ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 6, June 2017

Dynamic Secret Key Generation for Multi Data User in Cloud Computing

[1] Chethan R, [2] Divya D, [3] Kavya H, [4] Prashanth [5] Vani Saptasagar,Associate professor ise department
R R institute of technology. RR layout, Chikkbanavara , Bangalore.

***Abstract*— CLOUD computing is a subversive technology that is changing the way IT hardware and software are designed and purchased. Cloud computing provides abundant benefits including easy access, decreased costs, quick deployment and flexible resource management, etc. It has become highly popular for data owners to outsource their data to public cloud servers while allowing data users to retrieve this data. For privacy concerns, a secure search over encrypted cloud data has motivated several research works under the single owner model. However, most cloud servers in practice do not just serve one owner; instead, they support multiple owners to share the benefits brought by cloud computing. In this paper, we propose Dynamic Secret key generation for multi user in cloud computing. To enable cloud servers to perform secure search without knowing the actual data of both keywords and trapdoors, we systematically construct a novel secure search protocol. To rank the search results and preserve the privacy of relevance scores between keywords and files, we propose a novel additive order and privacy preserving function family. To prevent the attackers from eavesdropping secret keys and pretending to be legal data users submitting searches, we propose a novel dynamic secret key generation protocol and a new data user authentication protocol.**

***Index Terms*— Cloud Computing, multiple users, dynamic secret key, ranked keyword search**

1 INTRODUCTION

Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data canters that provide those services.

The services themselves have long been referred to as Software as a Service (SaaS).a Some vendors use terms such as IaaS (Infrastructure as a Service) and PaaS (Platform as a Service) to describe their products, but we show these because accepted definitions for them still vary widely. The line between “low-level” infrastructure and a higherlevel “platform” is not crisp. We believe the two are more alike than different, and we consider them together. Similarly, the a For the purposes of this article, we use the term Software as a Service to mean applications delivered over the Internet. The broadest definition would encompass any on demand software, including those that run software locally but control use via remote software licensing.Related term “grid computing,” from the high-performance computing community, suggests protocols to offer shared computation and storage over long distances, but those protocols did not lead to a software environment that grew beyond its community. The data centre hardware and software is what we will call a cloud. When a cloud is made available in a pay-as you- go manner to the general public, we call it a public cloud; the service being sold is utility computing. We use the term private cloud to refer to internal data canters of a business or other organization, not made available to the general public, when they are large enough to benefit from the advantages of cloud computing that we discuss here. Thus, cloud computing is the sum of SaaS and utility computing, but does not include small or data canters, even if these rely on virtualization for management. People can be users or providers of SaaS, or users or providers of utility computing. We focus on SaaS providers (cloud users)and cloud providers, which have received less attention than SaaS users. The same actor can play multiple roles. For instance, a cloud provider might also host its own customer-facing services on cloud infrastructure.[1]

Despite the many benefits of cloud computing, for security concerns, individuals and organization users are reluctant to movie their sensitive data, including emails, personal health records and government confidential files, to the cloud. This is because once sensitive data are outsourced to a remote cloud, the corresponding data owners lose direct control of these data [2]. Cloud service providers(CSPs) would promise to ensure owners’ data security using mechanisms like virtualization and firewalls. However, these mechanisms do not protect owners’ data privacy from the CSP itself. Encryption on sensitive data before outsourcing can preserve data security against CSP. However, data encryption makes the traditional data usage service based on plaintext keyword search a very challenging problem. The only solution to this problem is to download all the encrypted data and decrypt them

