Journal of Engineering Sciences
Vol 10, Issue 9, Sept /2019
ISSN NO:0377-9254

EFFICIENT KEYWORD SEARCH OVER ENCRYPTED CLOUD DATA

[1]G.SITA KALYANI, [2]Dr.B.HARI BABU
[1]M.Tech Student, [2]Professor
Department of CSE
KAKINADA INSTITUTE OF TECHNOLOGICAL SCIENCES (KITS), RAMACHANDRAPURAM

Abstract: With the advent of cloud computing, most of the data owners are outsourcing their complex data management systems from local sites to the commercial public cloud for great flexibility and economic savings. But sensitive data has to be encrypted before outsourcing, for protecting data privacy. However data encryption makes effective data utilization a challenging task. Traditional data utilization based keyword search on encrypted data is a difficult task. Thus, enabling an encrypted cloud data search service is of paramount importance. Considering the large number of data users and documents in the cloud, it is necessary to allow keyword search request and return documents in the order of their relevance to these keyword. In this paper we proposed a system that supports multi owner keyword ranked search over the encrypted cloud data with good key management scheme. Thorough security and performance analysis show that the proposed scheme guarantees high security and practical efficiency.

Index Terms—Searchable encryption, cloud computing, expressiveness, attribute-based encryption.

1 INTRODUCTION

Consider a cloud-based healthcare information system that hosts outsourced personal health records (PHRs) from various healthcare providers. The PHRs are encrypted in order to comply with privacy regulations like HIPAA. In order to facilitate data use and sharing, it is highly desirable to have a searchable encryption (SE) scheme which allows the cloud service provider to search over encrypted PHRs on behalf of the authorized users (such as medical researchers or doctors) without learning information about the underlying plaintext. Note that the context we are considering supports private data sharing among multiple data providers and multiple data users. Therefore, SE schemes in the private-key setting [1], [2], [3], which assume that a single user who searches and retrieves his/her own data, are not suitable. On the other hand, private information retrieval (PIR) protocols [4], [5], [6], which allow users to retrieve a certain data-item from a database which publicly stores data without revealing the data-item to the database administrator, are also not suitable, since they require the data to be publicly available. In order to tackle the keyword search problem in the cloud-based healthcare information system scenario, we resort to public-key encryption with keyword search (PEKS) schemes, which is firstly proposed in [7]. In a PEKS scheme, a ciphertext of the keywords called "PEKS ciphertext" is appended to an encrypted PHR. To retrieve all the encrypted PHRs containing a keyword, say "Diabetes", a user sends a "trapdoor" associated with a search query on the keyword "Diabetes" to the cloud service provider, which selects all the encrypted PHRs containing the keyword "Diabetes" and returns them to the user while without learning the underlying PHRs. However, the solution in [7] as well as other existing PEKS schemes which improve on [7] only support equality queries [8].

Set intersection and meta keywords1 [9], [10] can be used for conjunctive keyword search. However, the approach based on set intersection leaks extra information to the cloud server beyond the results of the conjunctive query, whilst the approach using meta keywords require 2 m meta keywords to accommodate all the possible conjunctive queries for m keywords. In order to address the above deficiencies in conjunctive keyword search, schemes such as the ones in [11], [12] were put forward in the public-key setting.

Ideally, in the practical applications, search predicates (i.e., policies) should be expressive such that they can be expressed as conjunction, disjunction or any Boolean formulas2 of keywords. In the above cloud-based healthcare system, to find the relationship between diabetes and age or weight, a medical researcher may issue a search query with an access structure (i.e., predicate) ("Illness = Diabetes" AND ("Age = 30" OR "Weight = 150-200")). SE schemes supporting expressive keyword access structures were presented in [8], [13], [14], [15]. Unfortunately, the scheme in [13] has exponentially increasing complexity [16], while the schemes in [8], [14], [15] are based on the inefficient bilinear pairing over composite-order groups [17]. Though there exist techniques [17] to convert pairing-based schemes from composite-order groups to prime-order groups, there is still a significant performance degradation due to the required size of the special vectors [18].

