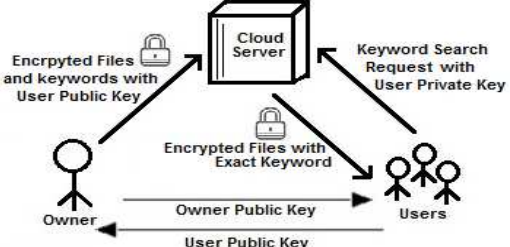
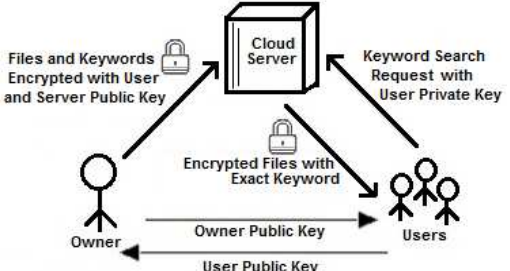
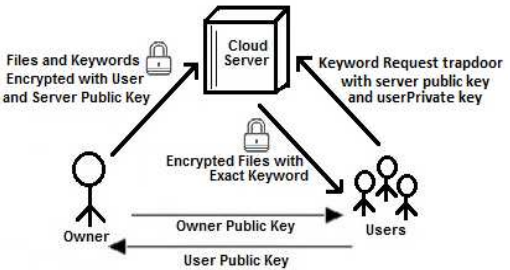
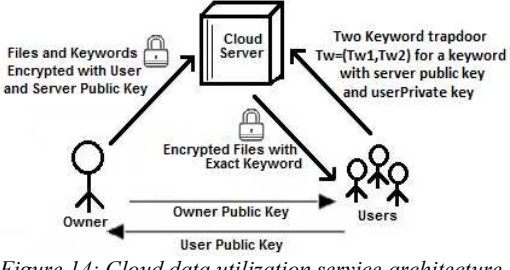
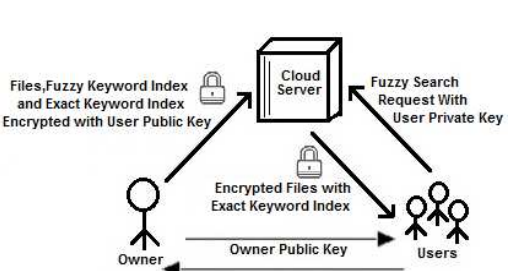
Information & Communication Engineering in 2009. Currently, he is a professor at R.M.K College of Engineering and Technology, Chennai. His research interests include VLSI Signal Processing, Image Processing, and Wireless Communication. He has received the prestigious Young Teacher Award and Er.A.Rajan Ambition achievement award from IET-UK

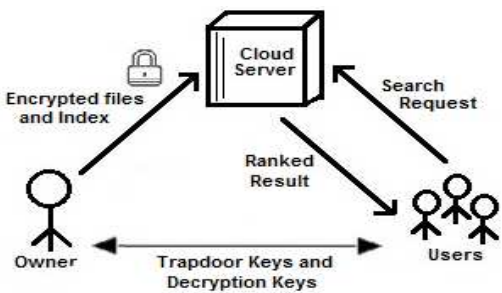
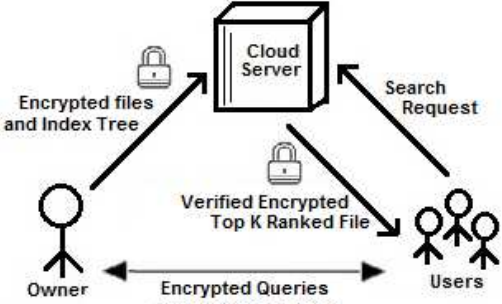
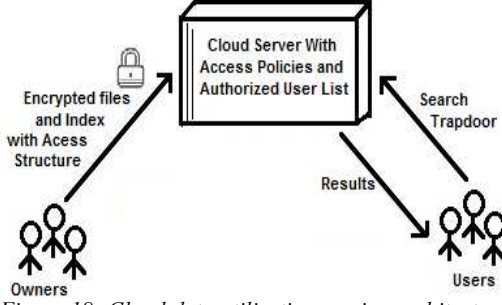
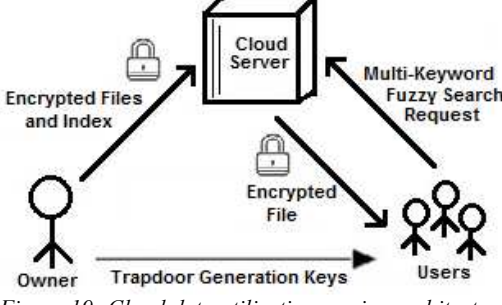
APPENDIX A – PERFORMANCE ANALYSIS OF SEARCHING TECHNIQUES

Search Method	Cloud Data utilization service Architecture	Performance complexity
<p>1. Secure and privacy preserving keyword search [22]</p>	 <p>Figure 2 : Cloud data utilization service architecture for Secure privacy preserving keyword search [22]</p>	<p><math>O(\text{Time}(A))</math></p>
<p>2. Authorized Private Keyword Search (APKS) [29]</p>	 <p>Figure 3: Cloud data utilization service architecture for Authorized Private Keyword Search [29]</p>	<p>Setup= <math>O(N^2)</math>                      Encryption=<math>O(N)</math>                      Search=<math>O(M \log N)</math>                      Where,                      N is total number of keywords and                      M is maximum size of the                      keyword set</p>
<p>3. Secure and Efficient Ranked Keyword Search[20]</p>	 <p>Figure 4: Cloud data utilization service architecture for Secure and Efficient Ranked Keyword Search[20]</p>	<p><math>O(\log M)</math> Where, M is domain score of keyword W</p>
<p>4. Secured fuzzy keyword search[1]</p>	 <p>Figure 5: Cloud data utilization service architecture [1] for Secured fuzzy keyword search</p>	<p>Fuzzy set cost - <math>O( W )</math>                      Storage cost - <math>O(MN)</math>                      Search cost <math>O(1)</math>                      Where W is keyword, N is the total number of keywords and M is the maximum size of the fuzzy keyword</p>



