JES
Journal of Engineering Sciences

Vol 10, Issue 9, Sept /2019
ISSN NO:0377-9254

EFECTIVE KEYWORD SEARCH OVER ENCODING DATA IN CLOUD

[1]Dr.K.RAMESHWARAIAH, [2]Mr. N. RAJENDER, [3]Mr.SR. RAMAKRISHNA CHARY
[1]Professor & HOD , [2]Associate Professor,[3]M.Tech Student
Department Of Computer Science And Engineering
Nalla Narasimha Reddy Education Society's Group of Institutions, Hyderabad

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

I. 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 extending from

[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., access policies or structures) 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 the composite-order groups [17]. Though there exist techniques [17] to convert pairingbased 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 the 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 retrieve encrypted data containing certain keywords. To outsource an encrypted document to the cloud, a data owner appends the encrypted document with encrypted keywords, and uploads the encrypted document and the corresponding encrypted keywords to the cloud. To retrieve all encrypted documents containing keywords satisfying a certain access structure 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 the 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 the ciphertexts. Specifically, given a ciphertext (without information on attributes), the attributes (keywords) in the ciphertext can be discerned by anyone using solely the public parameter. In the following, to keep our description compact and consistent, we will use access structure, access policy and search 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 be obtained 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 split 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.

II. LITERATURE SURVEY

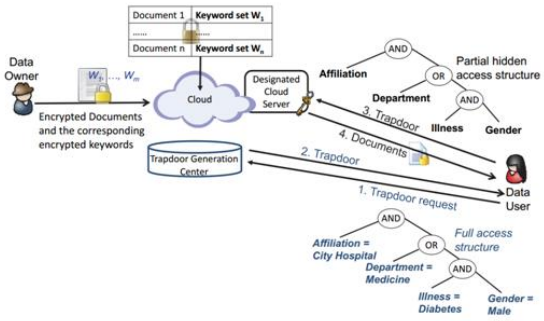
Software protection is one of the most important issues concerning computer practice. There exist many heuristics and ad-hoc methods for protection, but the problem as a whole has not received the theoretical treatment it deserves. In this paper, we provide theoretical treatment of software protection. We reduce the problem of software protection to the problem of efficient simulation on oblivious RAM.A machine is oblivious if thhe sequence in which it accesses memory locations is equivalent for any two inputs with the same running time. For example, an oblivious Turing Machine is one for which the movement of the heads on the tapes is identical for each computation. (Thus, the movement is independent of the actual input.) What is the slowdown in the running time of a machine, if it is required to be oblivious? In 1979, Pippenger and Fischer showed how a two-tape oblivious Turing Machine can simulate, on-line, a one-tape Turing Machine, with a logarithmic slowdown in the running time. We show an analogous result for the random-access machine (RAM) model of computation. In particular, we show how to do an on-line simulation of an arbitrary RAM by a probabilistic oblivious RAM with a polylogaithmic slowdown in the running time. On the other hand, we show that a logarithmic slowdown is a lower bound.

It is desirable to store data on data storage servers such as mail servers and file servers in encrypted form to reduce security and privacy risks. But this usually implies that one has to sacrifice functionality for security. For example, if a client

wishes to retrieve only documents containing certain words, it was not previously known how to let the data storage server perform the search and answer the query, without loss of data confidentiality. We describe our cryptographic schemes for the problem of searching on encrypted data and provide proofs of security for the resulting crypto systems. Our techniques have a number of crucial advantages. They are provably secure: they provide provable secrecy for encryption, in the sense that the untrusted server cannot learn anything about the plaintext when only given the ciphertext; they provide query isolation for searches, meaning that the untrusted server cannot learn anything more about the plaintext than the search result; they provide controlled searching, so that the untrusted server cannot search for an arbitrary word without the user's authorization; they also support hidden queries, so that the user may ask the untrusted server to search for a secret word without revealing the word to the server. The algorithms presented are simple, fast (for a document of length n, the encryption and search algorithms only need O(n) stream cipher and block cipher operations), and introduce almost no space and communication overhead, and hence are practical to use today.

