SPER Publications and Solutions Pvt. Ltd.

Original Article

Researchable encryption in cloud databases: a survey

Alireza Froozani Fard[1]*, Mohammadreza Mollahoseini Ardakani[2]

[1]Computer Engineering Department, Maybod Branch, Islamic Azad University, Maybod, Iran. [2] Assistant Professor, Computer Engineering Department, Meybod Branch, Islamic Azad University, Meybod, Iran.

Correspondence: Alireza Froozani Fard, Computer Engineering Department, Maybod Branch, Islamic Azad University, Maybod, Iran. Email: froo_kia1390@yahoo.com

ABSTRACT

Cloud computing is among the most critical technologies supporting the reliability, scalability, ease of establishment, and cost-effectiveness features for business development. Although the cloud offers a diversity of services, despite its advantages, cloud computing still faces open and enduring challenges in ensuring the confidentiality, integrity, and availability of its sensitive data. Data stored in the cloud is readily prone to disclosure to hackers. The data is encrypted before outsourcing to avoid this. Encrypted data becomes hard to search in this event. This article aims to gather and analyze most of the encryption methods in the cloud database and finally summarize them in a table to utilize their high-level techniques, methods, benefits, and drawbacks.

Keywords: Cloud computing, data encryption, outsourcing, keyword, index

Introduction

Cloud computing is a concept that has been gathering considerable attention nowadays in the field of networking and computers. Adopting cloud computing, companies, or even individuals can outsource to a cloud server instead of storing or processing information with personal systems and servers. As such, they can pay only the price of renting the service to the cloud provider rather than paying a large amount of money to create the infrastructure they require locally, hence saving the corresponding costs. Besides, employing cloud computing, the company or person can access information everywhere and anytime with the Internet. That is, this innovative technology enables companies and people to substitute the Internet platform with their hardware and software and storage and processing devices. It is only needed that the system, which can be a mobile phone, a personal computer, or a mobile computer, be connected to the cloud account through the Internet to benefit its features [1].

Access this article online

Website: www.japer.in	E-ISSN: 2249-3379

How to cite this article: Alireza Froozani Fard, Mohammadreza Mollahoseini Ardakani. Researchable encryption in cloud databases: a survey. J Adv Pharm Edu Res 2020;10(S1):81-88.
Source of Support: Nil, Conflict of Interest: None declared.

Cloud computing, as one of the most significant computational models in current years, has lessened investment in IT as well as enhanced computational competence [2-4]. Through the pay-per-use model, this technique presents consumers with extensive access to computing resources, including data storage, memory, processing, and virtual machines [5-7]. The chief objective of this schema is to provide computing capabilities in the figure of a metered service [8-12]. Consequently, according to Fig. 1, the cloud providers provide different scalable resources in the form of software services as a (SaaS), platform as a service PaaS, and infrastructure as a service (IaaS). [13-16].

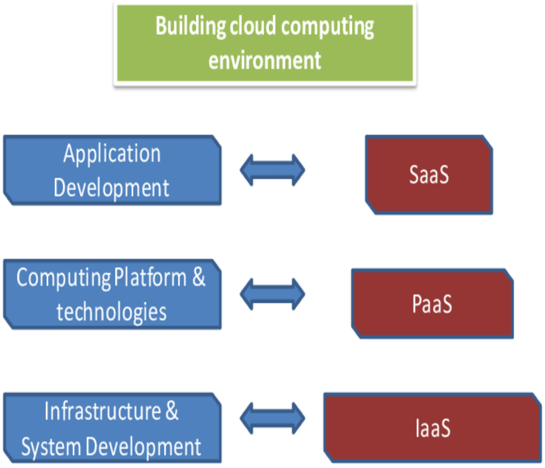
Building cloud computing environment
Application Development
SaaS
Computing Platform & technologies
PaaS
Infrastructure & System Development
IaaS

Fig. 1: Cloud computing structure [16]

With the arrival of cloud computing, more and more data is being conveyed to cloud servers by individual users and companies. Cloud services can regularly be divided into three types: Public cloud, private cloud, and hybrid cloud. Normally, the public cloud is not reliable. The private cloud is assumed semi-reliable or fully reliable, and the hybrid cloud is a compound of a public cloud and a private cloud [17,18].

