International Journal of Mathematical Archive-6(11), 2015, 171-178
Available online through www.ijma.info ISSN 2229 – 5046

A SURVEY OF SEARCHABLE ENCRYPTION TECHNIQUES USED IN CLOUD ENVIRONMENT

N. JAYASHRI*[1], T. CHAKRAVARTHY[2]

**[1]Research Scholar, Department of Computer Science,
AVVM Sri Pushpam College, Thanjavur. Tamilnadu. India.**

**[2]Asso. Professor, Department of Computer Science,
AVVM Sri Pushpam College, Thanjavur. Tamilnadu. India.**

(Received On: 09-10-15; Revised & Accepted On: 25-11-15)

ABSTRACT

T*oday the world has the ability to utilize scalable, distributed computing environments within the confines of the internet, a practice known as cloud computing. Within the cloud computing world, the virtual environment lets out the user access computing power that exceeds all that contained within the physical world. Individuals and enterprise produce more and more data that must be stored and utilized. They are motivated to outsource their local complex data management system to the cloud owing to its greater flexibility and cost efficiency. Data encryption before outsourcing is the simplest way to protect data privacy, there are some drawbacks in the encryption such as plain text keywords without decrypting it, and these techniques support only conventional Boolean keyword search without capturing any relevance of the file retrieval accuracy is a significant drawback in searchable encryption schemes. Information retrieval (IR) community has already been utilizing various scoring mechanisms to quantify and rank order the relevance of files in response to any given search query. In order to achieve more efficient solutions, almost all the existing works on searchable encryption literature resort to the weakend security guarantee, that is revealing the access pattern and search pattern. Access pattern refers to the outcome of the search result, which files have been retrieved. The search pattern includes the quality pattern among two search requests.*

Keyword: *Cloud, Data Outsourcing, Security, Encryption, Keyword Search, Relevance Score, Information Retrieval.*

1. INTRODUCTION

Cloud is a breakthrough technology [1]. It provides convenient on-demand network access to a centralized pool of configurable computing resources [2, 3]. It enables an economic paradigm of data service outsourcing, where individuals and enterprise customers can avoid committing large capital outlays in the purchase and management of both software and hardware and the operational overhead therein[4]. Using cloud storage, users can remotely store their data and enjoy the on-demand high-quality applications and services from shared quality applications and services from a shared pool configurable computing resources, without the burden of local data storage and maintenance and it cuts across different sectors of economy, politics, and functions. It inevitably poses new security risks towards the correctness of the data in the cloud [4, 5].

Cloud services are categorized into: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [4-7]. Cloud deployment models as further classified into: Public Cloud- The cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services. Private Cloud-this cloud infrastructure is operated solely for a single organization. It may be managed by the organization or a third party, and may exist on-premises or off-premises. Community Cloud - The cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns. Hybrid Cloud- The cloud infrastructure is a composition of two or more clouds (private, community, or public) [5-9]. NIST model Cloud Architectures and Deployment models shown in the figure Fig.1.

Corresponding Author: N. Jayashri*[1]
***[1]Research Scholar, Department of Computer Science,
AVVM Sri Pushpam College, Thanjavur. Tamilnadu. India.***

Although cloud computing's benefits are tremendous, security and privacy concerns are the primary obstacles to wide adoption [10, 14]. Because cloud service providers (CSPs) are separate administrative entities, moving to the commercial public cloud deprives users of direct control over the systems that manage their data and applications. Even if CSPs' infrastructure and management capabilities are much more powerful and reliable than those of personal computing devices, the cloud platform still faces both internal and external security and privacy threats, including media failures, software bugs, malware, administrator errors and malicious insiders. Noteworthy outages and security breaches to cloud services appear from time to time [13].

Cloud customers and providers need to guard against data loss and theft. Cloud environments are shared with many tenants, and service providers have privileged access to the data in those environments [1]. Thus confidential data hosted in a cloud must be protected using a combination of access control, contractual liability and encryption [11]. As individuals and enterprises produce more and more data that must be stored and utilized (emails, personal health records, photo albums, tax documents, financial transactions, and so on), they're motivated to outsource their local complex data management systems to the cloud owing to its greater flexibility and cost-efficiency [4]. However, once users no longer physically retain their data, its privacy and reliability can be at peril [13].

