International Journal of Engineering & Technology, 7 (2) (2018) 617-620

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i2.9254
Research paper

A study on searchable encryption schemes

Sunitha .S. Buchade [1] *, Prof. P. R. Devale [2]

[1] Department of Information Technology, Bharati Vidyapeeth University College of Engineering, Pune, India
[2]Department of Information Technology, Bharati Vidyapeeth University College of Engineering, Pune, India
**Corresponding author E-mail: puresoni06@gmail.com*

Abstract

To manage large volume of data, most of the organizations and data owners outsource their data to remote cloud storage servers. Since the remote servers are untrusted party, the data has to be encrypted to achieve security and privacy. But encryption and decryption causes communication overhead for many data utilization operations like searching and updating. To overcome this contradiction, searchable encryption notion was introduced. Searchable encryption (SE) is an ability of a server to search upon the ciphertext and retrieve the data without decrypting it. By searching on the ciphertext it protects the user's tactful data. This article concentrates on study of various resilient keyword search schemes in area of Cloud SE. The main basis for SE evolution is attaining the challenging task of efficient enciphered data utilization with privacy check. Thus schemes are developed considering their search capability, efficiency and security. In this paper we summarize and analyze various available keyword searchable schemes based on query expressiveness and query correctness. We review articles which have extensively researched in the field of SE with various keyword search schemes.

Keywords: *Cloud; Keyword Search; Query Expressiveness; Searchable Encryption; Wildcard Search.*

1. Introduction

Cloud systems and mobile devices both work in similar fashion as the data can be accessed from remote location by the data users. Cloud model is a large group of resources such as systems that are linked by networks that provide extended services for computing and file storage. Cloud enables stakeholders to use hardware and software that are operated at remote locations by third parties. Development in cloud has enabled facility for data availability, ease of data access and decreased infrastructure framework cost by outsourcing tactful data to remote cloud servers. Maintenance cost and overhead of local storage has been relieved by remote cloud storage. Users prefer cloud storage as it can be accessed at any time from any location.

Cloud storage has many security concerns that inhibit the data utilization by users effectively. The remote cloud server is a semi-honest and untrusted third party that might be curious about the data and might try to exploit the tactful data. Data outsourcing to the server means transfer of physical access of the data and delegation of administration rights to the server. So, it is mandatory to assure the privacy and security of user's tactful data. The realistic way to achieve this is to first encrypt the data and then outsource encipher. Encryption gives end-to-end security and privacy as and when the data is moved from the user's local storage. Though encryption assures privacy and security, it also complicates the things for the server to execute any meaningful operations especially search operations on encipher.

In a scenario of search operation, query keyword message is sent to remote cloud server and the associated documents are retrieved. Once the search operation is completed, the server returns the fetched documents as the result message. However, both stored content and query keyword content is revealed while searching upon encipher. Thus, though encryption secures the tactful data, it also inhibits search operation. A simple approach for search is to download the entire cipher, decrypt it and then search on decrypted or plaintext data. However, this approach is unrealistic. Subsequently, we need a solution which supports data privacy and simultaneously preserves search operation.

SE along with encryption function, it also performs keyword search on enciphers. In SE schemes, encrypted data or the ciphertext is outsourced to the server while preserving the search ability on them. The security is achieved by maintaining the document and keyword privacy. The two main dimensions of SE are Symmetric or private key Searchable encryption (SSE) and asymmetric or public key Searchable encryption with keyword search (PEKS). In Symmetric type (SSE), only users holding private key are authorized to generate ciphertext and trapdoors for search functionality. In asymmetric type (PEKS), users who have the knowledge of public key are authorized to generate ciphertext and only users holding private keys are authorized to build trapdoors for search functions on enciphers.

The SE scheme should not only provide effective search on encipher but must also satisfy other functionality problems faced while searching, like query representation, multi user group access permissions, users authorization and repudiation. Also, the returned search results must also be verified as they are returned by semi honest remote cloud server.