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 6, June 2017

locally. However, this method is obviously improper because it will cause a huge amount of communication overhead. Therefore, developing a secure search service over encrypted cloud data is of paramount important. Secure search over encrypted data has recently attracted the interest of many researchers.[3]first define and solve the problem of secure search over encrypted data. They propose the concept of searchable encryption, which is a cryptographic primitive that enables users to perform a keyword-based search on an encrypted dataset, just as on a plaintext dataset. Searchable encryption is further developed by secure indexes, searchable systemic encryption, public key encryption, secure conjunctive. However, these schemes are concerned mostly with single or Boolean keyword search. Extending these techniques for ranked multi-keyword search will incur heavy computation and storage costs. Secure search over encrypted cloud data.These researches not only reduce the computation and storage cost for secure keyword search over encrypted cloud data, but also enrich the category of search function, including secure ranked multi-keyword search, fuzzy keyword search, and similarity search. However, all these schemes are limited to the single-owner model. As a matter of fact, most cloud servers in practice do not just serve one data owner; instead, they often support multiple data owners to share the benefits brought by cloud computing. In certain scenario, only the authorized organizations can perform a secure search over this encrypted data contributed by multiple data owners. Such a Personal Health Record sharing system, where multiple data owners are involved, can be found at mymedwall.com. Compared with the single-owner scheme, developing a full-fledged multi-owner scheme will have many new challenging problems. First, in the single-owner scheme, the data owner has to stay online to generate trapdoors (encrypted keywords) for data users. However, when a huge amount of data owners are involved, asking them to stay online simultaneously to generate trapdoors would seriously affect the flexibility and usability of the search system. Second, since none of us would be willing to share our secret keys with others, different data owners would prefer to use their own secret keys to encrypt their secret data. Consequently, it is very challenging to perform a secure, convenient, and efficient search over the data encrypted with different secret keys. Third, when multiple data owners are involved, we should ensure efficient user enrolment and revocation mechanisms, so that our system enjoys excellent security and scalability.

In this paper, we propose Dynamic key generation for search protocol in a privacy preserving ranked multi-keyword search protocol in a multi-owner cloud model. To enable cloud servers to perform secure search without knowing the actual value of both keywords and trapdoors, we systematically construct a novel secure search protocol. As a result, many different data owners use different keys to encrypt their files and keywords. Authenticated data users can issue a query without knowing secret keys of these different data owners. To rank the search results and preserve the privacy of relevance scores between keywords and files, we propose a new additive order and privacy preserving function family (AOPPF), which helps the cloud server return the most relevant search results to data users without revealing any sensitive information. To prevent the attackers from eavesdropping secret keys and pretending to be legal data users submitting searches, we propose a novel dynamic secret key generation protocol and a new data user authentication protocol. As a result, attackers who steal the secret key and perform illegal searches would be easily detected. Furthermore, when we want to revoke a data user, PRMSM ensures efficient data user revocation. Extensive experiments on real-world datasets confirm the efficacy and efficiency of our proposed schemes. The main contributions of this paper are listed as follows:

_ We define a multi-owner model for privacy preserving keyword search over encrypted cloud data.

_ We propose an efficient data user authentication protocol, which not only prevents attackers from eavesdropping secret keys and pretending to be illegal data users performing searches, but also enables data user authentication and revocation.

_ We systematically construct a novel secure search protocol, which not only enables the cloud server to perform secure ranked keyword search without knowing the actual data of both keywords and trapdoors, but also allows data owners to encrypt keywords with self-chosen keys and allows authenticated data users to query without knowing these keys.

_ We propose an additive order and privacy preserving function family which allows data owners to protect the privacy of relevance scores using different functions according to their preference, while still permitting the cloud server to rank the data files accurately.

_ We conduct extensive experiments on real-world datasets to confirm the efficacy and efficiency of our proposed schemes.

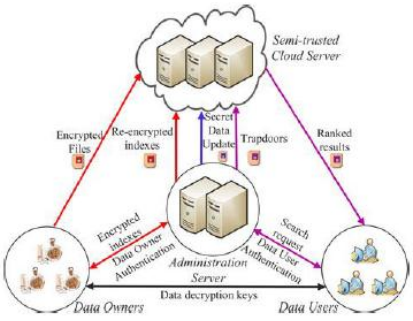


Fig. 1. Architecture of privacy preserving keyword search in a multi owner and multi-user cloud model.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 6, June 2017

2 PROBLEM FORMULATION

In further section we define System model and corresponding threat model

2.1 SYSTEM MODEL

In our multi-owner and user cloud computing model, four entities are involved, as illustrated in Fig. 1; they are data owners, the cloud server, administration server, and Data users.