Encryption preserves data security in the cloud. Yet, at the same time, the ability to search by the cloud service is hindered. Searching over encrypted data is uncommon in Encryption methods. In case the data is encrypted before being stored in the cloud, the primary notion of data for the cloud service will not be understandable. Because it doesn't have the key to decrypt the data. In new years, searching over encrypted data has been one of the most up-to-date subjects in encryption and security. One can attribute this to the significance of searching for data access, outsourcing bulk data to a cloud service, and finally, distrust of cloud services.

In this article, data outsourcing is introduced first in the second section. Then, data searchable encryption and the discussion of the search techniques on encrypted data are included in the third section. Lastly, the fourth section offers the conclusions of the examinations.

Data Outsourcing

In current years, numerous individuals, businesses, and companies have favored employing data outsourcing to benefit from attractive and diverse advantages of the cloud, using data storage and management services on cloud service providers. In the diverse designs of data outsourcing, several components have been viewed, depending on the intentions and services they present to the users. Most architectures comprise three entities: data owner, user, and cloud service. Fig. 2 depicts a modest design for data outsourcing [19, 20].

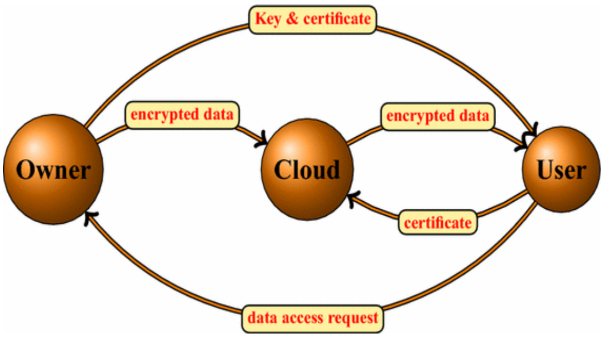

Fig. 2: Modest data outsourcing design [20]

Data owner

It is the actual data owner and supplier who negotiates with the cloud service and outsources its data to the cloud service to profit. It is supposed that the data owner does not have adequate experience or resources to preserve his data in local services and wants to outsource the data to answer the demands of users.

Cloud service

A cloud service is a semi-reliable professional entity that providing data storage and use services. In some cloud models, the two search and file services are available. In the design presented in Fig. 1, the index created by the data owner is given to the cloud service. The cloud service also conveys the results to the requestor. Secure outsourcing demands that the data owner and users be permitted to search the data, while the cloud service is unable to identify the data content.

Data user

A data user is a person who is verified by the data owner and has access to records stored in the cloud service. The user is a member of the system that sends search requests to the cloud service and gets a suitable response from them. In different models, user access for creating encrypted inquiries differs.

Nevertheless, the cloud server may suffer from many attacks from inside or outside, making it unreliable. Also, databases may include critical information, such as customer privacy and business secrets. For data owners, cloud storage increases concerns on the safety of outsourced critical data. Cloud service providers who store people's information may access the critical user information without consent. Accordingly, in case the concerns regarding keeping information security for data owners are not settled acceptably, they will be unwilling to outsource critical data, despite its numerous benefits [21]. Hence, before outsourcing from the cloud, the private database must be encrypted so that all critical data in the private dataset can be well preserved against the cloud. As such, the cloud cannot be informed about the content of the data it contains. Most traditional encryption systems concentrate on protecting against plaintext disclosure. But the execution of inquiry and analyze operation on the data encrypted in traditional methods is difficult. Therefore, in the current few years, the issue of searching over encrypted data, searchable encryption, has been a principal topic in data outsourcing research. Various models and frameworks have been introduced in this regard to satisfy security demands. Nevertheless, no model has hitherto been developed to thoroughly discuss the concerns in this field. Research in this domain and presentation of new models is still ongoing [22].

Searchable Data Encryption

The method of searching over encrypted data with data outsourcing was initially done by Sang et al in 2000. Searchable Encryption (SE) is known as an encryption method, which protects critical information versus the cloud service or any other illegal person in addition to providing data search conditions, including accuracy and effectiveness. The chief goal of this kind of encryption is to search over encrypted data in a fashion that, notwithstanding the search capability, the cloud service wouldn't be able to obtain information from system queries and data. What separates this method is the effectiveness, speed, timeliness, safety level of the data owner, and the search accuracy of the user [23, 24].