In this paper, we propose a public-key based expressive SE scheme in prime-order groups, which is especially suitable for keyword search over encrypted data in scenarios of multiple data owners and multiple data users such as the

cloud-based healthcare information system that hosts outsourced PHRs from various healthcare providers.

1.1 Overview of Our Proposed Scheme

Our expressive SE scheme consists of a trusted trapdoor generation center which publishes a public system parameter and keeps a master key in secret, a cloud server which stores and searches encrypted data on behalf of data users, multiple data owners who upload encrypted data to the cloud, and multiple data users who would like to retreive encrypted data containing certain keywords. To outsource an encrypted document to the cloud, a data owner appends the encrypted document with keywords encrypted under the public parameter and uploads the combined encrypted document and encrypted keywords to the cloud. To retrieve all the encrypted documents containing keywords satisfying a certain access structure (i.e., predicate or policy) such as ("Illness = Diabetes" AND ("Age = 30" OR "Weight = 150-200")), a data user first obtains a trapdoor associated with the access structure from the trapdoor generation center and then sends the trapdoor to the cloud server. The latter will conduct the search and return the corresponding encrypted documents to the data user.

The basic idea of our scheme is to modify a key-policy attributed-based encryption (KP-ABE) scheme constructed from bilinear pairing over prime-order groups. Without loss of generality, we will use the large universe KP-ABE scheme selectively secure in the standard model proposed by Rouselakis and Waters in [18] to illustrate our construction during the rest of the paper. In KP-ABE, a ciphertext is computed with respect to a set of attributes and an access policy is encoded into a user's private key. A ciphertext can be decrypted by a private key only if the set of attributes associated with the ciphertext satisfies the access policy associated with the private key. Access policies in [18] can be very expressive, supporting any monotonic Boolean formulas. At first sight, a KP-ABE scheme can be transformed to an expressive SE scheme by treating attributes as keywords to be searched, by directly transforming the key generation algorithm on attribute access structures to a trapdoor generation algorithm on keyword search predicates, and by using the decryption algorithm to test whether keywords in a ciphertext satisfy the predicate in a trapdoor. However, KPABE schemes (e.g., [18], [19]) are not designed to preserve privacy of attributes (keywords) associated with ciphertexts. Specifically, given the public parameter and a ciphertext, the attributes (keywords) in the ciphertext can be discerned by anyone. In the following, to keep our description compact and consistent, we will use access structure, policy and predicate interchangeably.

In order to hide keywords in a ciphertext, inspired by the "linear splitting" technique in [20], we firstly split ciphertext components corresponding to every keyword into two randomized complementary components. Thus, even though the ciphertext still contains information about the keywords, this information is computationally infeasible to obtain from the public parameter and the ciphertext. We secondly re-randomize trapdoor components corresponding to every keyword associated with an access structure to match the splitted components in the ciphertext.

In addition to hiding keywords in ciphertexts, we also need to preserve keyword privacy in a trapdoor which contains an access structure as a component. First, to preserve keyword privacy in an access structure, we adopt the method in [21] to divide each keyword into a generic name and a keyword value. Since keyword values are much more sensitive than the generic keyword names, the keyword values in an access structure are not disclosed to the cloud server, whereas a partial hidden access structure with only generic keyword names is included in a trapdoor and sent to the cloud server. Take the aforementioned keyword access structure ("Illness = Diabetes" AND ("Age = 30" OR "Weight = 150-200")) as an instance, "Illness", "Age" and "Weight" are the generic names whilst "Diabetes", "30" and "200" are the keyword values. Consequently, the partial hidden access structure ("Illness" AND "Age" OR "Weight") is included in the trapdoor. Second, as in all the PEKS schemes, trapdoors are subject to the offline keyword dictionary guessing attacks. That is, anyone who knows a trapdoor and the public parameter may discover the keyword values embedded in the trapdoor by launching exhaustive searching attacks on keyword values. As a remedy to such attacks, we assign a designated cloud server as introduced in [22] to perform the searching operations. We equip this designated server with a public and private key pair of which the public key will be used in trapdoor generation such that it is computationally infeasible for anyone without knowledge of the privacy key to derive keywords information from the trapdoor. Thus, trapdoors can be delivered to the cloud server over a public channel.