In the cloud scenario, a keyword is an index term that is encrypted and stored along with encrypted data on the remote cloud server. These terms are later on used by authorized user to search relevant data on encipher. However, as users are humans and humans tend to make errors, they can search a query which might have spelling mistakes, typos or might not know the full exact keyword or enter an incorrect keyword. This noted problem of searchable encryption is the query representation and its expression problem that needs to be paid attention by searchable schemes. Along with this, cloud is accessed by multiple users at a time with different access levels, so the searchable scheme should also satisfy accessibility problem thus maintaining the privacy. The search results are re-

turned by cloud which is untrusted, therefore searchable scheme should also verify the returned search results. So, other than the search ability, we can presume that data encryption creates difficulty for user authorization, access and repudiation process.
In this paper we summarize and analyze various available searchable schemes. The entered query keyword by the querier or user can be single or multiple and it might contain typos, spelling mistakes or can be incomplete. To perform search by using such queries and attain search efficiency is a challenging task. We review various papers in SE field related to search functionality with keywords. The remainder of this review article is structured as follows: Initially, we have presented a basic model of SE in Section 2. Then we have reviewed and discussed various searchable keyword schemes in Section 3. Finally we have concluded the paper in Section 4.

2. Model of traditional searchable encryption

A basic searchable encryption is combination of 3 units: a data owner A, a semi-honest remote server S, and group of authorized data users who have access to search. The functionality and ability of each unit is as follows:
Data owner: Is the unit which outsource a collection of files F = {F_1, F_2, …. , F_n} along with some keywords which is later used in search operation. The owner encrypts the files with associated keywords and then sends this enciphers to the remote server.
Authorized user of data: If an authorized user who has access to the data wants to search the files that contains the desired indexed keyword of interest, he/she has to send a query keyword message that is its trapdoor to the remote server. This keyword message has let to evolution of various searchable schemes which is discussed in Section 3 of this paper. Once search is completed, the remote server returns the files with the desired keywords to the authorized user.
Semi-honest remote server: The remote server unit performs the search tasks on encipher. When the remote server receives the query keyword request along with its trapdoor from an authorized user, it performs search operation upon the cipher and the associated files that contain the requested keyword are retrieved and sent to the user by the server. We assume that server is semi honest and curious. This means that server will follow the conventions properly, but it might attempt to analyse the requested data and try to gain additional information from it.

3. Literature Survey

This section describes various searchable keyword schemes based on query expressiveness and major research work on them.

3.1. Search using single keyword

Song, et al [1] proposed the first construction of SSE scheme. The basic structure of the scheme is to encrypt the data first, search and then decrypt. The concept of this scheme is to XOR the plaintext keywords with pseudorandom bit sequence to generate ciphertext, and later on the ciphertext sequence are split into left and right parts. For search operation the keywords are encrypted and sent to the server. The server checks and sends the relevant file and later on the files are decrypted XORing the ciphertext with the pseudorandom bit. Thus their method ensures the privacy of keywords and also the security of plaintext data. Their approach uses sequential scan for searching which makes the search operation slow as search time is proportional to the words of document collection in the server. The search also requires the size of the keywords to be pre fixed.
To achieve search efficiency Goh, et al [2] proposed a searchable symmetric scheme with secure index for each document. The secure index is made using bloom filter of plaintext files. Bloom filter is a probabilistic data structure used to check if an element exists in the given set.

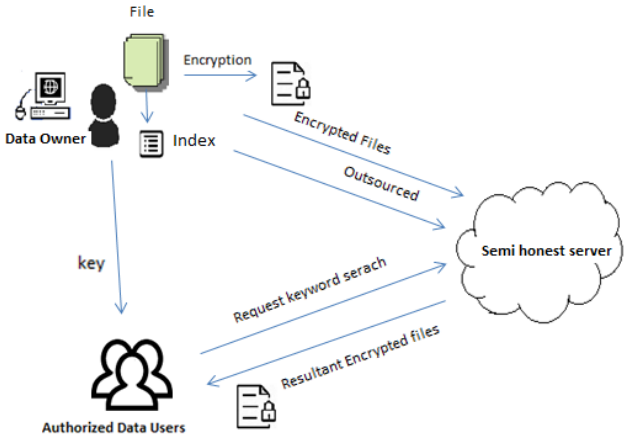

Fig. 1: System Model of Traditional SE Using Keyword.