Search over encrypted data techniques

Each document can be split into a set of words, and each word can be considered as a token or symbol. This subdivision can be a 64-bit block, a word, a sentence, or a somewhat smaller unit depending on the program's domain of interest. For simplicity, in the beginning, these words were considered to have an identical length. As there could only be one low-bandwidth network for the server, only documents including the W-word were anticipated to be restored.

- ### Search with the sequential scan of the document

A- Initial search

The document including the sequence $w_1, w_2, \dots, w_i$ is originally encrypted. This is done by obtaining XOR of plaintext with Pseudo-random bit sequences similar to Fig. 3. The sequence of pseudo-random values $s_1, s_2, \dots s_i$ is created employing stream cipher, in which each S_i is of n-m bit length.

To encrypt the n-bit Wi word that appears in position i, the pseudo-random bits take the Si to adjust the set Ti: = $<S_i, F_{ki} (S_i)>$ and yield the Ci: = Wi ⊕ Ti encrypted text as the output. Note that it is only the key owner that can generate $T_1, T_2, \dots T_i$ pseudo-random stream. No one else can do it. Of course, encryption can be online: so that each word accessed is encrypted. The selection of k_i keys is flexible. That is, one can use the same k key for all places in the document, or a new k_i key for each position regardless of the other keys [23].

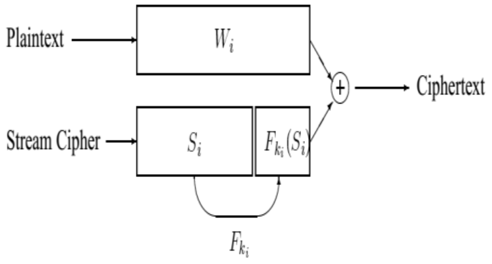

Fig. 3: The initial search

This arrangement enables searching the data without exposing anything else regarding the plaintext. To search the W-word, the corresponding W and k must be provided. The server can next search W in the encrypted text by examining whether Wi ⊕ Ci is for the similar S is of $<S, F_{ki} (S)>$ form. In circumstances where no key is granted, nothing can be discovered about the plaintext. Still, this design provides for limited control only and is not competent.

B) Controlled search

As stated before, if the user desires to search for the word W, $f_{k'}(W)$ and W must be presented to the server. This enables the server to know all the places where W may be. But ut doesn't show anything in places where $W_i \neq W$. The main idea is to choose the keys following $k_i{:} = f_{k'}(W_i)$. As such, k' is chosen randomly and is never disclosed; this is the desired goal. This design is incompetent yet: to permit the server to locate the word W, still, the owner has to give W to the server [23].

C) Hidden Search

It is not very pleasant for the data owner to expose the desired word to the server to search. The data owner desires the server to search and retrieve the answer without being aware of the word. This is achieved by applying a definite $E_{k''}$ encryption algorithm and encrypting each W word in the plaintext individually. Note that E is not authorized to use any random agents. Moreover, the calculation of $E_{k''}(x)$ relays only on x, and should not depend on the position i in the document where x is located. After the encryption, a succession of encrypted words $x_1, x_2, \dots x_l$ is made. Next, by stream cipher, the text is encrypted to achieve C_i (Fig. 4). Ci := Xi ⊕ Ti, $X_i{:} = E_{k''}(W_i)$, Ti := < Si , F_{ki}(Si).

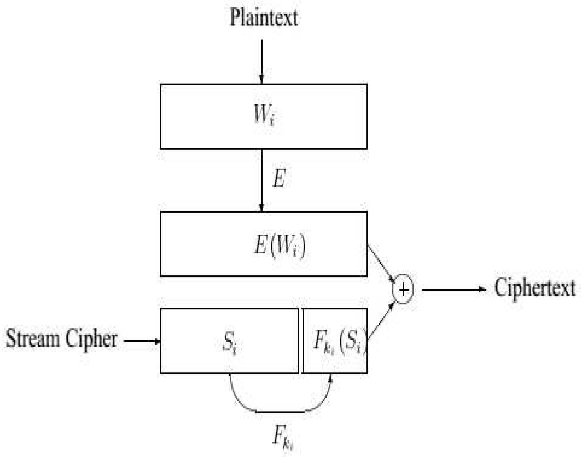

Fig. 4: Hidden search [23]

To search a W word, the owner calculates X: = $E_{k''}$(W) and k: = $f_{k'}$(X) and sends <X,K> to the server. This lets the server to search for W without disclosure. It is easily perceived that this design meets the hidden search feature as long as it is safe against the E-encryption. Given that the data owner provides the k_i keys following the $k_i{:}=f_{k'}(E_{k''}(W_i))$ formula, the plaintext can't be restored from the encrypted text anymore. Because he needs to recognize the $E_{k''}(W_i)$ before encrypting it. This goal defeats the encryption design, for even the experts can't decipher with access to the keys [23].

