ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

Efficient and Expressive Keyword Search over Encrypted Data in the Cloud

[1] Balne Sridevi, [2] Siddi Sravani
[1][2] Asst. Professor, CSE Dept., Balaji Institute of Technology and Science

Abstract - **In today's world, there are many new challenges for the security of data and access control when users outsource sensitive data for sharing on third party server known as cloud servers, which are not within the same trusted domain as data owners. The existing technique used to maintain the confidentiality of personal medical record (PMR) against untrusted servers by disclosing data decryption keys only to authorized users. However, in doing so, these solutions inevitably introduce complexity in key management also burden on the data owner in data management well as in key management. The problem of simultaneously achieving security and data confidentiality and finegrainedness of access control still remains unresolved. This paper addresses this challenge 1) Key management, 2) Defining and enforcing access policies based on data attributes, and, 3) Keyword search over the encrypted data. PMR(patient medical record)system users need to deal with complicated key management problem to accomplish fine-grained access control when their PMRs are encrypted using symmetric key cryptography or asymmetric key cryptography. With our scheme multi-authority attribute based access control (MAABAC) we can reduce the key management complexity for owners and users. For this users are divided into the two domains; professional domain and personal domain. To achieve security of PMR, key management, user revocation and efficient keyword search exploiting KP-ABE, Multi-authority attribute based access control(MA-ABAC), and uniquely combining it with techniques of proxy re-encryption.**

Keywords: Attribute based encryption, Cloud computing, Fine-grained access control, KP-ABE, MA-ABAC, User Revocation, Proxy Re-encryption.

INTRODUCTION

Much like the popularity of portable personal electronic devices, cloud storage service has been booming over the last decade. Its outstanding advantages, such as considerable storage space, flexible accessibility and convenient data retrieval, strongly catch the attention of Internet users. Accordingly, to date not only individuals but also industries prefer to remotely store their data to cloud servers, such that they can get rid of the burden of local data management and maintenance. This makes cloud storage service share a great piece of market cut in the field of data management even in the ear of big data. Remotely data storage delivers convenience to Internet users and meanwhile, brings security concerns. The fact that users cannot have full physical possession of their data immediately rises up two serious practical questions: how to guarantee the confidentiality of the data, and how to retrieve the data. For the first question, we usually tackle it by leveraging existing encryption cryptographic mechanisms, such that all outsourced data are encrypted and inaccessible to cloud servers. The encryption technology, with no doubt, enables us to protect the confidentiality of the data. However, it limits the flexibility of data retrieve to some extent. The premise of encryption technique is to prevent a cipher text holder from gaining access to the underlying knowledge of data. Without any knowledge related to the data, it looks impossible for a cloud server to fulfil any data retrieval task. A naive solution here for data retrieval is to allow the server to fully access the data, allocate the data and next return it to user. Nevertheless, this disgraces the meaning of encryption. To support data retrieval without loss of confidentiality, Searchable Encryption (SE) mechanisms (e.g. [27], [9]) have been proposed in the literature. SE has been studied and widely employed in real-world applications where data search is outsourced to untrusted cloud servers. SE allows a server to search in encrypted data on behalf of a data owner without accessing the information of the data and search query contents. In an SE, a user encrypts a file database and its search keywords, and next uploads them to a cloud server. When retrieving a file, the user delivers a token related to the keyword to the server so that the server then locates the corresponding encrypted file from the encrypted database. The flexibility and scalability of a SE system mainly depend on how we design search token as well as search keyword. From practical point of view, a more expressive search query yields a more precise data retrieval. We take an Electronic Health Records (EHRs) search as an example. In a EHRs system, a patient's medical record is usually encrypted and stored in a storage system. We suppose there is a patient Alice's encrypted record which is tagged with a keyword index "Alice". To search the medical record of Alice from its storage system, a hospital needs to find a file matching the keyword "Alice". However, "Alice", the search index, is quite common in usual. There are probably 10,000 patients associated with the same keyword. This definitely