We define a security model for expressive SE, which takes into account all adversarial capabilities of the standard SE security notion. The adversary is able to learn trapdoors over access structures of its choice, but it should not be able to learn any information about the keyword values in the challenge ciphertext. Note that since the Rouselakis-Waters KP-ABE scheme [18], which the proposed SE scheme is built upon, is selectively secure, our expressive SE scheme can only be proved to be selectively secure where the adversary has to commit the challenge keyword set in advance.

1.2 Contributions

Below we briefly summarize our contributions in this paper.

• We propose the first expressive SE scheme in the public-key setting from bilinear pairings in prime order groups. As such, our scheme is not only capable of expressive multi-keyword search, but also significantly more efficient than existing schemes built in composite-order groups.

• Using a randomness splitting technique, our scheme achieves security against offline keyword dictionary guessing attacks to the ciphertexts. Moreover, to preserve the privacy of keywords against offline keyword dictionary guessing attacks to trapdoors, we divide each keyword into keyword name and keyword value and assign a designated cloud server to conduct search operations in our construction.

• We formalize the security definition of expressive SE, and formally prove that our proposed expressive SE scheme is selectively secure in the standard model.

• We implement our scheme using a rapidly prototyping tool called Charm, and conduct extensive experiments to evaluate its performance. Our results confirm that the proposed scheme is sufficiently efficient to be applied in practice.

2 .RELATED WORK

Traditional searchable encryption [11, 12, 13, 14, 15, 16, 17, 18, 19] has been widely studied in the context of cryptography. Among those works, most are focused on efficiency improvements and security definition formalizations. The first construction of searchable encryption was proposed by Song et al. [12], in which each word in the document is encrypted independently under a special two-layered encryption construction. Goh [13] proposed to use Bloom filters to construct the indexes for the data files. For each file, a Bloom filter containing trapdoors of all unique words is built up and stored on the server. To search for a word, the user generates the search request by computing the trapdoor of the word and sends it to the server. Upon receiving the request, the server tests if any Bloom filter contains the trapdoor of the query word and returns the corresponding file identifiers. To achieve more efficient search, Chang et al. [16] and Curtmola et al. [17] both proposed similar "index" approaches, where a single encrypted hash table index is built for the entire file collection. In the index table, each entry consists of the trapdoor of a keyword and an encrypted set of file identifiers whose corresponding data files contain the keyword. As a complementary approach, Boneh et al. [14] presented a public-key based searchable encryption scheme, with an analogous scenario to that of [12]. In their construction, anyone with the public key can write to the data stored on the server but only authorized users with the private key can search. As an attempt to enrich query predicates, conjunctive keyword search, subset query and range query over encrypted data have also been proposed in [18, 20]. Note that all these existing schemes support only exact keyword search, and thus are not suitable for cloud computing.

Private matching [21], as another related notion, has been studied mostly in the context of secure multiparty computation to let different parties compute some function of their own data collaboratively without revealing their data to the others. These functions could be intersection or approximate private matching of two sets, etc. [22]. Private information retrieval [23] is an often-used technique to retrieve the matching items secretly, which has been widely applied in information retrieval from database and usually incurs unexpectedly computation complexity.

3. CHALLENGES OF SEARCHABLE ENCRYPTION

The original goal of searchable encryption is to provide privacy-preserving keyword searches of encrypted data against an intermediate gateway such as a mail server or a network router, A. B. Lewko et al.[4][5] [6]involves a message exchange process between the sender and the receiver. The searchable encryption scheme that enables keyword search over data encrypted with different keys. The scheme is practical and was designed to be included in a new system for protecting data confidentiality in client server applications against attacks on the server. we discuss about the architecture and security requirements for searchable encryption scheme.

