OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

A Secure Dynamic Search Scheme for Encrypted Data on Cloud with Access Control

Varalatchoumy M,
(Department Of ISE,BNM Institute Of Technology,India)

ABSTRACT: *A search scheme that retrieves encrypted files based on their rankings. It supports dynamic operations such as dynamic insertion and deletion. The index construction is based on the TF*IDF model. The keywords are converted into hash codes using MD5 algorithm and the files are encrypted using RSA algorithm. It provides access control wherein the files can be accessed by a certain class of users to whom the files belong by assigning a grade to the files as well as the users.*

Keywords : *Cloud, Security, Dynamic, Ranking, Searchable Encryption.*

I. INTRODUCTION

Cloud computing has been in trend lately. Every person wants to save their data on cloud so that they can access it easily from anywhere anytime and also to reduce storage costs on their servers. Since the evolution of cloud, the dependency of people on the cloud has increased. The benefits of storing data on cloud comes with a concern for security. If the cloud gets attacked, all the data may be stolen. So the data needs to be stored in an encrypted format in the cloud. This poses a problem when the data stored on the cloud needs to be searched and retrieved. As they are in encrypted format, a searchable encryption scheme is needed. The secure search scheme is meant to perform search on encrypted data stored in cloud and retrieve the best possible results. This search scheme has 5 main features: 1) Security: The cloud will not be able to make a connection with the keywords searched and the files retrieved because keywords are converted into hash codes and data on the cloud is already stored as cipher text before the search begins. 2) Ranked results: the results displayed are retrieved based on their ranks. The top-k results will be retrieved, k can be any number that denotes the number of results that we wish to retrieve. 3) Access control: the search happens among only those files that are related to a certain class of users. Therefore it results in search efficiency i.e, when there are large number of files that contain the keywords being searched, the search happens among those files that belong to the person searching. 4) Dynamic: when the files are inserted or removed from the cloud the ranks of each file update dynamically. Also the keywords and their weightage for each file will be removed when a file is deleted. Therefore the data owner doesn't have to manually go and update the file index and keyword index. 5) Multi-keyword support: this method not only works for single key words but also multi-keywords. This improves the search results by retrieving most relevant files that contain all of the queried keywords.

II. RELATED WORK

Song et al. [7] has proposed the first symmetric searchable encryption (SSE) scheme, and this scheme's search time is additive to the data collection size. In his work, each document is examined as words whose length are fixed in a sequence, and is indexed independently. This scheme supports update operations which are straight forward. Goh et al [8] has proposed definitions of formal security for SSE and scheme has been designed based on Bloom filter. In Goh's scheme search time is O (n), where n is the cardinality of collection of the document. Goh has also proposed a scheme that generates a sub-index that is bloom filter based on document for every keywords. So that the dynamic operations can be easily achieved by upgrading the Bloom filter along with the corresponding document. Curtmola et al. [10] has proposed two schemes SSE-1 and SSE-2. Both the schemes achieve the search time which is optimal. SSE-1 scheme is protected against chosen-keyword attacks (CKA1) and SSE-2 is protected against adaptive chosen-keyword attacks (CKA2). Cao et al. [26] achieved the first privacy-preserving multi-keyword ranked search scheme. In this scheme the documents and queries are entitled as dictionary size vector. \According to the number of query keywords which are matched the documents are ranked based on the "coordinate matching". Sun et al. [27] has presented a secure multi-keyword search scheme.

This scheme supports similitude-based ranking. The search algorithm of Sun et al. achieves better linear search efficiency. O¨ rencik et al. [28] has proposed a secure multi-keyword search method which makes use of local sensitive hash (LSH) functions to group the documents which are similar. Zhang et al [29] has proposed a scheme in a multi-owner model which deals with secure multi-keyword ranked search scheme. In this scheme, different secret keys are used by different data owners in order to encrypt the documents and keywords .Without knowing keys of these different data owners, authorized users can still query and search for keyword.

III. PROPOSED METHODOLOGY

The proposed search scheme ensures security by preventing the cloud from learning about the contents stored in it. This can be done by taking care of three things: 1) data privacy: the documents to be stored in cloud must be encrypted before uploading them. 2) Index privacy: the searchable index must also be in encrypted format. 3) Query privacy: the keywords used for searching the files must also be encrypted. By achieving these three security goals, the cloud is prevented from associating the index with the files and from finding the link between the query entered for searching and the retrieved files. This search scheme is also dynamic as it supports dynamic insertion and deletion. When a new file is uploaded, the index is dynamically updated i.e. the rank of keywords belonging to this file is calculated and updated while the ranks of keywords belonging to already existing files are recalculated with respect to the newly uploaded file. When a file is deleted on the private server, it removes that file on the cloud as well as all the keywords attached to that file. Once again the ranks are modified. There are two main modules here, the admin module and the user module. The admin is the one who uploads file to the cloud and the user is the one who performs the search. The admin will have a copy of all the files stored on cloud in his private server. The search happens on the files stored on the cloud. Here, the cloud is seen as an untrusted party. The Fig.1 below depicts architecture of the admin module and the user module.