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

increases the workload of the hospital to locate the real "Alice" file they need from the rest of other encrypted records (with the same keyword). To enhance the search expressiveness, one may replace a single keyword index with access formula, such as ("Alice" AND "1990" AND "Crystal Lake") or ("Alice" AND "Age < 20" AND "Student ∈ NYU"). Actually, the most powerful expressive way to represent a search query is to leverage regular language. Using regular language to describe a data to be encrypted is extremely common in daily life. For instance, a Facebook user may directly write down a description, e.g., "my birthday party with best friends Bob and Kate", for an uploaded photo. Furthermore, suppose a tax form is encrypted and archived in some tax authority. The authority may need to search one of the tax forms based on an exact sentence or paragraph of the tax form, such as "Alice have paid $ 8,000 tax in total in 2014", in which the number is encrypted. Some more recent applications for regular language search are online genetic relatedness test and chemical compound search. Suppose a language space only contains "A,G,C,T", a search queried may upload a masked search pattern "ACGGTTCT" to an encrypted genetic database to request the server to return all possible matching encrypted DNA sequences. Unfortunately, there is no SE supporting regular language search in the literature. Designing flexible and scalable regular language search without loss of data by confidentiality and query privacy that becomes the main motivation of our work. Searchable Symmetric Encryption (SSE) and Public key Encryption with Keyword Search (PEKS) are two types of SE. SSE generally enjoys better search efficiency than that of PEKS. It provides a limited level of expressiveness for search. It is not difficult to see that the limitation of expressiveness actually inherits from some original limitation design in symmetric encryption1, such that it is difficult for SSE to support expressive search query (e.g. formula search, subset queries). Therefore, we deal with the e case of PEKS to achieve more search expressiveness in this paper.

II. KEY POLICY ATTRIBUTE-BASED ENCRYPTION (KPABE)

Data encryption is the most effective in regard to preventing sensitive data from unauthorized access. In earlier public key encryption or identity-based encryption systems, encrypted data is targeted for decryption by a single known user. To address these emerging needs, Sahai and Waters [4] introduced the concept of attribute-based encryption (ABE). As an alternative of encrypting to individual users, in ABE system, one can embed an access policy into the cipher- text or decryption key. Hence, data access is self-enforcing from the cryptography, needing no trusted mediator. ABE can be viewed as an extension of the notion of identity-based encryption in which user identity is generalized to a set of expressive attributes instead of a single string specifying the user identity. Compared with identity-based encryption ABE has significant advantage that it achieves flexible one-to many encryption as a substitute of one-to-one; it is envisioned as a promising tool for addressing the problem of secure and fine-grained data sharing and decentralized access control. There are two types of ABE depending on which of private keys or cipher texts that access policies are associated with. KP-ABE is a public key cryptography primitive for one-to many communications. In KP-ABE, files are associated with attributes for each of which a public key component is defined [5]. The encrypt or associates the set of attributes to the message by encrypting it with the corresponding public key components. For each user an access structure is assigned, which is usually defined as an access tree over data attributes, i.e., inner nodes of the access tree are threshold gates and leaf nodes are associated with attributes. User secret key is defined to reflect the access structure so that the user is able to decrypt a cipher text if and only if the data attributes satisfy his access structure KP-ABE schemes are suitable for structured organizations with rules about who may read particular documents. In a cipher text-policy attribute-based encryption (CP-ABE) system [9], when a sender encrypts a message, they specify a specific access policy in terms of access structure over attributes in the cipher text, stating what kind of receivers will be able to decrypt the cipher text. Users possess sets of attributes and obtain corresponding secret attribute keys from the attribute authority. Such a user can decrypt a cipher- text if his/her attributes satisfy the access policy associated with the cipher text. Thus, CP-ABE mechanism is conceptually closer to earlier role-based access control method [18].

III PROXY RE-ENCRYPTION

A basic goal of public-key encryption is to allow only the key or keys selected at the time of encryption to decrypt the cipher text or change the cipher text to a different key needs encryption of the message with the new key, which gives access to the original clear text and to a reliable copy of the new encryption key. This seems a fundamental, and quite desirable, property of good cryptography; it should not be possible to change the key with which a message can be decrypted by an untrusted party. Here, on the other hand [1] Proxy Encryption

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

(PRE) is a cryptographic primitive in which a semi-trusted proxy is able to convert a cipher text encrypted under A's public key into another cipher text that can be opened by B's private key without seeing the underlying plaintext. A Proxy Re-Encryption scheme allows the proxy, given the proxy re-encryption key rpk, to translate cipher texts under public key pk an into cipher texts under public key pkb and vice versa [10].