Bloom filter along with trapdoors of unique keywords for each file is constructed and uploaded on the server. For search operation, the trapdoors for the search keyword is computed and sent to the server. Once the request is received the server uses bloom filter to check if the trapdoor of the requested search keyword is present in the file and sends the corresponding file identifiers. In their method, instead of scanning and searching the entire ciphertext, we can directly search the index thus reducing the search time. However, the major disadvantage of bloom filter is that they have false positive probability and it is inevitable.

3.2. Search using conjunctive keyword

This type of search results in fetching of document with several keywords in single query. This increases the usability of search operation and improves encrypted data utilization. A simple mechanism for several queried keywords is to search for each keyword separately and then combine the results of all the searches. However, it is time consuming and inefficient.
Golle, et al [3] presented two schemes for search of conjunctive keywords. In the first scheme the communication cost is proportional to the number of files and documents, but major of the cost can be induced offline even before submitting the conjunctive query. The security notion of their scheme is maintained by DDH assumption. The requirement of second scheme is constant communication. In this scheme, a special set of keyword attribute corresponding to each document is maintained, for example in email the keyword attributes can be considered as ‘From’, ‘To’, ‘Subject’. The communication cost of the second scheme is in linear with number of keyword attributes. The security notion of their scheme is maintained by Advance hardness assumption. The major drawback of these schemes is that keyword attributes are learnt by the server and thus the keyword attribute privacy is lost.
Ballard, et al [4] proposed two schemes for search of conjunctive keyword over symmetrically encrypted plaintext. The first construction scheme is based on secret sharing method with threshold of Shamir’s. The method calculates polynomial interpolation and it consists of two algorithms Share and Recover. The other construction is built on Bilinear Maps. Their scheme obtains constant communication overhead by placing the computational overhead on the remote server. The construction security relies on novel variant of XDH assumption. The first construction generates trapdoors linear to the number of searching documents. To address this issue the second construction is built where trapdoors of constant are required.
Faber, et al [5] considered conjunctive keyword query as a Boolean query. They extended their scheme to support wildcard, range, phrase and substring queries. In their approach, the seldom keywords are searched first and then search results of other keywords are applied. Their system is the first Boolean query supporting sub linear SSE construction.

3.3. Search using fuzzy keywords

In existing solutions of SE, the user creates a trapdoor of the queried or searched keyword and sends it to the remote server and the remote server based on the trapdoor returns the matched files or documents to the user. However, if the predefined keyword does not match the search string, such as 'colour' and 'color' the search operation fails. The traditional methods concentrated only on retrieving the files that exactly matched the search string or the queried keywords. The typos, simple spelling mistakes and format representing inconsistencies were not considered in earlier methods.

To overcome this Li, et al [6] proposed a notion of fuzzy search of keywords. The search was performed with pre-computed similarity structure of the keywords. To calculate the string similarity of keywords, Jin used the concept of Edit distance. Edit distance is number of operations needed to convert one word into another word. It consists of techniques like substitution, insertion and deletion. Based on the edit distance he introduced two approaches as the solution for the search of fuzzy keywords.

The first is the straightforward method. It gives the insights of how the basic fuzzy keyword search works. A set is constructed by listing all the possible variants of the queried keyword that fulfils the similarity criterion. For example, the given enumerated list after substituting with '*' at the first or any other positioned letter of the keyword 'hat' is: {hat*, *at, *hat, h*at, h*t, ha*t, ha*, hat*} However, the basic fuzzy search method introduces storage complexity problem. Increase in size of set, increases the size of index. Hence Li introduced an advanced technique known as wildcard based fuzzy set construction. In straight forward technique all the possible variants were listed even if the operation was performed on the same position. Wildcard is used to represent the edit operation to be performed at the given position. Thus the set of number of possible variants constructed is less and the index size is also reduced. However, with increase in keyword error tolerance, the index file also expands and thus resulting in storage complexity. Their search approach is effective only for single keyword, for multiple keywords their search process has to be repeated multiple times.