For the former concern, data encryption before outsourcing is the simplest way to protect data privacy and combat unsolicited access in the cloud and beyond [1]. But encryption also makes deploying traditional data utilization services — such as plaintext keyword search over textual data or query over database — a difficult task [13,15]. The trivial solution of downloading all the data and decrypting it locally is clearly impractical, due to the huge bandwidth cost resulting from cloud scale systems [13, 16]. Moreover, aside from eliminating local storage management, storing data in the cloud serves no purpose unless people can easily search and utilize that data [13].

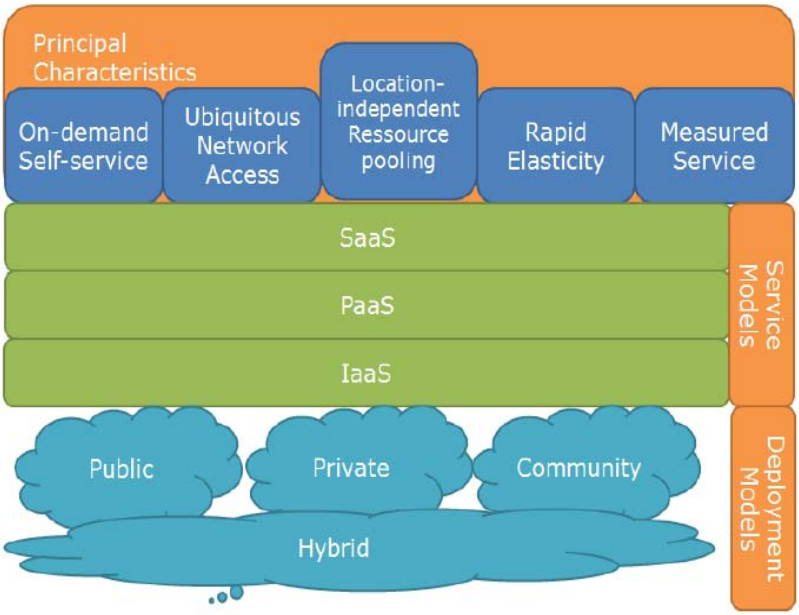

Fig.-1: NIST Cloud Architecture

This problem on how to search encrypted data has recently gained attention and led to the development of *searchable encryption* techniques. In this context, numerous interesting yet challenging problems remain, including similarity search over encrypted data, secure ranked search over encrypted data, secure multi keyword semantic search, secure range query, and even secure search over non textual data such as graph or numerical data.

2. INFORMATION RETRIEVAL

Author A. Singhal [25] discuss about the techniques that are used in modern information retrieval. Early Information Retrieval (IR) systems were Boolean systems which allowed users to specify their information need using a complex combination of Boolean ANDs, ORs and NOTs. Boolean systems have several shortcomings, e.g., there is no inherent notion of document ranking, and it is very hard for a user to form a good search request. Even though Boolean systems usually return matching documents in some order, e.g., ordered by date, or some other document feature, relevance ranking is often not critical in a Boolean system. The three most used models in IR research are the vector space model, the probabilistic models, and the inference network model.

The vector space model text is denoted by a vector of terms [16]. Terms are usually words and phrases. Every word in the vocabulary turns into an independent dimension in a very high dimensional vector space. Probabilistic ranking means ranking the documents by decreasing probability of their query relevance, while true probabilities do not exist in IR system. Probabilistic models *calculate* the probability of relevance of documents for a query. In an Inference network model document retrieval is designed as an inference process [17]. Most of the IR systems techniques can be implemented based on this model. Using inverted list is the implementation method for above mentioned models. Since all documents are indexed based on the terms they have, indexing is the method of producing, constructing and collecting document representation, and the resultant inverted files are known as inverted index.

Single words are used as the terms in most of the IR systems. Non-informative words or function words such as *the, in, of, a, etc* also known as *stop-words*, are commonly neglected. Conflating different forms of the same word to its core form, known as *stemming* in IR system, it is also used in many of the methods. Poor stemming causes a system failure. Stemming is perhaps more favourable for languages having many word modulations [18]. Multi-word phrases are also used as index terms in some systems. A phrases match in the document is believed more than single word match, because in reality phrases are more significant than single word.