- ### Index-based search

Considering the assumption that some of the data are bulky, the successive scan may not be as efficient as adequate. For some applications, i.e. massive databases, a general method to speed up the search is to employ a pre-calculated index. This index includes a list of keywords. For each keyword, there is a list of pointers to the documents where the keywords emerge. Keywords are favorite words that may be searched later. This index is initially created for plaintext documents. The index and plaintext are then encrypted. The encrypted text is then stored in the server. It is simplistic to imagine that encryption of keywords is enough and the locations can be free and exposed in the index. Although this facilitates the search for the answer to searches, it also sends much information to the servers and paves the way of different statistical attacks. A simple method is to also encrypt document pointers in each index list. consequently, when E(W) is searched in the server and a match is found, the encrypted list returns those conforming locations through the index. A general drawback of this kind of search is that whenever the documents vary, the index is to be updated. This can lead to significant overhead [25, 26].

Here are some types of indexes:

A) Direct index

The direct index contains tuples of the document and keyword list. In this method, the (key; value) index (Table 1) is used to construct the index. **Key** represents a document, and the **value** is the list of keywords utilized in it. The index table is encrypted in multiple ways to retain it protected from the sight of the cloud service. The security gap of the direct index structure is the revelation of keyword distribution information over time [27,28].

Table 1: The direct index

Key	Value
D1	w1,w3,…w7
D2	w1,w5,…w11
….	….
D3	w2,w5,…w10

B) Inverse index

The reverse index was formed because of some direct index security flaws, such as exposure of keyword distribution information in documents. The same (key; value) structure is used in this index too; except that **key** indicates a keyword and **value** represents its corresponding documents [29].

Table 2: The Inverse index

Key	Value
w1	D1,D2,..D18
w2	D2,D7,..D25
….	….
wm	D4,D12,..Dn

C) Bitmap index

This index was proposed in 1985 by Wang [30]. Corresponding to each unique value per column, a bitmap is created in this index. The user can refer to the corresponding bitmap of the value searching for to extract the sentences containing the desired value. Bitmap indexes have various types depending on the sort of application. One can refer to the simple bitmap index, range bitmap index, interval bitmap index, multi-component bitmap index, etc.

D) Binary tree index

This index was proposed by Rudolf Bayer in 1971 [31]. For the binary index search, a number is initially compared to the root value of a tree. If this number is greater than the key, it will next be compared to the number in the subdirectory node on the right. And otherwise, it will be compared with the number in the subdirectory node on the left. This process is iterated successively in the tree to end at the leaves, which are the search results. In the leaves, the IDs are encrypted. Hence, after the search, these encrypted IDs should be sent to the client and sent back to the cloud once more after the decryption to return the desired line [32]. Binary tree indices have various types depending on the nature of the application. The B + Tree index [33-35], the MMB Cloud-Tree index [36], the U^2-Tree index [37], and so on are some of those indices.

The general search process with an index on the encrypted data is classified into five steps [38]:

1. Extraction of document keywords
2. Building a searchable index
3. Creating a search trapdoor

4- Searching the index based on the trapdoor

5. Returning the search results

• Keyword-based Search

1- Search with a single keyword

In this method, first, a list of discrete keywords is extracted by the file owner from the file set. When an authenticated user tries to search for a keyword, a search request is created in the figure of a trapdoor regarding the keyword. Following receiving the request from the server, the index stored in the cloud is searched and the set of files including the keyword is returned to the user. If the outsourcing environment is partial cloud data, the traditional searchable encryption designs are consistent with Boolean search keywords, Allowing the user to search for encrypted cloud data without violating data privacy [39, 40]. But in the nonexistence of massive cloud space, this file is not supported in the same way. In these circumstances, the user has to browse the restored files without decryption in similar files, only resulting in overhead processing. This also results in network traffic and undesired computing cost. Ranked keyword searching was proposed to overcome this critical bug, as well as ensure the accuracy of file recovery. The rank is defined by considering the certain criteria such number of keywords to increase the usability of system with recovering the files in the ranked keyword [41].