Kuzu, et al [7] addresses the issues of fuzzy multiple query keyword search by adopting bloom filers and LSH technique. Instead of edit distance, Euclidean distance between the words is used to calculate the keyword similarity criterion. Their approach also addresses the issues of typos, spelling checks and format representation inconsistencies for multiple keywords in one round and in single query. LSH function when given similar or closely related input, they result in same output hash values. So user entered queries with small typos may hash same value as the exact keyword does, thus making search operation feasible and easy. LSH function requires vectors as input so keywords are converted to bigram set and then later on represented as vectors. LSH function is used to build the bloom filter based secure index i.e, a file secure index is created using bloom filter that contains vector representation of keywords inserted by LSH functions. The searching is performed using inner product matching algorithm. The proposed method eliminates the requirement of predefined dictionary and provides an efficient ranked search technique with constant index size.

3.4. Verifiable search

The cloud server can be curious or semi honest and can return incomplete, inaccurate or partial results. The cloud which is untrusted and malicious is the one to execute the search operations. Verifiable search of keywords can check for integrity and completeness of the search result. This verifies the accuracy of search results produced by hardware or software, corrupted storage, or semi trusted remote server.

Zheng, et al [8] addressed this issue with a novel technique named VABKS which is based on ABE scheme. Their technique uses trusted authority to assign credentials to the data members. To verify the authenticity of search results of ABKS scheme they use VABKS. However, the verification of authenticity of search results is costly. The scheme developed is prone to offline attacks as the search tokens and encrypted keyword can be attained easily by an attacker.

Liu, et al [9] presented a novel VABKS scheme based on KP-ABE. The model is supported by bloom filter, Access trees, various complexity assumptions, bilinear pairings and KP-ABE. Unlike the Zheng, et al [8] approach, who considered a secure private channel for issuing token, the KP-ABE approach does not need such assumptions. To overcome the requirement of secure channel, public key and secret key for remote server and data user is made. So they can utilize the public key of the remote server to re-encrypt the search tokens and ciphertext of keywords. This assures only the remote server can match the search token and encrypted keyword text. Thus it can hinder the offline attack. Utilization of public key of the remote server to re-encrypt the cipher text of keyword can guarantee only the remote server can perform the search operations. Their proposed technique is comparatively competent in verifying the integrity and authenticity of search results.

3.5. Ranked search

This search operation results in fetching of relevant documents to the users query keywords. The search operation is enhanced as the irrelevant documents are excluded from fetching. This reduces the data traffic and improves usability of the system.

Cao, et al [10] presented the first approach for rank basis search of multiple keywords with importance to privacy in cloud computing called MRSE. The efficient MRSE maintains the index privacy, data privacy, search privacy and access pattern. Coordinate matching calculates the similarity of multiple keywords that returns refined results. In their paper, coordinate matching concept is used for computing the similarity measure of keywords to the searched keyword. Coordinate matching is realized by inner product matching, that is the number of searched keywords in the documents. It mathematically computes the similarity extent of that document to the searched keyword. While creating secure index, each document is related to the binary vector as a child index where every single bit denotes whether equivalent keyword exists in the document. Even the search query is denoted as binary vector where every single bit implies whether the equivalent keyword exists in the requested search, so the similarity can be determined by inner product of data vector with the query vector. MRSE uses secure computation of inner product, which is taken from KNN technique and this resulted in two enhanced MRSE schemes that meet the privacy requirements. MRSE _I scheme includes random number 'r' in the extended dimension in every single query vector. The additional randomness causes additional difficulty for the remote server to learn the association between the received trapdoors. This also reduces the chances of keyword identification. However, there is no assurance of keyword privacy if the remote server has some background information of the used dataset. So to preserve privacy in the background model MRSE_II was proposed. In MRSE_II the privacy of keyword is leaked because the random variable is constant in the data vector. To mitigate the constant property in any document, more than one dummy keyword has to be added into every single data vector. The major drawback is their ranking of search results as it is only based on the number of matching keywords and not taking into account the significance of different keywords. Thus, there results are not very precise.

Sun, et al [11] suggested a ranked search scheme for multiple keywords by considering "cosine measure" concept. Their scheme obtains better search efficiency than linear search but at the price of search accuracy.