DATA OWNER: Data owner have the set of files ,they create the index file ad send that file to the application server . Finally Data owner encrypt that file and send encrypted file to the cloud server .as a\well as send the encryption key to the data user .

CLOUD SERVER: Upon receiving the trapdoor, the cloud server searches the encrypted index of each data owner and returns the corresponding set of encrypted files.

ADMINISTRATION SERVER: Application server re-encrypt the index file of authenticated user and send that re-encrypted file to the cloud server

DATA USERS: Data user send keywords to search to words the application server, application server send that request to the cloud server if the data user are the authenticated user by creating the trapdoor

2.2 THREAT MODEL

In our threat model, we assume the administration server is trusted. The administrative server can be any trusted third party, e.g., the certificate authority in the public key infrastructure, the aggregation and Distribution layer and the third party auditor in [2]. Data owners and data users who passed the authentication of the administration server are also trusted.

2.3 GOALS AND SECURITY

To enable privacy preserving ranked multi keyword search in the multi-owner and multi-user cloud environment, our system design should simultaneously satisfy security and performance goals.

Ranked multi-keyword search over multiowner: The proposed scheme should allow multikeyword search over encrypted files which would be encrypted with different keys for different data owners. It also needs to allow the cloud server to rank the search results among different data owners and return the top-k results.

Data owner scalability: The proposed scheme should allow new data owners to enter this system without affecting other data owners or data users, i.e., the scheme should support data owner scalability in a plug-and-play model.

Data user revocation: The proposed scheme should ensure that only authenticated data users can perform correct searches. Moreover, once a data user is revoked, he can no longer perform correct searches over the encrypted cloud data.

Security goals: Keyword Semantic Security, Keyword secrecy, Relevance score secrecy.

3 DATA USER AUTHENTICATION PROTOCOL

To prevent attackers from pretending to be actual data users performing search and trying to access the data, so data user authentication before the server re-encrypts keywords for user Traditional user authentication are of three way First, user and data owner share a secret key Second, the user encrypts his personally identifiable information using and sends the encrypted data to the data owner.

Third, the data owner decrypts the received data with and authenticates the decrypted data. However, this method has two main drawbacks. First, since the secret key shared between the user and the data owner remains unchanged. Second, once the secret key is revealed to attackers, the data owner cannot distinguish between the legal user and the attackers; the attackers can pretend to be legal requesters without being detected. In this section, we first give an overview of the data user authentication protocol. Then, we introduce how to achieve secure and efficient data user authentication. Finally, we demonstrate how to detect illegal searches and how to enable secure and efficient data user revocation.

3.1 Overview

Now we give an example to illustrate the main idea of the user authentication protocol Assume Alice wants to be authenticated by the administration server, so she starts a conversation with the server. The server then authenticates the contents of the conversation. If the contents are authenticated, both Alice and the server will generate the initial secret key according to the conversation contents. After the initialization, to be authenticated successfully, Alice has to provide the historical data of their conversations. If the authentication is successful, both Alice and the administration server will change their secret keys according the contents of the conversation. In this way, the secret keys keep changing dynamically; without knowing the correct historical data, an attacker cannot start a successful conversation with the administration server.

3.2 User Authentication

Before we introduce the dynamic key generation method and the authentication protocol, we first introduce the format of the authentication data the authentication data consists of five parts. The request counter field records the number of search requests that the data user has submitted. The last request time field asks the data user to provide the historical data of his previous request time. The personally identifiable data (e.g., passport number, telephone number) field is used to identify a specific data user, while the random number and CRC field are further used to check whether the authentication

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 6, June 2017

data has been tampered with. The key point of a successful authentication is to provide both the dynamically changing secret keys and the historical data of the corresponding data user. Let ki;j denotes the secret key shared between administration server and the jth data user Uj after I instances of search requests, and di;j denotes the authentication data for the ði þ 1Þth request of Uj. Our authentication protocol runs in the following six steps.

A. Data user Uj prepares his authentication data di;j, i.e., Uj needs to fill in all the fields of authentication data based on his historical data.