2- Search with Multi keywords

Nowadays, owing to the massive amount of data stored in the cloud and the great amount of search data connected with each keyword, studies are leaning toward multi-keyword searches to improve user search and search efficiency. Given that the user can search with Multi keywords in a single search, the search precision will also enhance. Employing multiple keywords narrows the number of results, and returns only the identical documents. Besides, to promote the status of a scheme, the ranked search with multi keywords on the encrypted data was recommended [42-44].

3- Search with the fuzzy keyword

In the search designs discussed earlier, only the accurate keyword searching is supported. But existing encryption methods are not perfect for the cloud computing scenario and cannot endure minor inconsistencies and typos in the keyword. Commonly, user search input may not precisely match the preset keywords. Hence, to overcome this problem and to cover the users' search more reliably, the search with fuzzy keywords was presented. The fuzzy keyword search design returns the search results following the following rules:

- In case the user search input matches the predefined keyword exactly, the server returns the files including the keyword.
- If there is a text or spelling inconsistency in the search entry, the server returns the nearest match results based on the keyword connotative similarity. This search architecture is illustrated in Fig. 5 [45, 46].

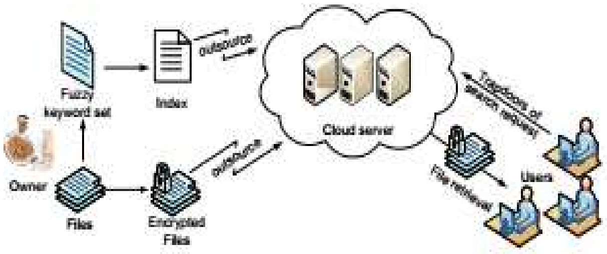

Fig. 5: The fuzzy keyword search architecture

4- Conjunctive-keyword search

If the user is interested in searching for documents holding each of multi keywords, either the server capacities for each of the keywords should be provided individually to the user and rely on quadruple calculation for the right set of documents, or to facilitate such searches, the user may store additional data, such as the keyword, on the server. Neither of these solutions is acceptable. The former allows the server to learn documents that correspond to each separate keyword in the search while the latter leads to exponential storing. To overcome this difficulty and enhance search results in response to user requests, the Conjunctive-keyword search was proposed [47]. In the following, with the development of this method, more flawless schemes are presented and better results in response to searches are offered [48-50].

5- synonyms-keyword search

In a real search scenario, the search of cloud customers may be synonymous with predefined keywords. To better and fully cover the search issued by the user, the search was granted with synonymous keywords. But cloud servers are better to support this keyword search feature for encrypted files. Examples in this regard include a web browser search tool for searching web pages. Also, search expressions or highlighting them or color-coding synonyms on a web page to provide the words through which are found by them is notable [51, 52].

It cannot be certainly claimed that the search techniques mentioned regarding the encrypted data cover all available techniques. However, attempts have been made for most of these methods to be examined and pointed. It is worth noting that some of these methods overlap with other(s). In other words, the techniques and methods of them are used in combination, and this is one of the necessities of these techniques. In conclusion, Table 3 reviews the various search methods, the types of performance, the advantages and disadvantages of each, so that they can be adequately employed and exploited concerning the ends needed in the layouts ahead.

Table 3: Encrypted Data Search Techniques

Row	Technique	Method	Performance	Benefits	Drawbacks
1	successive scan	Initial search [23]	The document is first encrypted with XOR of plain text with pseudo-random bits, and then the W and k of the server must be searched for the word	simplicity	Disclosure of W and k
2		Controlled search [23]	Similar to initial search, except that **k** is chosen randomly as well	Nondisclosure of encryption key directly	Disclosure of W
3		Hidden search [23]	Like the initial search, however, words are also encrypted and then used with the definitive encryption algorithm Ek."	Nondisclosure of encryption key and directly	Disclosure of statistical information
4	**Index-based search**	Direct index [27,28]	tuples of the document and keyword list. **Key** represents a document, and the **value** is the list of keywords utilized in it	Fast searching	Disclosure of a keyword's distribution information as time passes
5		Reverse index [29]	Similar to direct index. Except that **key** represents a keyword and **value** represents the associated documents	Lower searching time	Unsecure against statistical attacks
6		Bitmap index [30, 53-56]	For each **value**, the bitmap is referred to while searching over a specific value	Lesser intricacy	High calculation overhead