3.1 Searchable encryption Architecture:

Searchable encryption (SE) enables the users to generate a search token from the searched keyword in such way that given a token, the cloud server can retrieve the encrypted contents containing the searched keyword. Basically, the search token represents an encrypted query over the encrypted data and can be generated only by users with the appropriate secret key. Fig. 1 shows the basic architecture and working principle of a searchable encryption scheme. The architecture comprises mainly four entities: data owner, data user, cloud service provider and key generator. A brief description of the entities and their operations are given below.

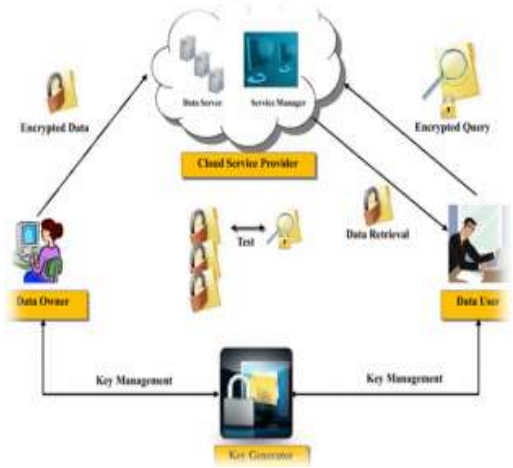


Figure: 1 Architecture of a searchable encryption scheme.

• **Data owner:** The data owner is the entity which generates and encrypts the data and uploads them to the cloud server. It can be either an organization or an individual. To use the service, the data owner uses its application which consists of a data processor for uploading new contents to the cloud. It encrypts the data and metadata with a cryptographic scheme that enables searching capability.

• **Data user:** This entity is also a subscriber to the cloud storage which sends encrypted queries to the cloud service provider to search for a specific encrypted data. There may be more than one data user in the system and in some scenario, the data owner and the data user might be the same entity.

• **Cloud service provider**: This entity provides the data storage and retrieval service to the subscribers. The cloud service provider consists of cloud data server and cloud service manager.

The first entity is used to store the outsourced encrypted data whereas the latter one is used for data management in the cloud. Upon receiving the encrypted search queries from the data user, the cloud service provider tests on the encrypted queries and encrypted metadata in the cloud storage. The encrypted data that satisfies the search criteria is retrieved and sent back to the data owner upon completion of the test. The cloud service provider should not learn any information from the operation

• **Key generator**: This entity is considered to be a trusted third party which is responsible for the generation and management of the encryption/ decryption keys. User specific keys are generated and distributed during the setup of the system.

3.2 Searchable Encryption Security Requirements

In general, the following requirements should be satisfied when constructing a searchable encryption scheme.

• **Retrieved data**: Server should not be able to distinguish between documents and determine search contents.

• **Search query**: Server should not learn anything about the keyword being searched for. Given a token, the server can retrieve nothing other than pointers to the encrypted content that contains the keyword.

• **Query generation**: Server should not be able to generate a coded query. The query can be generated by only those users with the relevant secret key.

• **Search query outcome**: Server should not learn anything about the contents of the search outcome.

• **Access patterns**: Server should not learn about the sequences and frequency of documents accessed by the user.

• **Query patterns**: Server should not learn whether two tokens were intended for the same query.

4. Results and Analysis

4.1. Snapshot of File Upload Page

The upload page shown in figure 2, contains two text boxes, in the first box user has to enter the words for building index and in the 2nd text box, user has to enter priority word if the file contains more word related information but the occurrence of that word in the file is less.