B. Data user Uj encrypts di;j with the current secret key ki;j and submits the encrypted authentication data ðdi;jÞki;j to the administration server.

C. After submitting the authentication data, the data User Uj generates another secret key kiþ1;j ¼ ki;j _ Hðdi;jÞ, and stores both ki;j and kiþ1;j.

D.Upon receiving Uj's encrypted authentication data, the administration server decrypts it with ki;j.

E. The administration server checks the request counter, last request time, personally identifiable data and CRC, respectively. If the authentication succeeds, the administration server first generates a new secret key kiþ1;j ¼ ki;j _ Hðdi;jÞ, then he replies a confirmation data diþ1;j, and encrypts it with kiþ1;j. Otherwise, the administration server encrypts diþ1;j with secret key ki;j.

F. Upon receiving a reply from the administration server, the data user Uj will try to decrypt it with kiþ1;j. If the decrypted data contains the confirmation data, the authentication is successful. Otherwise, the authentication is regarded as being unsuccessful. The data user deletes the new generated secret key kiþ1;j and considers whether to start another authentication.

Fig. 3 shows an example of successful authentication between the administration server and the data user. As we can see, after each successful authentication process, the secret key will be changed dynamically according to the previous key and some historical data. Therefore, once an attacker steals a secret key, he can hardly get any benefits. On one hand, if the attacker knows nothing about the historical data of the legal data user, he cannot even construct a legal authentication data. On the other hand, if the legal data user performs another successful authentication, the previous secret key will be expired.

3.3 Illegal Search Detection

In our scheme, the authentication process is protected by the dynamic secret key and the historical information. We assume that an attacker has successfully eavesdropped the secret key k0;j of Uj. Then he has to construct the authentication data; if the attacker has not successfully eavesdropped the historical data, e.g., the request counter, the last request time, he cannot construct the correct authentication data. Therefore this illegal action will soon be detected by the administration server. Further, if the attacker has successfully eavesdropped all data of Uj, the attacker can correctly construct the authentication data and pretend himself to be Uj without being detected by the administration server. However, once the legal data user Uj performs his search, since the secret key on the administration server side has changed, there will be contradictory secret keys between the administration server and the legal data user. Therefore, the data user and administration server will soon detect this illegal action.

3.4 Data User Revocation

Different from previous works, data user revocation in our scheme does not need to reencrypt and update large amounts of data stored on the cloud server. Instead, the administration server only needs to update the secret data Sa stored on the cloud server. As will be detailed in the next section, Sa ¼ gka1_ka2_ra , where ka1 and ka2 are the secret keys of the administration server, and ra is randomly generated for every update operation. Consequently, the previous trapdoors will be expired. Additionally, without the help of the administration server, the revoked data user cannot generate the correct trapdoor Twh0 . Therefore, a data user cannot perform correct searches once he is revoked.

4. RELATED WORKS

4.1 FUZZY KEYWORD SEARCH OVER ENCRYPTED DATA IN CLOUD COMPUTING

With the evolution in Cloud Computing more and more sensitive data is being incorporated into the cloud. To ensure security and privacy these data are first encrypted before being uploaded onto the cloud servers thus making search a complicated task. Although in traditional cloud computing encryption searching schemes allows user to search encrypted data through keywords securely. These techniques employed exact keyword search and will fail if there are any morphological variants or spelling errors. This leads to low in efficiency and also affects system usability very badly. Fuzzy keyword search increases the system usability by allowing matching the exact or closet match text to the stored keywords and retrieving the approximate closest results. We shall be using edit distance to quantify keywords. We ensure the privacy of the data against unauthenticated users by encrypting the data using AES encryption before uploading to the cloud servers. We tend to resolve this problem by using a cloud server and employing fuzzy keyword search based on N grams. Thus efficiency of our proposed system would be demonstrated through experimental results. With the advancement in cloud computing, cloud servers are widely being used for storing data centrally. This includes various social accounts, game data, website login and more type of data. The cloud services provides relief to user as it reduces storage overheads and risk of losing the data due to hardware failures i.e. it might