We present a single-database computationally private information retrieval scheme with polylogarithmic communication complexity. Our construction is based on a new, but reasonable intractability assumption, which we call the Φ-Hiding Assumption (ΦHA): essentially the difficulty of deciding whether a small prime divides $\Phi(m)$, where m is a composite integer of unknown factorization.

III. SYSTEM DESIGN AND ANALYSIS



Architecture Diagram

The architecture of our SE system is shown in Fig. 1, which is composed of four entities: a trusted trapdoor generation centre who publishes the system parameter and holds a master private key and is responsible for the trapdoor generation, data owners who upload the encrypted data to a public cloud, data users who are privileged to search and access the encrypted data, and a designated cloud server who executes the keyword search operations for data users. To enable the cloud server to search over ciphertexts, the data owners append every encrypted document with encrypted keywords4 . A data user issues a trapdoor request by sending an access structure over keywords to the trapdoor generation centre which generates and returns a trapdoor corresponding to the access structure. We assume that the trapdoor generation centre has a separate authentication mechanism to verify each data user and then issue them the corresponding trapdoors. After obtaining a trapdoor, the data user sends the trapdoor with a corresponding partial hidden access structure (i.e., the access structure without keyword values) to the designated cloud server. The latter performs the testing operations between each ciphertext and the trapdoor using its private key, and forwards the matching ciphertexts to the data user.

IV. IMPLEMENTATION

MODULES:

- Data Owner
- Data User
- Cloud
- TGC

MODULES DESCRIPTION:

Data Owner:

In Data Owner module, Initially Data Owner must have to register their detail and TGC will authorize the registration by sending user id, name through email. After successful login data Owner can upload files into cloud server with File access policy. He/she can view the files that are uploaded in cloud.

Data User:

In Data user module, Initially Data user must have to register their detail and TGC will authorize the registration by sending user id, name through email. After successful login he/ can search all the files upload by data owners. He/she can send search request to TGC then TGC will send the trapdoor key. After entering the Trapdoor key he/she can view the file

Cloud:

In Cloud module, Cloud can view all the Data owners and data user's details. Cloud can able see the files in cloud uploaded by the data owners.

TGC:

In TGC module, TGC can view all the Data owners and data user's details. TGC will authorize data owners and data users. Also TGC will approve and send the Trapdoor key to the users. TGC can able see the files in cloud uploaded by the data owners.

V. DISCUSSION AND ANALYSIS

In this section, we discuss the properties as well as extensions of our expressive SE scheme.

5.1 Keyword Privacy

Keyword Value Guessing Attacks on Ciphertexts. We briefly review the encryption algorithm of the KP-ABE scheme in [18], and then show that there exists a keyword value guessing attack if it is directly transformed into a searchable encryption scheme. Let m denote the size of W, and W1, ..., Wm ∈ Zp be the specific values of W. It randomly chooses μ, z1, ..., zm ∈ Zp, and outputs a ciphertext CT = C, D, {(Ci , Di)}i∈[1,m].

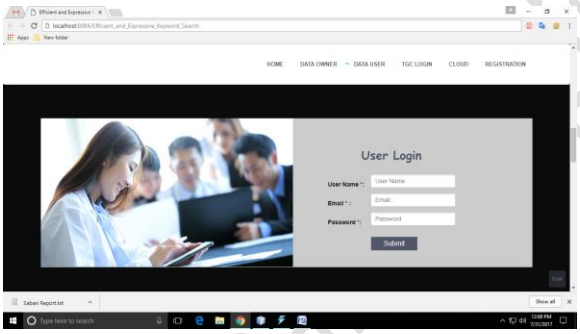
Keyword Value Guessing Attacks on Trapdoors. Concerning this security requirement, we need to tackle two problems in our construction. First, keywords associated with a trapdoor must be hidden from the access structure. We address this problem by separating each keyword into a generic name and a keyword value, i.e., each keyword is in the form of "generic name = keyword value", and a partial hidden access structure, i.e., the full access structure with keyword values being removed (See Fig. 1) is incorporated in a trapdoor and given to the designated cloud server. Second, the entire trapdoor should be immune to the offline keyword value guessing attacks [25]. In our SE system, we resort to a weaker security notion by requiring that a trapdoor will not disclose information about the keyword values in the ciphertext to an adversary excluding the cloud server who executes the searching operations. We assign a designated cloud server [22] to conduct search and equip it with a public and private key pair. Since the components in a trapdoor are tied with the public key of the server, only the designated cloud server with the corresponding private key is capable to learn the keyword values hidden in the trapdoor by performing offline guessing attacks.