7		Binary tree index [31-37]	Searching for the user's request in tree index. The IDs are encrypted in the leaves. These IDs are returned after searching	More coverage	High calculation overhead
8	**Keyword-based search**	Search with a single keyword [39-41]	Searching for user-encrypted text is done using a keyword	Exact searching equal to the keyword	Unsecure against statistical attacks
9		Search with multi keywords [42-44]	Searching for user-encrypted text is done using multiple keywords	Multiple user request search	Higher overhead with searching of similar documents
10		Search with the fuzzy keyword [45, 46]	Searching over the encrypted text using keywords and covering typos and incorrect keywords	Better matching of user requests with documentation	High cost
11		Conjunctive-keyword search [47-50]	Searching for user-encrypted text is done continuously using multiple keywords	Coverage of the user's request with Conjunctive words	High cost
12		synonyms-keyword search [51, 52]	Searching for user-encrypted text is done using a keyword and covering synonymous keywords	Better covering of the user's requests	Ambiguous words lead to returning of irrelevant records

Conclusion

This article reviews multiple search methods in encrypted cloud data. Examining several techniques for searching encrypted data enables users to conduct searches based on the nature of requirements and desires while enhancing security. The cloud server can search the encrypted data but cannot access critical information in the data. Of course, doing this sort of search extends the search cost. The principal objective of all these methods is to prevent knowing critical information from the set of documents, keywords, indexes, and searches during searching. As such, users' privacy is preserved.

References

1. Azhir E, Jafari Navimipour N, Hosseinzadeh M, Sharifi A, Darwesh A. Deterministic and non-deterministic query optimization techniques in the cloud computing. Concurrency and Computation: Practice and Experience. 2019 Sep 10;31(17):e5240.
2. Aznoli F, Navimipour NJ. Cloud services recommendation: reviewing the recent advances and suggesting the future research directions. J Netw Comput Appl. 2017;77:73-86.
3. Naseri A, Navimipour NJ. A new agent-based method for QoS-aware cloud service composition using particle swarm optimization algorithm. J Ambient Intell Humaniz Comput. 2018;x:1-14.
4. Chiregi M, Navimipour NJ. A comprehensive study of the trust evaluation mechanisms in the cloud computing. J Serv Sci Res. 2017;9(1):1-30.
5. Milani BA, Navimipour NJ. A comprehensive review of the data replication techniques in the cloud environments: major trends and future directions. J Netw Comput Appl. 2016;64:229-238.
6. Buyya R, Broberg J, Gośćiński A. Cloud Computing: Principles and Paradigms 2011 Hoboken.
7. Proaño J, Carrión C, Caminero B. Empirical modeling and simulation of an heterogeneous Cloud computing environment. Parallel Computing. 2019 Apr 1;83:118-34.
8. Vakili A, Navimipour NJ. Comprehensive and systematic review of the service composition mechanisms in the cloud environments. Journal of Network and Computer Applications. 2017 Mar 1;81:24-36.
9. Sheikholeslami F, Jafari NN. Auction-based resource allocation mechanisms in the cloud environments: a review of the literature and reflection on future challenges. Concurr Comp Pract E. 2018;30(16):e4456.
10. Cheng L, Tachmazidis I, Kotoulas S, Antoniou G. Design and evaluation of small–large outer joins in cloud computing environments. J Parallel Distrib Comput. 2017;110:2-15.
11. Navimipour NJ, Navin AH, Rahmani AM, Hosseinzadeh M. Behavioral modeling and automated verification of a cloud-based framework to share the knowledge and skills of human resources. Comput Ind. 2015;68:65-77.
12. Souri A, Navimipour NJ, Rahmani AM. Formal verification approaches and standards in the cloud computing: a comprehensive and systematic review. Comput Stand Interfaces. 2018;58:1-22.
13. Chiregi M, Navimipour NJ. A new method for trust and reputation evaluation in the cloud environments using the recommendations of opinion leaders' entities and removing the effect of troll entities. Comput Hum Behav. 2016;60:280-292.
14. Navimipour NJ, Rahmani AM, Navin AH, Hosseinzadeh M. Expert cloud: a cloud-based framework to share the knowledge and skills of human resources. Comput Hum Behav. 2015;46:57-74.
15. García ÁL, del Castillo EF, Fernández PO. Standards for enabling heterogeneous IaaS cloud federations. Comput Stand Interfaces. 2016;47:19-23.
16. 16. Yoganandani, P. S., Rahul Johari, Kunal Krishna, Rahul Kumar, and Sumit Maurya. "C.: Clearing The Clouds On Cloud Computing: Survey Paper." International Journal of Recent Development in Engineering and Technology (2014): 117-121.