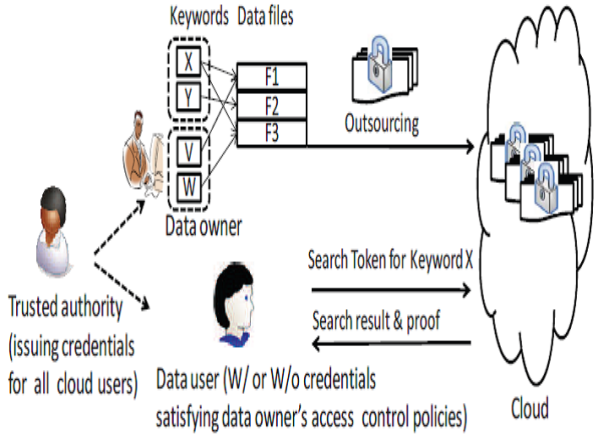

Fig 1: Architecture of admin and user module

A.The admin module: the admin has the authority to decide who can access which files. This is done by assigning grade to the user during user registration. A grade is a unique ID that is assigned to every user who registers to the admin's website. This grade is also assigned to a file before uploading it. So, a user can access a file only when it belongs to his grade. The admin module has three sub modules.

1. Ranking module: when the admin uploads a files, the keywords are extracted from that file by removing whitespaces, special characters and unnecessary words like pronouns etc. Once the keywords are extracted, their rankings are determined by two parameters. Term Frequency (TF) and Inverse Document Frequency (IDF). Term Frequency is the frequency of a keyword that occurs in a file. Inverse Document Frequency is the number of documents that contain a keyword. the rank of a file is determined by multiplying the TF value with the IDF value calculated using the given formulas.

2. Hashing module: After determining the ranks of the keywords, each keyword is concatenated with the grade assigned to the file to which it belongs. Then, these concatenated keywords are converted into hash codes. A hash code is a one way encrypted code of fixed size. Any cryptographic hashing algorithm can be used to

convert keywords into hash codes. Here, Message Digest 5 (MD5) algorithm has been adopted to convert the keywords into hash codes. MD5 converts the keywords into hash codes of the size of 128bits.

3. Encryption module: After hashing process is done, the original file is retained and encrypted using RSA. RSA private-key cryptography is adopted in this scheme. Each file is encrypted with the admin's private key. This provides sender authentication ensuring that the files are uploaded by admin. After the encryption of the file, it gets stored in the cloud. This way, the contents of the file will remain safe even if the cloud is attacked.

B. The user module: the user must first register with the admin so that the admin can assign a grade to the user. After registration the user will be given the details of his userID, grade and password via email. The User module has two sub modules:

1. Key Request module: the user has to send a request to the admin asking for the public key of the admin. The admin has the option of either accepting the key request or rejecting it. If it is accepted, the user receives the public key via email.

2. Search module: The user can then enter the keywords to be searched. These keywords are separated and each keyword is concatenated with the grade of the user which is taken from the user details database. These concatenated keywords are converted into hash codes using the same MD5 algorithm. The hash codes are compared with the existing hash table and if there is a match then all the files that are linked to the hash codes are retrieved based on their ranks. The user then needs to give the public key to download and get a decrypted version of the file.

IV. EXPERIMENTAL RESULTS

Results show that this search scheme is cost effective as it makes use of simple techniques like TF*IDF model, grade assigning and the usage of existing algorithms like MD5 and RSA to achieve a useful searching scheme. Since the hash codes are used to form the searchable index, search efficiency increases as the hash codes are of fixed size irrespective of the size of the keyword and can be compared quickly. The contents of a small file "sample.txt" is shown below:

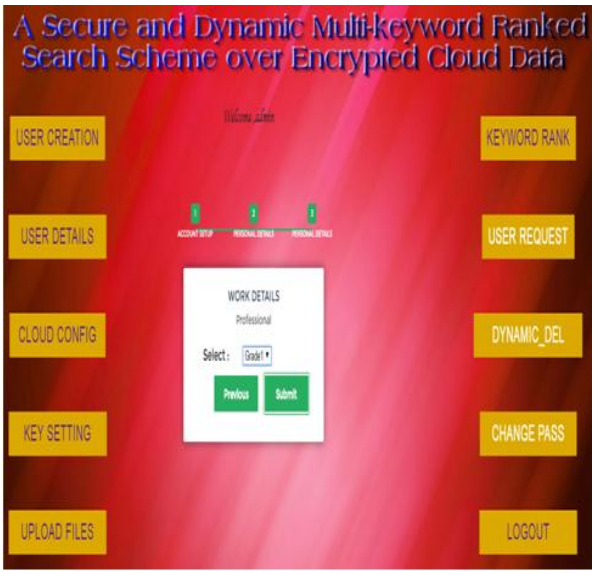

Fig 2. Snapshot 1

Upon clicking the User Creation button on the left side, a dialog box appears asking for the details of the user. The snapshot above shows the admin setting the grade for a user. Once the user is registered successfully, an email is sent to the user providing the grade and other details.

Fig 3. Snapshot 2

When the admin clicks on UPLOAD FILES button on the left side, a grade is selected for the file. Once the grade is submitted the keywords are extracted and displayed to us along with its rank.

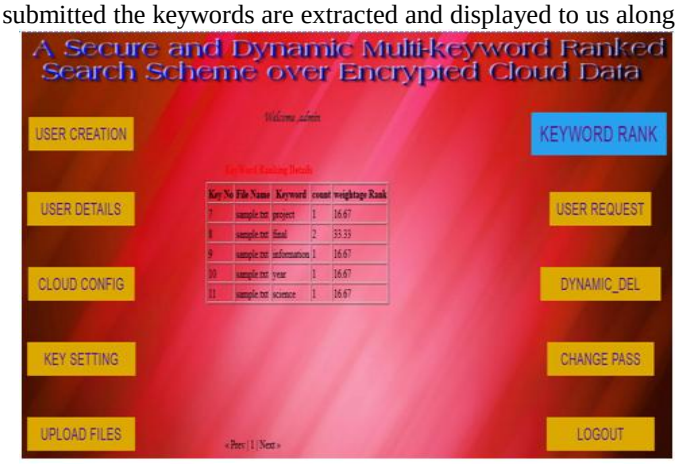

Fig 4. Snapshot 3

In the snapshot above, a file “sample.txt” is uploaded. This file has 6 keywords: information, science, final, year and project, where the keyword “final” has occurred twice and the rest of them have occurred only once. Hence “final” has a rank of 33.33(2/6) while the remaining have 16.67(1/6).

Fig 5. Snapshot 4

The registered user can send the key request by clicking on the SEND_REQUEST button on the left side. The user receives an acknowledgement as “key request sent successfully” when the Send Request button at the center is clicked.

Fig 6. Snapshot 5

The admin can check for key requests from the user by clicking on the User Request button on the right side. By clicking on the Send Public Key option next to the userID cell, The admin issues the public key to the user. This goes to the user's email id which has been provided by the user during registration.

Fig 7. Snapshot 6

When the user enters the keywords in the search bar the files containing those keywords are retrieved. The keywords entered here are "science" and "project".

Fig 8. Snapshot 7

The file "sample.txt" which was uploaded previously contains both the keywords "science" and "project". Therefore this file is displayed in the search results. When the Download button next to the search results is clicked, the user will be asked to give the public key of the admin.

Fig 9. Snapshot 8

V. CONCLUSION

A dynamic search scheme is proposed, which supports the canonical multi-keyword ranked search and also dynamic operations like insertion and deletion of documents in a secure and efficient way. A special tree based index structure has been constructed and Greedy Depth- first Search algorithm has been proposed to obtain better linear search efficiency. The scheme is designed to meet functional requirements of proposed system.It is very useful to store any documents in secure way in public cloud storage. This scheme comes to use where large number of files exist and top ranked files are to be retrieved at a faster rate. This system can be used in corporate, government officers and education institutes. In the proposed scheme, the data owner is responsible to generate information and update it and sends the updated information to the cloud server. Thus, the data owner will store index tree which is in unencrypted format and also stores the information for recalculating Inverse document frequency (IDF) values. Such data owner will not be suitable for cloud computing model. It will be difficult in future to design a dynamic searchable scheme where only the cloud server will complete update operation and ability of supporting multi-keyword ranked search. In a multi-user scheme there are many secure challenges, Firstly same secret key are used by all the users in a symmetric (SE) scheme for trapdoor generation. Here user revocation is bigger challenge, if there is a need for user revocation then rebuild the index and new keys which are secure are distributed to all the authorized users. Secondly, the assumptions of symmetric SE schemes are that the all the data users are trustworthy. But the data users who are dishonest may lead to many secure problems. For instance, dishonest data user may search for the document and also might distribute the documents which are decrypted to the unauthorized users and even dishonest data user may distribute the secure keys to unauthorized users. In the future works, Searchable encryption (SE) schemes will be improved to handle these problems.