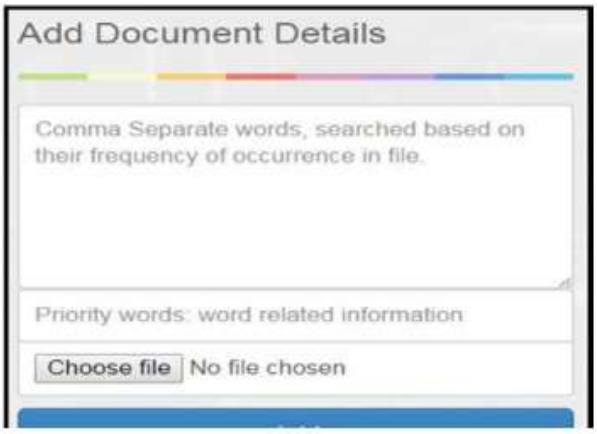


Figure 2. File Upload page.

Figure 3. Output of Search Request.

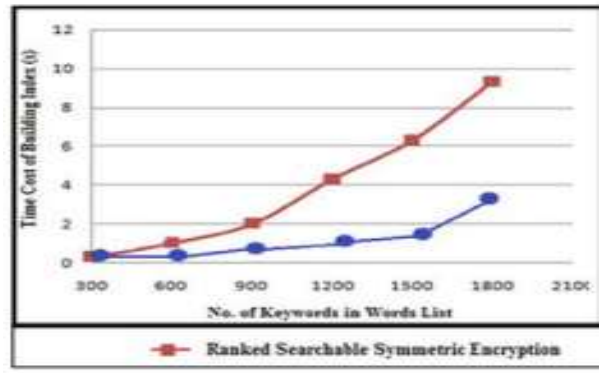
4.2. Snapshot of Search Result Returned to the User.

Figure 3, shows the files returned to the user as search result. From the figure we can see that, the user want to retrieve files containing information related to keyword "search" send the request, files returned as search result are in rank order according to their relevance with respect to the word specified by the requesting user. In the above figure, four files are returned to the requesting user, first file displayed with golden border is priority file, and this file is displayed first because it contains more information related to the word "search" even if the number of occurrence of the word is only 41. Files displayed after priority are displayed based on frequency of occurrence of the word in the file. There are three files displayed after the priority file, the file named "ieee.pdf" with 118 frequency of occurrence is displayed before files keyword_search.pdf and secure_data.pdf with frequency 51 and 1 respectively.

Performance of Building Index: above figure 4, shows time cost of building index in Ranked Searchable Symmetric Encryption (RSSE) and in our presented scheme. From the figure we can see that time cost of building index increases with increase in the size of word list and increase in size of dataset.

In Ranked Searchable Symmetric Encryption (RSSE) scheme, complete file collection is scanned and a list of keywords is selected for building searchable index for complete file collection. So RSSE scheme takes much computation time to build the index. Also whenever a new file is added to the file collection, updating the index will become an overhead

In our system, at the time of file upload, we are asking the data owner to enter the keywords that the file is relevant to and Index is built for these keywords only for that particular file, thus reducing the size of word-list and size of dataset for each word. Reduction in size of both word-list and dataset results in reduction in time cost of building index. And there will not be much overhead in updating the index when a new file is added to the file collection.



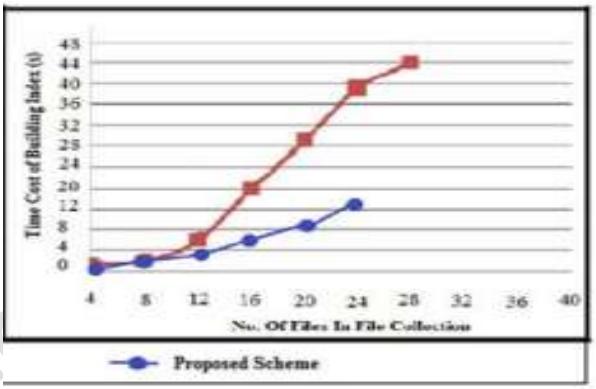


Figure 4. Time Cost of Building Index.