17. Huang Z, Liu S, Mao X, Chen K, Li J. Insight of the protection for data security under selective opening attacks. Information Sciences. 2017 Oct 1;412:223-41.

18. Li J, Huang X, Li J, Chen X, Xiang Y. Securely outsourcing attribute-based encryption with checkability. IEEE Transactions on Parallel and Distributed Systems. 2013 Oct 21;25(8):2201-10.

19. Xia Z, Wang X, Sun X, Wang Q. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE transactions on parallel and distributed systems. 2015 Feb 11;27(2):340-52.

20. Qiu L, Sun X, Xu J. Categorical quantum cryptography for access control in cloud computing. Soft computing. 2018 Oct 1;22(19):6363-70.

21. Fox A, Griffith R, Joseph A, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I. Above the clouds: A berkeley view of cloud computing. Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS. 2009 Feb 10;28(13):2009.

22. Poh GS, Chin JJ, Yau WC, Choo KK, Mohamad MS. Searchable symmetric encryption: designs and challenges. ACM Computing Surveys (CSUR). 2017 May 26;50(3):1-37.

23. Song DX, Wagner D, Perrig A. Practical techniques for searches on encrypted data. InProceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000 2000 May 14 (pp. 44-55). IEEE.

24. Bösch C, Hartel P, Jonker W, Peter A. A survey of provably secure searchable encryption. ACM Computing Surveys (CSUR). 2014 Aug 25;47(2):1-51.

25. Orencik C, Selcuk A, Savas E, Kantarcioglu M. Multi-Keyword search over encrypted data with scoring and search pattern obfuscation. International Journal of Information Security. 2016 Jun 1;15(3):251-69.

26. Kamara S, Papamanthou C, Roeder T. Dynamic searchable symmetric encryption. InProceedings of the 2012 ACM conference on Computer and communications security 2012 Oct 16 (pp. 965-976).

27. Cash D, Grubbs P, Perry J, Ristenpart T. Leakage-abuse attacks against searchable encryption. InProceedings of the 22nd ACM SIGSAC conference on computer and communications security 2015 Oct 12 (pp. 668-679).

28. Chang YC, Mitzenmacher M. Privacy preserving keyword searches on remote encrypted data. InInternational Conference on Applied Cryptography and Network Security 2005 Jun 7 (pp. 442-455). Springer, Berlin, Heidelberg.

29. Curtmola R, Garay J, Kamara S, Ostrovsky R. Searchable symmetric encryption: improved definitions and efficient constructions. Journal of Computer Security. 2011 Jan 1;19(5):895-934.

30. Wong, H. K. (1985). Bit transposed files.

31. Bayer R. Binary B-trees for virtual memory. InProceedings of the 1971 ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control 1971 Nov 11 (pp. 219-235).

32. Comer D. Ubiquitous B-tree. ACM Computing Surveys (CSUR). 1979 Jun 1;11(2):121-37.

33. Geetha S, Deepalakshmi P. Enhanced Energy in Sensors by Avoiding Voids and Saving Sensitive Data on Cloud Using B+ Tree Index with Retrieval of Query Predicates. Mobile Networks and Applications. 2019 Feb 15;24(1):234-47.

34. Chen Q, Ye S, Li Y, Zhang M, Zhu W. A Multi-keyword Searchable Encryption Based on B+ Tree. DEStech Transactions on Computer Science and Engineering. 2017(iceiti).

35. Wu S, Jiang D, Ooi BC, Wu KL. Efficient B-tree based indexing for cloud data processing. Proceedings of the VLDB Endowment. 2010 Sep 1;3(1-2):1207-18.

36. Karthikeyan MB, Sanjay C, Kumar SR, Seerangan LV, Manikandan E. MMB Cloud-Tree: Verifiable Cloud Service Selection. International Journal of Advanced Engineering, Management and Science. 2018;4(4):239987.