Even though there has been some debate over the years, the two desired properties that have been accepted by the research community for measurement of search effectiveness are recall: the proportion of relevant documents retrieved by the system; and precision: the proportion of retrieved documents that are relevant [19]. It is well accepted that a good IR system should retrieve as many relevant documents as possible (i.e., have a high recall), and it should retrieve very few non-relevant documents. Techniques that tend to improve recall tend to hurt precision and vice-versa. One measure that deserves special mention is average precision, a single valued measure most commonly used by the IR research community to evaluate ranked retrieval. Average precision is computed by measuring precision at different recall points and averaging [20]. The most critical piece of information needed for document ranking in all models is a term's weight in a document. Another technique that has been shown to be effective in improving document ranking is query modification via relevance feedback. A state-of-the-art ranking system uses an effective weighting scheme in combination with a good query expansion technique. Three main factors come into play in the final term weight formulation. a) Term Frequency. b) Document Frequency. c) Document Length.

In the early years of IR, researchers realized that it was quite hard for users to formulate effective search requests. It was thought that adding synonyms of query words to the query should improve search effectiveness. Researchers developed techniques to automatically generate thesauri for use in query modification. Most of the automatic methods are based on analyzing word co occurrence in the documents. Relevance feedback [21] is motivated by the fact that it is easy for users to judge some documents as relevant or non-relevant for their query. *Pseudo-feedback*, a variant of relevance feedback [22]. In pseudo-feedback the IR system assumes that the top few documents retrieved for the initial user query are "relevant", and does relevance feedback to generate a new query. Pseudo feedback has been shown to be a very effective technique, especially for short user queries. Cluster hypothesis states that documents that cluster together will have a similar relevance profile for a given query [23]. Natural Language Processing (NLP) has also been proposed as a tool to enhance retrieval effectiveness, but has had very limited success [24].

The field of information retrieval has come a long way in the last forty years, and has enabled easier and faster information discovery. In the early years there were many doubts raised regarding the simple statistical techniques. With exponential growth in the amount of information available, information retrieval will play an increasingly important role in future.

3. SEARCHING METHODOLOGIES

3.1 Symmetric Key Encryption using Sequential Scan Method.

D. Song, D. Wagner, and A. Perrig 2000 [15] describe cryptographic schemes for the problem of searching on encrypted data and provide proofs of security for the resulting crypto systems. they provide query isolation for searches, meaning that the untrusted server cannot learn anything more about the plaintext than the search result; they provide controlled searching, so that the untrusted server cannot search for an arbitrary word without the user's authorization; they also support hidden queries, so that the user may ask the untrusted server to search for a secret word without revealing the word to the server. These schemes all take the form of probabilistic searching: a search for the word returns all the positions where _ occurs in the plaintext, as well as possibly some other erroneous positions. It may control the number of errors by adjusting a parameter in the encryption algorithm. The user will be able to eliminate all the false matches (by decrypting), so in remote searching applications, false matches should not be a problem so long as they are not so common that they overwhelm the communication channel between the user and the server.

3.2. Conjunctive Keyword Search over Symmetrically Encrypted Data.

L. Ballard, S. Kamara, and F. Monrose, - 2005[31] - present two provably secure and efficient schemes for performing conjunctive keyword searches over symmetrically encrypted data. First scheme is based on Shamir Secret Sharing and provides the most efficient search technique in this context to date. Although the size of its trapdoors is linear in the number of documents being searched. Alternative based on bilinear pairings that yields constant size trapdoors. Latter construction is not only asymptotically more efficient than previous secure conjunctive keyword search schemes in the symmetric setting, but incurs significantly less storage overhead. L. Ballard *et.al* [31] - work in the setting where each document is associated with a list of keywords. In particular, they make the following assumptions: (*i*) the number of keywords associated with a document remains fixed and (*ii*) no keyword appears at two different locations in a list. The first constraint can be satisfied by simply adding null keywords to the list, while the second can be satisfied by prepending each keyword with a field name or the value of a counter. To reduce computational burden, trapdoors specify which positions should be searched within an index.