5. CONCLUSION

Data security is very important in cloud computing. Encrypting data for security makes effective data utilization a challenging task. To achieve efficient data retrieval from encrypted data collection, we are using concept of ranked searchable symmetric encryption model with some improvements and presented a system which will efficiently retrieve files containing information related to specified keyword in rank order from an encrypted file collection, i.e topmost files contain information more relevant to the word than other files. Our presented system support multiple data owners to upload files, provide good key management scheme with improved performance of building index. Our system also considers the files which contain more word related information but less occurrence of the word in that file, by assigning priority to the file with respect to the word. The result analysis show that the application developed support the secure and efficient data retrieval.

References

[1]. Cong Wang, Ning Cao, Kui Ren and Wenjing Lou.Enabling Secure and Efficient Keyword Search over

Outsourced Cloud Data.IEEE Transaction on Parallel and Distributed Systems, VOL. 23, NO. 8;August 2012.

[2]. D. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data.in Proc. of IEEE Symposium on Security and Privacy;2000, pp. 44-55.

[3]. Secure Indexes for Searching Efficiently on Encrypted Compressed Data E.-J. Goh. Technical Report 2003/216, Cryptology ePrint Archive, http://eprint.iacr.org/,2003.

[4] P. Mell and T. Grance, "Draft nist working definition of cloud computing," Referenced on Jan. 23rd, 2010 Online at http://csrc.nist.gov/ groups/SNS/cloud-computing/index.html, 2010.

[5] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep. UCB-EECS-2009-28, Feb 2009.

[6] S. Kamara and K. Lauter, "Cryptographic cloud storage," in Proceedings of Financial Cryptography: Workshop on Real-Life Cryptographic Protocols and Standardization 2010, January 2010.

[7] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing," 2009, http://www.cloudsecurityalliance.org.

[8] I. H. Witten, A. Moffat, and T. C. Bell, "Managing gigabytes: Compressing and indexing documents and images," Morgan Kaufmann Publishing, San Francisco, May 1999.

[9] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. of IEEE Symposium on Security and Privacy'00, 2000.

[10] Wei Zhang, Yaping Lin, Sheng Xiao, Jie Wu, Siwang Zhou, "Privacy Preserving Ranked Multi-Keyword Search for Multiple Data Owners in Cloud Computing", IEEE Transactions on Computers, Vol. 65, No. 5, May 2016.

[11] Kalyani Sonawane, Rahul Dagade, "A Survey on Multi-Keyword Ranked Search over Encrypted Cloud Data with Multiple Data Owners", International Journal of Computer Applications(0975-8887), Volume 162 No 11, March 2017.

[12] Ming Li, Shucheng Yu, Ning Cao, Wenjing Lou, "Authorized Private Keyword Search over Encrypted Data in Cloud Computing", 31st International Conference on Distributed Computing Systems, 2011.

[13] Cong Wang, Sherman S.M. Chow, Qian Wang, Kui Ren, Wenjing Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage", IEEE Transactions on Computers, Vol. 62, No. 2, February 2013.

[14] Wenhai Sun, Bing Wang, Ning Cao, Ming Li, Wenjing Lou, Y. Thomas Hou, Hui Li, "Verifiable Privacy-Preserving MultiKeyword Text Search in the Cloud Supporting Similarity-Based Ranking", IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 11, November 2014.

[15] Ning Cao, Cong Wang, Ming Li, Kui Ren, Wenjing Lou, "Privacy Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data", IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 1, January 2014.

[16] Zhangjie Fu, Xingming Sun, Zhihua Xia, Lu Zhou, Jiangang Shu, "Multi-keyword Ranked Search Supporting Synonym Query over Encrypted Data in Cloud Computing", 2013.

[17] Zhangjie Fu, Xingming Sun, Nigel Linge, Lu Zhou, "Achieving Effective Cloud Search Services: Multi-keyword Ranked Search over Encrypted Cloud Data Supporting Synonym Query", IEEE Transactions on Consumer Electronics, Vol. 60, No. 1, February 2014.