Xia, et al [12] proposed a dynamic rank search scheme for multiple keywords by a secure tree. Chen, et al [13] enhanced search efficiency of ranked search for multiple keywords, by using hierarchical clustering index. In their scheme, search time growth is linear when the size of data set growth is exponential. Table 1, shows summary of different SSE schemes reviewed in this paper.

In the table1, n denotes number of documents, w denotes keywords, |D (v)| denotes number of documents containing the keyword v, i denotes polynomial interpolation, |S| denotes the size of a set, c denotes comparison, m represents number of distinct keywords in the file.

3.6. Search with wildcard characters in keyword

Suga, et al [14], introduced a wildcard ('*') search method using Bloom filters. The wildcard represents any character. Their search comparison supported character matches of keywords rather than the whole exact match of keyword. So the search is performed by checking if any character is matching other than wildcard character. For example if the queried keyword is "h*t" , then the keywords matching 'h' and 't' in order with one character between them is checked for existence in the index. This verification step is executed by bloom filters. Bloom filter is used for creating a trapdoor and an index. One individual Bloom filter is used to store one keyword's characters. So for the keyword "h*t" if {"hot, hat, hut..."} are present in the index, then their corresponding files are fetched from the cloud. Pseudo random functions are used when characters are added to Bloom filters as they improve security. However, this method works only when one wildcard is expressed as one character. To overcome this drawback Hu, et al [15], proposed an improved dynamic SSE scheme in which, one wildcard is expressed as multiple characters. For every keyword, a single set is created by recording the character position. The character except wildcard is recorded. A bloom filter is used to create an index and trapdoor. In both, the characters are recorded in regular and reverse order based on position of the wildcard character. This scheme is secure to adaptive attacks.

Table 1: Analysis of Various SSE Schemes

Scheme	Query type	Search time	Security	Dynamics
Song, [1]	Single	O(n,w)	IND-CPA	No
Goh, [2]	Single	O(n)	IND1-CKA	Yes
Kamara,[20]	Single	O(|D(v)|)	IND-CKA2	Yes
Golle, [3]	Conjunctive	O(|D(v)|)	IND1-CKA2	No
Ballard, [4]	Conjunctive	ni	IND1-CKA2	No
Li, [6]	Fuzzy	n|S|c	IND1-CKA1	No
Hu, [15]	Wildcard	O(v')	IND1-CKA2	Yes

4. Conclusion

We presented a study of important SE schemes using keyword search with their approach for query expressiveness. Advantages and drawbacks of the approaches are highlighted for the above schemes. The presented SE schemes cover the evolution and development mainly focussed in three main directions: the efficiency, security and query expressiveness.

Though the traditional SE schemes have made progress, there are still some issues and drawbacks to be considered: First, from the point of SSE model, the complexity of some search schemes is proportional to the size of the documents stored on the remote server. A scheme in which search complexity is linear gains optimal results of search time. Moreover, in the multiuser setting the efficiency cannot be achieved due to large datasets used in applications leading to computational complexity. Second concern with large cloud data is how to develop the SE schemes that improves the scalability and efficiency without foregoing the security of data. Also there is no standard model developed for security that performs search operation without revealing the search and access pattern. Third concern is, we need to facilitate the scheme that supports resilient query representations. Query Expressiveness affects the users search experience and thus is very significant. There has been much research led for the extension and improvements in query expressiveness. However, there is need of advanced schemes with better security and search efficiency for achieving query expressiveness.

References

[1] Song D X, Wagner D, Perrig A, Practical techniques for searches on encrypted data, 2000 IEEE Symposium on Security and Privacy, Berkeley, California, USA, 2000, 44-55.

[2] Goh E J, Secure Indexes, IACR cryptology eprint archive, 2003, 216.

[3] Golle P, Staddon J, Waters B. Secure conjunctive keyword search over encrypted data. The 2nd International Conference on Applied Cryptography and Network Security, Yellow Mountain, China, 2004, 31-45. https://doi.org/10.1007/978-3-540-24852-1_3.