SCKS-SS. It is based on Shamir's threshold secret sharing scheme, and is provably secure in the standard model. SCKS-SS is semantically secure against chosen keyword Attacks. second construction, SCKS-XDH, achieves constant transmission overhead at the cost of placing a larger computational burden on the server. The security of the scheme is based on a new variant of the External Diffie-Hellman (XDH) assumption. L. Ballard et.al. [31] note that SCKS-SS is the most computationally efficient construction for index generation and searching. L. Ballard *et.al* [31] also mention that SCKS-XDH incurs significantly less storage and transmission overhead as it need not store or send elements in integer groups. Additionally, SCKS-XDH is more efficient for BuildIndex as it requires only $m(n + 1)$ multiplications, is faster than PREVIOUS METHODS for trapdoor generation. SCKS-SS requires slightly less than 2 seconds to generate 10, 000 indexes with 10 keywords each, while SCKS-XDH requires 445 seconds.

3.3. Searchable Symmetric Encryption (SSE)

The research people R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky [26] consider the natural extension where an arbitrary group of parties other than the owner can submit search queries. They formally define SSE in this multi-user setting, and present an efficient construction. Recall that IND2-CKA does not guarantee the privacy of user queries and then highlight technical issues with the simulation-based definition. R. Curtmola *et.al* [26] address both these issues by proposing new game-based and simulation-based definitions that provide security for both indexes and trapdoors. All previous work on SSE falls within the non-adaptive setting. R. Curtmola *et.al* [26] presents two constructions which prove secure under our new definitions. First scheme is only secure in the non-adaptive setting, but is the most efficient SSE construction to date. In fact, it achieves searches in one communication round, requires an amount of work from the server that is linear in the number of documents that contain the keyword, requires constant storage on the client, and linear storage on the server. While the construction in also performs searches in one round, it can induce false positives, which is not the case for our construction.

Second construction is secured against an adaptive adversary, but at the price of requiring a higher communication overhead per query and more storage at the server. While our adaptive scheme is conceptually simple, note that constructing efficient and provably secure adaptive SSE schemes is a non-trivial task. R. Curtmola *et.al* [26] constructions can also handle updates to the document collection. Point out an optimization which lowers the communication complexity per query from linear to logarithmic in the number of updates. They formally define searchable encryption in the multi-user setting, and present an efficient construction that does not require authentication, thus achieving better performance than simply using access control mechanisms.

3.4. Keyword Search on Public Key Encryption

Public Key Encryption with Keyword Search, D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano [27] discuss about keyword search issues in asymmetric encryption. Boneh *et.al* [27] says that a PEKS is semantically secure against an adaptive chosen keyword Attack. They present a two constructions for public-key searchable encryption: (i) An efficient system based on a variant of the Decision Diffie-Hellman assumption and (ii) A limited system based on general trapdoor permutations, but less efficient. The non-interactive searchable encryption scheme (PEKS) above is semantically secure against a chosen keyword attack in the random oracle model assuming BDH is intractable. Second PEKS construction is based on general trapdoor permutations, assuming that the total number of keywords that the user wishes to search for is bounded by some polynomial function in the security parameter. Source in distinguishability can be attained from any trapdoor permutation family, where for a given security parameter all permutations in the family are defined over the same domain. Constructing a PEKS is related to Identity Based Encryption (IBE), though PEKS seems to be harder to construct. It showed that PEKS implies Identity Based Encryption, but the converse is currently an open problem [27].

3.5. Deterministic Encryption

In this work M.Bellare, A.Boldyreva and A.O'Neill [28] present as-strong-as-possible descriptions of privacy, and structures achieving them, for public-key encryption methods where the encryption algorithm is deterministic. Public-key encryption with keyword search (PEKS) is a solution that provably provides strong privacy but search takes time linear in the size of the database. The practical community indicates that they want a search on encrypted data to be as efficient as on unencrypted data. Deterministic encryption allows just this. The encrypted fields can be stored in the data structure, and one can find a target ciphertext in time logarithmic in the size of the database.

The basic idea is to associate a "tag" to a plaintext, which can be computed both by the client to form a particular query and by the server from a ciphertext that encrypts it, so that it can index the data appropriately in standard data structures and search according to the tags. In Encrypt-and-Hash ESE scheme they tag message with its hash. This scheme explains that the structure achieves security when min-entropy of the data is high enough to preclude a dictionary attack by the adversary against the scheme. In the case of min entropy of the data is not high, structure allows for bucketization. RSA-DOAEP as a stand-alone noticeably privacy **C**hosen **C**iphertext **A**ttack (CCA) insecure, when perfectly combined in an "encrypt-then-sign" fashion with a secure digital signature scheme. It achieves CCA Security in the natural "outsider security" model. This may come with no additional cost, for example in the database-security applications, which also requires authenticity.