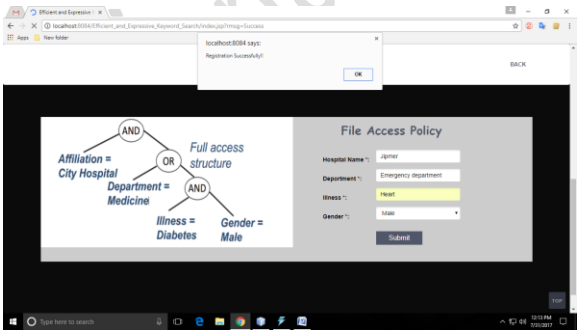
5.2 Extensions

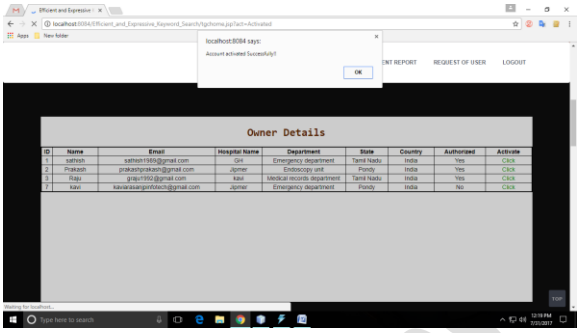
Our expressive SE scheme can be extended in several ways. • Expressive searchable encryption for range search. Range search is an important requirement for searchable encryption in many applications. By defining the keywords in a hierarchical manner as shown in [27], we can directly expand our SE system to support a class of simple range search [27]. Take a keyword name "Age" with keyword values from 0 to 100 as an example. The path of the leaf node "11-20" is ("0- 100", "0-30", "11-20"), and "0-30", "0-10" are simple ranges from level-2 and level-3, respectively. • Anonymous KP-ABE. Our SE system is built by making the Rouselakis-Waters KP-ABE scheme [18]

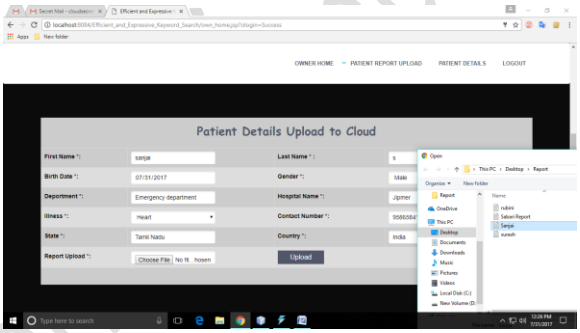
anonymous. Therefore, it can be easily extended to obtain an unbounded and anonymous KP-ABE scheme in the prime-order groups without random oracles, where given a ciphertext, an adversary learns no information on the associated attribute set. • Anonymous hierarchical identity-based encryption (HIBE). The Rouselakis-Waters KP-ABE scheme in [18] can be converted to an HIBE scheme using nonrepeating identities, "AND" policies and delegation capabilities [19]. Since our SE scheme can be used to construct an anonymous KP-ABE scheme, it can be further converted to an anonymous HIBE scheme using the same method as in [9].

VI.RESULTS









VII.CONCLUSION

In this 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 primeorder 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) PrivacyPreserving 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.