ISSN (Online) 2394-2320

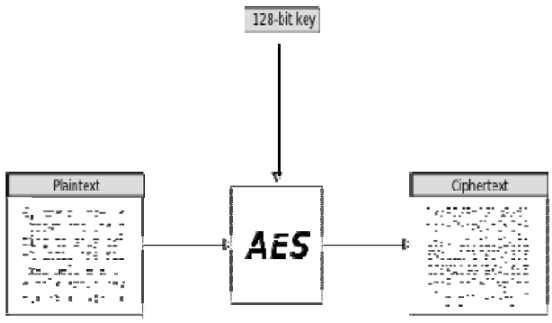
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 6, June 2017

happen the hard disk of our system or due to malicious activity and we would end up losing all the important data. The other problem may be poor maintenance and low configuration service as compared to cloud configuration services. On the other hand cloud also has some drawbacks because cloud servers cannot be trusted by the data owners so it is the user's responsibility to encrypt the data before upload. By implementing data encryption, there's overhead of data utilization in more efficient manner as the data is secured and cannot be accessed by unauthenticated users. Also, in cloud computing, data owners share their outsourced data with large number of users due to which privacy of the data is not ensured. Thus it is required that every individual should retrieve specific data files which they are looking for within a session. To apply this type of system we need to deal with keyword search that retrieve the required files instead of retrieving all the encrypted files [1].

We are implementing fuzzy keyword search over cloud without compromising the privacy of our data. By employing fuzzy keyword search the usability of our system is enhanced. Users can search their text with possible values and get the desired result when exact keyword match fails. This failure of exact keyword could be because of some spelling or morphological error. Thus fuzzy keyword search helps to overcome this and give desired results to the user. In our proposed system, edit distance technique to quantify keywords similarity by implementing the advanced algorithm technique for storing, matching and searching fuzzy keyword sets. These algorithms eliminate the need for storing all fuzzy keywords to improve efficiency in terms of privacy as well as overhead of storing large number of keywords by reducing the number of keywords which helps us to retrieve fast data and overhead of matching to all fuzzy keyword is reduced. We shall be implementing AES encryption algorithm before uploading our documents over the cloud servers. This is done to ensure secure and privacy of our data against unauthenticated users

Advanced Encryption Standard (AES)

AES is a block cipher technique with block size of 128 bits. It mainly allows three different key lengths: 128, 192 or 256 bits. We propose to encrypt our data by using AES with 128 bit key length. For 128-bit keys the encryption process consists of 10 rounds. In each case all the rounds are identical other than the last round. A 4 byte word is formed from 16 byte encryption key which is expanded into key schedule consisting of 44 4- byte words. A 4x4 matrix of bytes known as state array is obtained from 128-bit input block. The input state is XORed with first four words of the schedule before any round based processing could start [12].



Advantage:

- Thissystem provide more practical and effective fuzzy keyword search constructions with regard to both storage and searchefficiency.
- Storing their data into the cloud, the data owners can be relieved from the burden of data storage and maintenance soas to enjoy the ondemand high quality data storage service.

4.2 LT Codes-based Secure and Reliable Cloud Storage Service

With the increasing adoption of cloud computing for data storage, assuring data service reliability, in terms of data correctness and availability, has been outstanding. While redundancy can be added into the data for reliability, the problem becomes challenging in the "pay-as-you-use" cloud paradigm where we always want to efficiently resolve it for both corruption detection and data repair, we design a secure cloud storage service which addresses the reliability issue with near-optimal overall performance. By allowing a third party to perform the public integrity verification, data owners are significantly released from the heavy work of periodically checking data integrity. To completely free the data owner from the burden of being online after data outsourcing, this paper proposes an exact repair solution so that no metadata needs to be generated on the fly for repaired data. The performance analysis and experimental results show that our designed service has comparable storage and communication cost, but much less computational cost during data retrieval than erasure codes based storage solutions. It introduces less storage cost, much faster data retrieval, and comparable communication cost comparing to network coding-based distributed storage systems. The many advantages of cloud computing are increasingly attracting individuals and organizations to move their data from local to remote cloud servers. In addition to major cloud infrastructure providers, such as Amazon, Google, and Microsoft, more and more third-party cloud data service providers are emerging which are dedicated to offering more accessible and user friendly storage services to cloud customers.