3.6. Confidentiality-Preserving Rank-Ordered Search

A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A.L. Varna, S. He, M. Wu, and D.W. Oard [29] introduces a new framework for confidentiality preserving rank-ordered search and retrieval over large document collections. An emerging critical issue that must be addressed is how to protect data collections and indices through encryption, while providing efficient and effective search capabilities to authorized users. In addition to outsider attacks, security measures should also be taken against potential insider attacks. The requirements of balancing privacy and confidentiality with efficiency and accuracy pose significant challenges to the design of search schemes for a number of search scenarios. The goals of [29] are to explore a framework to securely rank-order documents in response to a query, and develop techniques to extract the most relevant document(s) from a large encrypted data collection.

To accomplish the goals, they collect term frequency information for each document in the collection to build indices, as in traditional retrieval systems for plaintext. Further secure these indices that would otherwise reveal important statistical information about the collection to protect against statistical attacks. During the search process, the query terms are encrypted to prevent the exposure of information to the data centre and other intruders, and to confine the searching entity to only make queries within an authorized scope. Utilizing term frequencies and other document information, apply cryptographic techniques such as order-preserving encryption to develop schemes that can securely compute relevance scores for each document, identify the most relevant documents, and reserve the right to screen and release the full content of relevant documents. The proposed framework has comparable performance to conventional searching systems designed for non-encrypted data in terms of search accuracy.

This searchable layer of encryption is referred as the *inner-layer encryption*, which is denoted by $TF(s)$. Inner-layer encryption can be done via cryptographic tools such as homomorphic encryption (HME) and order preserving encryption (OPE); the computation of relevance score should be adapted accordingly to support encrypted domain computation. They use OPE in [this] to demonstrate the concept for secure ranking of relevance. By introducing the order-preserving encryption on raw term frequency values, the OPE enables document search on the data centre side while preventing it from learning the critical term frequency information. When a query contains a single term, the OPE can achieve effective search as the baseline model by accurately identifying the target documents. The techniques introduced by [29] are first attempts to bring together advanced information retrieval capabilities and secure search capabilities. In addition to focus on securing indices, other important security issues include protecting communication links and combating traffic analysis. These will need to be addressed in future work.

3.7. Secure Ranked Search

This work was proposed by C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, in the year 2010[30]. Ranked search greatly enhances system usability by returning the matching files in a ranked order regarding to certain relevance criteria (e.g., keyword frequency), thus making one step closer towards practical deployment of privacy-preserving data hosting services in Cloud Computing. Inverted index is a widely used indexing structure that stores a list of mappings from keywords to the corresponding set of files that contain this keyword, allowing full text search. For ranked search purposes, the task of determining which files are most relevant is typically done by assigning a numerical score, which can be pre computed, to each file based on some ranking function a ranking function is used to calculate relevance scores of matching files to a given search request. The most widely used statistical measurement for evaluating relevance score in the information retrieval community uses the TF× IDF rule, where TF (term frequency) is simply the

number of times a given term or keyword appears within 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.

The above scheme clearly satisfies the security guarantee of SSE, i.e., only the access pattern and search pattern is leaked. However, the ranking is done on the user side, which may bring in huge computation and post processing overhead. To effectively support ranked search over encrypted file collection, now resort to the newly developed cryptographic primitive – order preserving symmetric encryption (OPSE) to achieve more practical performance. The OPSE is a deterministic encryption scheme where the numerical ordering of the plaintexts gets preserved by the encryption function. OPSE is then said to be secure if and only if an adversary has to perform a brute force search over all the possible combinations to break the encryption scheme. As the encrypted scores are order-preserved, server can execute the top-*k* retrieval about as fast as in the plaintext domain. Hence, the overall search time cost is almost as efficient as on unencrypted data.