37. Gao X, Gao Y, Zhu Y, Chen G. U 2-Tree: A Universal Two-Layer Distributed Indexing Scheme for Cloud Storage System. IEEE/ACM Transactions on Networking. 2019 Jan 25;27(1):201-13.

38. Sonawane K, Dagade R. A Survey on Multi-Keyword Ranked Search over Encrypted Cloud Data with Multile Data Owners. International Journal of Computer Applications. 2017;162(11).

39. Chen R, Mu Y, Yang G, Guo F, Huang X, Wang X, Wang Y. Server-aided public key encryption with keyword search. IEEE Transactions on Information Forensics and Security. 2016 Aug 10;11(12):2833-42.

40. Wu TY, Chen CM, Wang KH, Pan JS, Zheng W, Chu SC, Roddick JF. Security Analysis of Rhee et al.'s Public Encryption with Keyword Search Schemes: A Review. J. Netw. Intell.. 2018 Feb;3(1):16-25.

41. Hu M, Gao H, Gao T. Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data by Chaos Based Arithmetic Coding and Confusion. IJ Network Security. 2019 Jan 1;21(1):105-14.

42. Fernandes, B. J., Anusha P., Louella M. M. e Colaco. Privacy Preserving Multi-Keyword Ranked Search of Medical Data. International Journal of Engineering Science, 2017; 10452.

43. Dai H, Ji Y, Yang G, Huang H, Yi X. A privacy-preserving multi-keyword ranked search over encrypted data in hybrid clouds. IEEE Access. 2019 Dec 31;8:4895-907.

44. Nabil M, Alsharif A, Sherif A, Mahmoud M, Younis M. Efficient multi-keyword ranked search over encrypted data for multi-data-owner settings. In2018 IEEE International Conference on Communications (ICC) 2018 May 20 (pp. 1-6). IEEE.

45. Li J, Wang Q, Wang C, Cao N, Ren K, Lou W. Fuzzy keyword search over encrypted data in cloud computing. In2010 Proceedings IEEE INFOCOM 2010 Mar 14 (pp. 1-5). IEEE.

46. Karande S. Multi-keyword Ranked Fuzzy Keyword Search Over Encrypted Cloud Data. InInnovations in Computer Science and Engineering 2019 (pp. 543-549). Springer, Singapore.
47. Golle P, Staddon J, Waters B. Secure conjunctive keyword search over encrypted data. InInternational conference on applied cryptography and network security 2004 Jun 8 (pp. 31-45). Springer, Berlin, Heidelberg.
48. Hu C, Song X, Liu P, Xin Y, Xu Y, Duan Y, Hao R. Forward secure conjunctive-keyword searchable encryption. IEEE Access. 2019 Mar 5;7:35035-48.
49. Wang P, Xiang T, Li X, Xiang H. Public key encryption with conjunctive keyword search on lattice. Journal of Information Security and Applications. 2020 Apr 1;51:102433.
50. Wang Y, Wang J, Sun S, Miao M, Chen X. Toward forward secure SSE supporting conjunctive keyword search. IEEE Access. 2019 Sep 27;7:142762-72.
51. Fu Z, Sun X, Xia Z, Zhou L, Shu J. Multi-keyword ranked search supporting synonym query over encrypted data in cloud computing. In2013 IEEE 32nd International Performance Computing and Communications Conference (IPCCC) 2013 Dec 6 (pp. 1-8). IEEE.
52. Fu Z, Shu J, Sun X, Linge N. Smart cloud search services: verifiable keyword-based semantic search over encrypted cloud data. IEEE Transactions on Consumer Electronics. 2014 Nov;60(4):762-70.
53. Wu K, Otoo EJ, Shoshani A. Compressing bitmap indexes for faster search operations. InProceedings 14th International Conference on Scientific and Statistical Database Management 2002 Jul 24 (pp. 99-108). IEEE.
54. Stockinger K, Wu K. Bitmap indices for data warehouses. InData Warehouses and OLAP: Concepts, Architectures and Solutions 2007 (pp. 157-178). IGI Global.
55. Chan CY, Ioannidis YE. An efficient bitmap encoding scheme for selection queries. InProceedings of the 1999 ACM SIGMOD international conference on Management of data 1999 Jun 1 (pp. 215-226).
56. Chan CY, Ioannidis YE. Bitmap index design and evaluation. InProceedings of the 1998 ACM SIGMOD international conference on Management of data 1998 Jun 1 (pp. 355-366).