[4] Ballard L, Kamara s, Monrose F. Achieving efficient conjunctive keyword searches over encrypted data. The 7th International Conference on Information and Communications Security, Beijing, China, 2005, 414-426.

[5] Faber S, Jarecki S, Krawczyk H, et al. Rich queries on encrypted data: beyond exact matches. The 20th European Symposium on Research in Computer Security, Vienna, Austria, 2015, 123-145. https://doi.org/10.1007/978-3-319-24177-7_7.

[6] Li J, Wang Q, Wang C, et al. Fuzzy keyword search over encrypted data in cloud computing. The 29th IEEE International Conference on Computer, San Diego, USA, 2010, 441-445. https://doi.org/10.1109/INFCOM.2010.5462196.

[7] Kuzu M, Islam M S, Kantarcioglu M. Efficient similarity search over encrypted data. The28th IEEE International Conference on Data Engineering, Washington, USA, 2012, 1156-1167. https://doi.org/10.1109/ICDE.2012.23.

[8] Zheng Q, Xu S, Ateniese G. VABKS: verifiable attribute-based keyword search over outsourced encrypted data. IEEE Conference on Computer Communications, Toronto, Canada, 2014, 522-530. https://doi.org/10.1109/INFOCOM.2014.6847976.

[9] Liu P, Wang J, Ma H, et al. Efficient verifiable public key encryption with keyword search based on KP-ABE. The 9th International Conference on Broadband and Wireless Computing Communication and Applications (BWCCA), Guangdong, China, 2014, 584-589. https://doi.org/10.1109/BWCCA.2014.119.

[10] Cao N, Wang C, Li M, et al. Privacy preserving multi-keyword ranked search over encrypted cloud data. The 30th International Conference on Computer Communications, Shanghai, China, 2011, 829-837. https://doi.org/10.1109/INFCOM.2011.5935306.

[11] Sun W, Wang B, Cao N, et al. Verifiable privacy-preserving multi-keyword text search in cloud supporting similarity based ranking. IEEE transactions on Parallel and Distributed Systems, 2014, Vol. 25, 3025-3035. https://doi.org/10.1109/TPDS.2013.282.

[12] Xia Z, Wang X, Sun X, et al. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. IEEE transctions on parallel and distributed systems, 2016, Vol. 27, 340-352.

[13] Chen C, Zhu X, Shen P, et al. An efficient privacy-preserving ranked keyword search method. IEEE Transactions on Parallel and Distributed Systems, 2016, Vol. 27, 951-963. https://doi.org/10.1109/TPDS.2015.2425407.

[14] Suga T, Nishide T, Sakurai K. Secure keyword search using Bloom filter with specified character position. International Conference on Provable Security. Springer Berlin Heidelberg, 2012, 235-252. https://doi.org/10.1007/978-3-642-33272-2_15.

[15] Hu C, Han L. Efficient wildcard search over encrypted data. International Journal of Information Security. 2015, 1-9.

[16] Wang Y, Wang J, Chen X. Secure searchable encryption: a survey. Journal of Communications and Information Networks, 2016, Vol 1, 52-65. https://doi.org/10.1007/BF03391580.

[17] Kale P V, Welekar R. A survey on different techniques for encrypted cloud data. International Conference on Intelligent Computing and Control System, 2017, 245-247. https://doi.org/10.1109/ICCONS.2017.8250718.

[18] Li B, Jiang J. Search on encrypted data: state of arts. Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), IEEE, 2016, 2022-2031.

[19] Zhang J, Semantic-based searchable encryption in cloud: issues and challenges. 2015 First International Conference on Computational Intelligence Theory, Systems and Applications, 2015, 163-165. https://doi.org/10.1109/CCITSA.2015.29.

[20] Kamara S, Papamanthou C, Roeder T. Dynamic searchable symmetric encryption. Proceeding of ACM Conference on Computer and Communications Security (CCS), USA, 2012, 965-976. https://doi.org/10.1145/2382196.2382298.

[21] Bosch C, Hartel P, Jonker W, et al. A survey of provably secure searchable encryption. ACM Computing Surveys, Vol 47, 2014, 1-51. https://doi.org/10.1145/2636328.