A. ATOMIC PROXY CRYPTOGRAPHY

A basic goal of public-key encryption is to allow only the key or keys selected at the time of encryption to decrypt the cipher text. To change the cipher text to a different key requires encryption of the message with the new key, which implies access to the original text and to a reliable copy of the new encryption key. An atomic proxy function allows an untrusted party to convert cipher text between keys without access to either the original file or to the secret component of the old key or the new key.

B USER REVOCATION

In case of user revocation, the data owner define updated tree structure and data file re-encryption. Whenever the data owner revoke user, the data owner first determines a minimal set of attributes without which the leaving user's access structure will never be satisfied. Next, he updates tree structure.

IV. SYSTEM ARCHITECTURE

In proposed system we divide the system users into the personal domain and professional domain. Personal domain users are like friends, family. Professional domain users are from different sectors healthcare, student, research etc. Owner encrypt PMR file and obtain secret key. Owner then again encrypt file by using different set of attributes with the particular access policy. For this we are using KP-ABE. While encrypting data in personal domain owner consider relation. If the user satisfies that relation then and then only he will able to access the file. In professional domain PMR file accessible to the user if he satisfies access policy given for the each attribute authority. Each attribute authority in system governs disjoint subset of user attributes. We are using MA-ABAC policies during encryption .Owner is free to set different policies. Owner can add/delete/modify the policy also they can dynamically change the policy. Our system also supports user revocation. User who wants file send a request as keyword to the cloud server they will get a file only when they satisfy the access policy set by the owner.

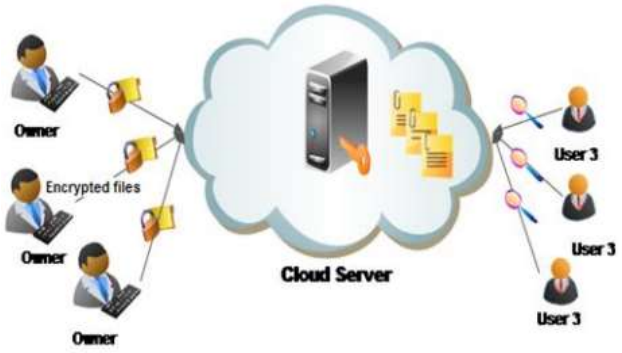

Fig.1. System Architecture for PMR Sharing

V METHODS

A. SYSTEM SETUP AND KEY DISTRIBUTION

System defines universe of data attributes for personal domain users and professional domain users. Each PMR owner generates its public /master keys. public keys published via user's profile in an social-network (HSN).User from personal domain send s a request to get PMR file .Owner sends specific secret key when user satisfy the access policy set by the corresponding owner .when request is from professional domain they will get secret key from attribute authority 3.2 PMR ENCRYPTION Owner outsource the encrypted PMR file to the cloud server. Each PMR file encrypted under the certain fine grained and attribute based access policy for users from professional domain and for the users from personal domain owner encrypt the file with attributes eg relation.

B. AUTHORIZED KEYWORD SEARCH AND ACCESS

Users from any domain search over the encrypted data. User send a request as a keyword to the cloud and will get file which contains that keyword ,only when user satisfy the access policy set by the owner. Only authorized users can decrypt the PMR file who have attribute based suitable key

C.USER REVOCATION

When user revoked the user will not get access to the file further. 3.4 POLICY UPDATES PMR owner can updates the access policy for existing PMR file. D. HANDLE DYNAMIC POLICY CHANGE

Our scheme should support the dynamic add/modify/delete of part of the document access policies or data attributes by the owner.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