REFERENCES

[1]. K.Ren,C.Wang,Q.Wangetal.,"Security challenges for the public cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, 2012.
[2]. S. Kamara and K. Lauter, "Cryptographic cloud storage," in Financial Cryptography and Data Security. Springer, 2010, pp. 136– 149.
[3]. C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.
[4]. O. Goldreich and R. Ostrovsky, "Software protection and simula- tion on oblivious rams," Journal of the ACM (JACM), vol. 43, no. 3, pp. 431–473, 1996.
[5]. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Advances in Cryptology- Eurocrypt 2004. Springer, 2004, pp. 506–522.
[6]. D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith III, "Public key encryption that allows pir queries," in Advances in Cryptology-CRYPTO 2007. Springer, 2007, pp. 50–67.
[7]. D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on. IEEE, 2000, pp. 44– 55.
[8]. E.-J. Goh et al., "Secure indexes." IACR Cryptology ePrint Archive, vol. 2003, p. 216, 2003.
[9]. Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proceedings of the Third in- ternational conference on Applied Cryptography and Network Security. Springer-Verlag, 2005, pp. 442–455.
[10]. R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient con- structions," in Proceedings of the 13th ACM conference on Computer and communications security. ACM, 2006, pp. 79–88.

[11]. J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in INFOCOM, 2010 Proceedings IEEE. IEEE, 2010, pp. 1–5.
[12]. M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient similarity search over encrypted data," in Data Engineering (ICDE), 2012 IEEE 28th International Conference on. IEEE, 2012, pp. 1156–1167.
[13]. C. Wang, K. Ren, S. Yu, and K. M. R. Urs, "Achieving usable and privacy-assured similarity search over outsourced cloud data," in INFOCOM, 2012 Proceedings IEEE. IEEE, 2012, pp. 451–459.
[14]. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS,VOL:PP YEAR:2015 13
[15]. B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi- keyword fuzzy search over encrypted data in the cloud," in IEEE INFOCOM, 2014.
[16]. P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in Applied Cryptography and Network Security. Springer, 2004, pp. 31–45.
[17]. Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in Proceedings of the First international conference on Pairing-Based Cryptography. Springer-Verlag, 2007, pp. 2–22.
[18]. L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient con- junctive keyword searches over encrypted data," in Proceedings of the 7th international conference on Information and Communications Security. Springer-Verlag, 2005, pp. 414–426.
[19]. D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in Proceedings of the 4th conference on Theory of cryptography. Springer-Verlag, 2007, pp. 535–554.
[20]. B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive-subset keywords search," Journal of Network and Computer Applications, vol. 34, no. 1, pp. 262–267, 2011.
[21]. J. Katz, A. Sahai, and B. Waters, "Predicate encryption support- ing disjunctions, polynomial equations, and inner products," in Advances in Cryptology–EUROCRYPT 2008. Springer, 2008, pp. 146–162.
[22]. E. Shen, E. Shi, and B. Waters, "Predicate privacy in encryption systems," in Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography. Springer-Verlag, 2009, pp. 457–473.
[23]. A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption," in Proceedings of the 29th Annual international conference on Theory and Applications of Cryptographic Techniques. Springer-Verlag, 2010, pp. 62–91.
[24]. A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A. L. Varna, S. He, M.Wu,andD.W.Oard,"Confidentiality-preserving rank-ordered search," in Proceedings of the 2007 ACM workshop on Storage security and survivability. ACM, 2007, pp. 7–12.
[25]. S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+ r: Top- k retrieval from a confidential index," in Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology. ACM, 2009, pp. 439–449.
[26]. C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," Parallel and Distributed Systems, IEEE Transactions on, vol. 23, no. 8, pp. 1467–1479, 2012.
[27]. N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data,"in IEEE INFOCOM, April 2011, pp. 829–837.
[28]. W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Privacy-preserving multi-keyword text search in the cloud sup- porting similarity-based ranking," in Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ACM, 2013, pp. 71–82.
[29]. C.Orencik, M.Kantarcioglu, and E.Savas,"A practical and secure multi-keyword search method over encrypted cloud data," in Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on. IEEE, 2013, pp. 390–397.