3.8. Fuzzy Keyword Search

J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, [32] for the first time formalize and solve the problem of effective fuzzy keyword search over encrypted cloud data while maintaining keyword privacy. J. Li *et.al* [32] exploit edit distance to quantify keywords similarity and develop an advanced technique on constructing fuzzy keyword sets, which greatly reduces the storage and representation overheads. The fuzzy keyword search scheme returns the search results according to the following rules: 1) if the user's searching input exactly matches the pre-set keyword, the server is expected to return the files containing the keyword1; 2) if there exist typos and/or format inconsistencies in the searching input, the server will return the closest possible results based on pre-specified similarity semantics

There are several methods to quantitatively measure the string similarity. J. Li *et.al* [32] resort to the well-studied edit distance for their purpose. The edit distance ed(w_1, w_2) between two words w_1 and w_2 is the number of operations required to transform one of them into the other. The three primitive operations are i) Substitution, ii) Deletion and iii) Insertion. The key idea behind our secure fuzzy keyword search is two-fold: a) building up fuzzy keyword sets that incorporate not only the exact keywords but also the ones differing slightly due to minor typos, format inconsistencies, etc.; b) designing an efficient and secure searching approach for file retrieval based on the resulted fuzzy keyword sets. J. Li *et.al* [32] proposed to use a wildcard to denote edit operations at the same positions. The larger the pre-set edit distance, the more storage overhead can be reduced: with the same setting of the example in the straightforward approach, the proposed technique can help reduce the storage of the index from 30GB to approximately 40MB. J. Li *et.al* [32] explain the accuracy of the methods in terms of two characteristics, that is, wholeness and soundness.

3.9. Multi-Keyword Ranked Search Supporting Synonym Query

In recent years, an increasing number of researchers have engaged in the field of searchable encryption over encrypted cloud data. The existing searchable encryption schemes support only exact or fuzzy keyword search [22], not support semantics-based multi-keyword ranked search. In the real search scenario, it is quite common that cloud customers' searching input might be the synonyms of the predefined keywords, not the exact or fuzzy matching keywords due to the possible synonym substitution and/or her lack of exact knowledge about the data [23]. Therefore, synonym-based multi-keyword ranked search over encrypted cloud data remains a very challenging problem.

In this work, for the first time, Z.Fu, X.Sun, Z.Xia, L.Zhou, and J.Shu - 2013 [33] explain an effective approach to solve the problem of synonym-based multi-keyword ranked search over encrypted cloud data. Z.Fu. *et.al* (2013) make contributions mainly in two aspects: synonym-based search for supporting synonym query and multi-keyword ranked search for achieving more accurate search result. The sensitive frequency information can be well protected by introducing some dummy keywords, which is not adopted in basic scheme [33]. In this work Z.Fu.*et.al* [33] use Vector Space Model to build document index. To improve search efficiency, a tree-based index structure is used which is a balance binary tree. searchable index tree constructed with the document index vectors. So by traversing the tree we can find the related documents. Finally, Z.Fu *et.al* [33] analyze the performance of the schemes [33] in detail, including privacy, search efficiency, search accuracy, by the experiment on real-world dataset.

3.10. Privacy-Preserving Semantic Multi-Keyword Ranked Search

In this work [34], L.Chen, X.Sun, Z.Xiaand, and Q.Liu - 2014 propose a semantic multi-keyword ranked search scheme over the encrypted cloud data, which simultaneously meets a set of strict privacy requirements. Methodologies used in this work are listed, i) **Latent Semantic Search:** We aim to discover the latent semantic relationship between terms and documents. The proposed scheme tries to put similar items near each other in some space in order that it could return the data user the files contain the terms latent semantically associated with the query keyword. ii) **Multi-keyword Ranked Search:** It supports both multi-keyword query and support result ranking. iii) **Latent Semantic Analysis:** In information retrieval, latent semantic analysis is a solution for discovering the latent semantic relationship.

It adopts singular-value decomposition (SVD) to find the semantic structure between terms and documents. iv) **Secure *k-NN***: In this scheme [34], distances between database points to a query point are computed for finding the nearer neighbors to the query point. L.Chen *et. al* [36] adopts Gauss-Jordan to calculate the inverse matrix. The key producing time is determined by the range of the matrix. Also, the proposed method [34] that processed by *SVD* algorithm will burn up the time.