VI. PRACTICAL ANALYSIS:

We leverage the Java Pairing Based Cryptography Library [23] to calculate the system running time shown in Table VI. Our testbed is: Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz, 3 GB RAM, Ubuntu 10.04. For the fairness of the practical comparison, we will use different pairing types - one is Type a with 160-bit group order (the embedding degree of the curve is 2) for the implementation of [31]'s KP-ABKS scheme; one is Type a1 with 1024 bits based field size and k = 2 for the implementation of [6]'s hidden vector encryption construction; and one is Typed with 159 bits based field and k = 6 for the implementation of our system (in which we assume a search token only has one final successful state). The above pairing types are chosen based on the recommendation introduced in [24], and all the data is without pre-processing. We suppose all schemes listed in the Tables must at least achieve a security level comparable to a symmetric key cryptosystem with an 80-bit key. That is, an elliptic curve cryptosystem with around 160-bit key is needed. Therefore, we set n = 160 bits, the group elements in Gξ1 are set to be 160 bits, and the group elements from GT and GT ,ξ2 are set to be 1024 bits, respectively, where ξ1 ∈ {1, 2, p, q} and ξ2 ∈ {p, q}. We further set the following four experimental samples: Test 1: l = 10; Test 2: l = 30; Test 3: l = 60; Test 4: l = 100. Table VII is the comparison of concrete communication cost.

VII. CONCLUSION

In the paper, we present a novel framework for data outsourcing and sharing on the hybrid cloud computing. It consists of a trusted private cloud and public cloud storage. In the framework, the storage server is able to perform search on encrypted data without learning the underlying plaintexts in the public key setting, X. Zhou [11] proposed a cryptographic primitive called public-key encryption with keyword search (PEKS). Since then, considering different requirements in practice, e.g., communication overhead, searching criteria and security enhancement, various kinds of searchable encryption systems have been put forth. However, there exist only a few public-key searchable encryption systems that support expressive keyword search policies, and they are all built from the inefficient composite-order groups. In this paper, we focused on the design and analysis of public-key searchable encryption systems in the prime-order groups that can be used to search multiple keywords in expressive searching formulas.

REFERENCES

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. J. Cryptology, 21(3):350–391, 2008.

[2] J. Baek, R. Safavi-Naini, and W. Susilo. On the integration of public key data encryption and public key encryption with keyword search. In ISC, vol. 4176 of LNCS, pp. 217–232. Springer, 2006.

[3] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In CRYPTO, vol. 4622 of LNCS, pp. 535–552. Springer, 2007.

[4] S. Benabbas, R. Gennaro, and Y. Vahlis. Verifiable delegation of computation over large datasets. In CRYPTO, vol. 6841 of LNCS, pp. 111–131. Springer, 2011.

[5] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In EUROCRYPT, vol. 3027 of LNCS, pp. 506–522. Springer, 2004.

[6]. Wang B, Yu S, Lou W, Hou T (2014) Privacy-Preserving Multi-Keyword Fuzzy Search over Encrypted Data in the Cloud. In: INFOCOM'14. IEEE, Piscataway, N.J, USA. pp 2112–2120

[7]. Cao N, Wang C, Li M, Ren K, Lou W (2014) Privacy-preserving multi-keyword ranked search over encrypted cloud data. In: IEEE Transactions on Parallel and Distributed Systems. IEEE, Piscataway, N.J, USA Vol. 25, no. 1. pp 222–233

[8] C. Bosch, Q. Tang, P. H. Hartel, and W. Jonker. Selective document ¨ retrieval from encrypted database. In ISC, vol. 7483 of LNCS, pp. 224– 241. Springer, 2012.

[9] J. Camenisch, M. Kohlweiss, A. Rial, and C. Sheedy. Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data. In PKC, vol. 5443 of LNCS, pp. 196–214. Springer, 2009.

[10] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou. Privacy-preserving multikeyword ranked search over encrypted cloud data. IEEE Trans. Parallel Distrib. Syst., 25(1):222–233, 2014.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

Balne Sridevi currently working as an Assistant Professor in CSE Department at BALAJI INSTITUTE OF TECHNOLOGY &SCIENCE, Narsampet, Warangal and has 13+ years of experience in Academic. Research areas include Information Security, Mobile and Cloud computing, Data Mining, Network Security etc.

Siddi Sravani currently working as an Assistant Professor in CSE Department at BALAJI INSTITUTE OF TECHNOLOGY & SCIENCE, Narsampet, Warangal and has 3+ years of experience in Academic. Research areas include Information Security, data mining etc…