<p>5. Privacy assured searchable cloud storage [9]</p>	 <p>Figure 6: Cloud data utilization service architecture for Privacy assured searchable cloud storage [9]</p>	<p><math>O( W )</math> Where <math>W</math> is the keyword</p>
<p>6. K-gram based fuzzy keyword Ranked search [15]</p>	 <p>Figure 7: Cloud data utilization service architecture for k-gram based fuzzy keyword Ranked search [15]</p>	<p><math>O(N)</math> Where, <math>N</math> is total number of keywords.</p>
<p>7. Verifiable fuzzy keyword search (VFKS) [14]</p>	 <p>Figure 8: Cloud data utilization service architecture for verifiable fuzzy keyword search [14]</p>	<p>Storage cost - <math>O(MN)</math> Search cost - <math>O(1)</math> Verify cost - <math>O(1)</math> Where, <math>N</math> is the total number of keywords <math>M</math> is the maximum size of the fuzzy keyword</p>
<p>8. Privacy-preserving Multi-keyword Text Search [21]</p>	 <p>Figure 9: Cloud data utilization architecture for Privacy-preserving Multi-keyword Text Search [21]</p>	<p><math>O( W )</math> Where <math>W</math> is the number of keyword</p>
<p>9. Secure Multi keyword Top-k Retrieval search [18]</p>	 <p>Figure 10: Cloud data utilization service architecture for Secure Multikeyword Top-k Retrieval [18]</p>	<p>Setup = <math>O(\lambda)</math> Trapdoor = <math>O(l)</math> Score = <math>O(Nl)</math> Decryption = <math>O(N)</math> Where, <math>N</math> is the total number of keywords</p>

<p>10. Public-Key Encryption with Keyword Search [24]</p>	 <p>Figure 11: Cloud data utilization service architecture for PEKS[24]</p>	<p>Time cost of proxy = <math>O(N)</math>                  Communication cost = <math>O(2T)</math>                  Where,                  N is the total number of keywords                  T is the average number of keywords searchable ciphertext matching the query</p>
<p>11. Secure Channel Free Public-Key Encryption with Keyword Search (SCF-PEKS) [25]</p>	 <p>Figure 12: Cloud data utilization service architecture for SCF-PEKS [25]</p>	<p><math>O(2^N)</math>                  Where,                  N is the total number of keywords</p>
<p>12. Trapdoor Indistinguishability Public-Key Encryption with Keyword Search [33]</p>	 <p>Figure 13: Cloud data utilization service architecture for IND-PEKS [33]</p>	<p><math>O(2N)</math>                  Where N is the total number of keywords</p>
<p>13. New Trapdoor Indistinguishability Public-Key Encryption with Keyword Search [26]</p>	 <p>Figure 14: Cloud data utilization service architecture for New Trapdoor Indistinguishability PEKS [26]</p>	<p><math>O(3N)</math>                  Where,                  N is the total number of keywords                  Where <math>TW = (Tw1, Tw2)</math></p>
<p>14. Public-Key Encryption with Fuzzy Keyword Search [19]</p>	 <p>Figure 15: Cloud data utilization service architecture for PEFKS [19]</p>	<p>Time cost of proxy = <math>O(N)</math>                  Communication cost = <math>O(2T)</math>                  Time Cost of Receiver = <math>O(2T)</math>                  Where, N is the total number of keywords and                  T is the average number of keywords searchable ciphertext matching the query</p>

<p>15. Privacy-Preserving Multi-Keyword Ranked Search [8]</p>	 <p>Figure 16: Cloud data utilization service architecture for multi-keyword ranked search [8]</p>	<p>Index construction time:  <math>MRSE - I = O(mn^2)</math>  <math>MRSE - II = O(m(n+U)^2)</math>                  Trapdoor generation time:  <math>MRSE - I = O(n^2)</math>  <math>MRSE - II = O((n+U)^2)</math>                  Search time:  <math>MRSE - I = O(mn)</math>  <math>MRSE - II = O(m(n+U))</math></p>
<p>16. Verifiable Privacy-Preserving Multi-keyword Text [27] Search</p>	 <p>Figure 17: Cloud data utilization service architecture for Verifiable Privacy-Preserving Multi-keyword Text [27] Search</p>	<p><math>O( W )</math>                  Where W is the number of keyword</p>
<p>17. Attribute-based Keyword Search [31]</p>	 <p>Figure 18: Cloud data utilization service architecture for Attribute-based Keyword Search [31]</p>	<p>Setup = <math>O(N)</math>                  New user = <math>O(2N+1)</math>                  Building index = <math>O(N+1)</math>                  Generating Trapdoor = <math>O(2N+1)</math>                  Per-index Search = <math>O(N+1)P</math>                  Where,                  N is the number of attributes                  P is the number of keyword pairs</p>
<p>18. Privacy-Preserving Multi-Keyword Fuzzy Search [32]</p>	 <p>Figure 19: Cloud data utilization service architecture for Privacy-Preserving Multi-Keyword [32] Fuzzy Search</p>	<p>Search cost = <math>O(N)</math>                  Where,                  N is number of files in the data Set.</p>