4. DISCUSSION

Different types of conventional and cloud based "searchable encryption" methods and "keyword searching" methods are examines in the above sections. In this section we are going to discuss about methodologies and algorithms are used in these works. And also some advantages of these methods are also listed here. Techniques used by D. Song *et.al* [15] have a number of essential advantages: i) Provably secure, ii) support query isolation and hidden search, iii) simple and fast and iv) This scheme almost not initiate any communication and space overhead. This scheme is also very flexible, and it can easily be enhanced to carry more advanced search queries. L. Ballard *et.al* [31] Secure Conjunctive Keyword Search (SCKS) using XDH is less efficient in terms of index generation and searching than SCKS-SS. R. Contrary to the natural use of searchable encryption, they [26] only guarantee security for users that perform all their searches at once. The disadvantage of the method proposed by D. Boneh. *et.al* [27] is that the public key size increases linearly with the entire dictionary size. If researchers enclose an upper-bound on the entire number of keyword trapdoors that the client will liberate to the email gateway it can do much better using cover-free families and can allow keyword dictionary to be of exponential size.

M.Bellare *et.al* [28] present a deterministic encryption method which is simply a family of injective trapdoor functions. Author also observe that bucketization can enhance privacy at the cost of additional processing by the recipient. The techniques presented by A. Swaminathan *et,al* [31] first endeavour to produce together advanced information retrieval capabilities and secure search capabilities. Through detailed security analysis shows the method proposed by C. Wang. *et.al* [30] is secure and privacy-preserving, while correctly realizing the aim of ranked keyword search. J. Li *et.al* [32] show that their proposed approach is secure and privacy-preserving, whereas they perfectly appreciate the goal of fuzzy keyword search. The results show Z.Fu.*et.al* [33] their proposed solution is resourceful and effective in managing synonym-based searching. A method proposed by L.Chen *et.al* [34] return not only the accurate matching files, but also the files with the terms latent semantically related to the query keyword.

5. CONCLUSION

Until now, research related to searching encrypted data on cloud has devoted significant attention to supporting technologies. This survey emphasizes the importance of advances in keyword searching for fully realizing the cloud paradigm. We provided evidence that an increasing effort is being made to adapt searching techniques to the development of cloud environments. We highlighted the need for further work because the focus so far has been on applying well-known searching techniques rather than developing new techniques specifically for cloud environments. Each single problem favours a certain searching method. We suggest that we are still in a situation in which a holistic approach has not been achieved. Being aware of the needs of each particular environment will allow us to identify different complementary techniques that can fulfil our needs when suitably assembled.

In this paper we discussed the problem of keyword searching and gave an overview of keyword based searching techniques. Although cloud have been predominantly used for storage purpose, they provide efficient operation.

REFERENCES

1. P.Hofmann(2010). Cloud Computing: "The Limits of Public Clouds for Business Applications." Published by IEEE Computer Society.
2. Mell P, Grance T (2012). "The NIST definition of cloud computing". NIST Spec Publication 800:145. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.
3. "Definition of Cloud computing." (2012). http://en.wikipedia.org/wiki/Cloud_Computing.
4. A.Huth and J.Cebula. "The Basics of Cloud Computing." US_CER. United States Computer Emergencyteam
5. W.Jansen. T.Grance "The NIST Definition of Cloud Computing," US Nat'l Inst. of Science and Technology, 2011; http://csrc.nist.gov/publications/nistpubs/800-144/SP800-144.pdf.
6. "The Future of Cloud Computing-Part 2: Current and next phases". Terrence Lillard http://www.embedded.com / design/embedded-internet-design/4227429/The-future-of-cloud-computing---Part-2--Current-and-next-phases
7. http://searchcloudcomputing.techtarget.com/definition
8. K.Kwang. R.Choo. "Cloud computing: Challenges and future Directions". Trends & Issues in crime and criminal justice No. 400 October 2010. Australian Institute of Criminology.
9. Kuyoro S. O., Ibikunle F. & Awodele O. " Cloud Computing Security Issues and Challenges". IJCN, Volume (3) : ,Issue (5) : 2011

10. L.Crusader. "Problems Faced by Cloud Computing." https://dl.packetstormsecurity.net/ papers/general/ Problems FacedbyCloudComputing.pdf.
11. H.Takabi and J.B.D Joshi, G.J. Ahn. "Cloud Computing. Security and Privacy Challenges in Cloud Computing Environments." Copublished by the IEEE Computer & Reliability Societies. November/December 2010
12. Lori M. Kaufman, B.Potter,."Can Public-Cloud Security Meet Its Unique Challenges?" July/august 2010. Copublished by the IEEE Computer and Reliability Societies.
13. G.Pallis. "View from the Cloud ". January/February 2012. Published by the IEEE Computer Society. IEEE Internet Computing.
14. "Security Guidance for Critical Areas of Focus in Cloud Computing," Cloud Security Alliance, Dec. 2009; https:// cloudsecurityalliance.org/csaguide.pdf.
15. D. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," Proc. IEEE Symp. Security and Privacy, 2000.
16. G.Salton, A.Wong, and C. S. Yang. "A Vector Space Model for Information Retrieval." Communications of the ACM, 18(11):613–620, November 1975.
17. H.Turtle. "Inference Networks for Document Retrieval". Department of Computer Science, University of Massachusetts, Amherst, MA 01003, 1990. Available as COINS Technical Report 90-92.
18. D.Hull. "Stemming algorithms - a Case Study for Detailed Evaluation". Journal of the American Society for Information Science, 47(1):70–84, 1996.
19. C. W. Cleverdon. "The Cranfield Tests on Index Language Devices". Aslib Proceedings, 19:173– 192, 1967.
20. G.Salton and M. J. McGill. "Introduction to Modern Information Retrieval ". McGraw Hill Book Co., New York, 1983.
21. J. J. Rocchio. "Relevance feedback in information retrieval ". In Gerard Salton, editor, The SMART Retrieval System—Experiments in Automatic Document Processing, pages 313–323, Englewood Cliffs, NJ, 1971. Prentice Hall, Inc.
22. C.Buckley, J.Allan, G.Salton, and A.Singhal. "Automatic query expansion using SMART". In Proceedings of the Third Text REtrieval Conference (TREC-3), pages 69–80. NIST Special Publication 500-225, April 1995.
23. A. Griffiths, H. C. Luckhurst, and P.Willett. "Using interdocument similarity in document retrieval systems". Journal of the American Society for Information Science, 37:3–11, 1986.
24. T. Strzalkowski, L. Guthrie, J. Karlgren, J. Leistensnider, F. Lin, J. Perez-Carballo, T.Straszheim, J.Wang, and J. Wilding. "Natural language information retrieval TREC- 5 Report ". In Proceedings of the Fifth Text REtrieval Conference (TREC-5), 1997.
25. A. Singhal, "Modern Information Retrieval: A Brief Overview", IEEE Data Eng. Bull., vol. 24, no. 4, pp. 35-43, 2001.
26. R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," Proc. ACM Conf. Computer and Comm. Security (CCS '06), 2006.
27. D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search," Proc. Int'l Conf. Advances in Cryptology (EUROCRYP '04), 2004.
28. M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and Efficiently Searchable Encryption," Proc. Ann. Int'l Cryptology Conf. Advances in Cryptology (Crypto '07), 2007.
29. 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.
30. C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure Ranked Keyword Search over Encrypted Cloud Data," Proc. IEEE 30th Int'l Conf. Distributed Computing Systems (ICDCS '10), 2010.
31. L. Ballard, S. Kamara, and F. Monrose, "Achieving Efficient Conjunctive Keyword Searches over Encrypted Data," Proc. Int'l Conf. Information and Comm. Security (ICICS '05), 2005.
32. J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy Keyword Search over Encrypted Data in Cloud Computing," Proc. IEEE INFOCOM '10, 2010.
33. Z.Fu, X.Sun, Z.Xia, L.Zhou, and J.Shu – "Multi-keyword Ranked Search Supporting Synonym Query over Encrypted Data in Cloud Computing" - 2013.
34. L.Chen, X.Sun, Z.Xiaand, and Q.Liu – "An Efficient and Privacy-Preserving Semantic Multi-Keyword Ranked Search over Encrypted Cloud Data" – 2014 - International Journal of Security and Its Applications Vol.8, No.2 (2014), pp.323-332.

Source of support: Nil, Conflict of interest: None Declared