Although existing techniques have provided solutions for

them individually, the main challenge for cloud storage service is to simultaneously provide these capabilities at minimal cost. This is because in cloud computing both data storage and transmission are charged in the "pay-as-you-use" manner. Moreover, it is important to set cloud customers free by minimizing the complexity imposed on them in terms of computation/communication cost and burden of being online. Existing solutions address the reliability issue by adding data redundancy to multiple servers. These techniques can be categorized into replication-based solutions and erasure codes-based ones. Data replication is the most straightforward way of adding redundancy. The advantage of replication is its simplicity in data management. Repair of data on corrupted servers is also straightforward by simply copying the entire data from a healthy server. The main drawback of replication is its high storage cost. Moreover, replication-based solutions cannot satisfy the high-throughput requirement in distributed storage service like cloud computing, where a large number of users may access the service concurrently. This is because different users may want to access different pieces of data on a server, which would cause less effect hits but frequent disk I/O requests. This is because every block of data on a server is useful for decoding the original data, which leads to a high cache hit rate of the system.

This is achieved by recoding encoded packets in the healthy servers during the repair procedure. However, as network coding utilizes Gaussian elimination for decoding, the data retrieval in terms of computation cost is more expensive than erasure codes- based systems. Moreover, [12] adopts so called functional repair for data repair, i.e., corrupted data is recovered to a correct form, but not the exact original form. While this is good for reducing data repair cost, it requires the data owner to produce new verification tags, e.g., cryptographic message authentication code, for newly generated data blocks. Contributions are summarized as follows, We are among the first to explore the problem of secure and reliable cloud storage with the efficiency consideration for both data repair and data retrieval.

1) Our proposed cloud storage service provides a better overall efficiency of data retrieval and repair than existing counterparts. It also greatly reduces cost and burden of being online for the data owner by enabling public integrity check and exact repair.

2) The advantages of our proposed service are validated via both numerical analysis and experimental r e s u l t s .

Advantage:

- The advantage of replication is its simplicity in data management. Repair of data on corrupted servers is also straightforward by simply copying the entire data from a healthy server.
- Our proposed service are validated via both numerical analysis and experimental results, its be a main advantage of this system.

4.3 Deterministic and Efficiently Searchable Encryption

We present as-strong-as-possible definitions of privacy, and constructions achieving them, for public-key encryption schemes where the encryption algorithm is deterministic. We obtain as a consequence database encryption methods that permit fast (i.e. sub-linear, and in fact logarithmic,time) search while provably providing privacy that is as strong as possible subject to this fast search constraint. One of our constructs, called RSADOAEP, has the added feature of being length preserving, so that it is the first example of a public-key cipher. We generalize this to obtain a notion of efficiently-searchable encryption schemes which permit more flexible privacy to search-time trade-offs via a technique called bucketization. Our results answer much-asked questions in the database community and provide foundations for work done there. The classical notions of privacy for public-key encryption schemes, namely in distinguish ability or semantic security under chosen-plaintext or chosen-cipher text attack, can only be met when the encryption algorithm is randomized. This paper treats the case where the encryption algorithm is deterministic. We begin by discussing the motivating application

.Fast search. Remote data storage in outsourced databases is of increasing interest [51]. Data will be stored in encrypted form. (The database service provider is not trusted.) We are interested in a public key setting, where anyone can add to the database encrypted data which a distinguished "receiver" can retrieve and decrypt. The encryption scheme must permit search (by the receiver) for data retrieval. Public-key encryption with keyword search (PEKS) is a solution that provably provides strong privacy but search takes time linear in the size of the database. Given that databases can be terabytes in size, this is not allowed. The practical community indicates that they want search on encrypted data to be as efficient as on unencrypted data, where a record containing a given field value can be retrieved in time logarithmic in the size of the database. Deterministic encryption allows just this. The encrypted fields can be stored in the data structure, and one can find a target cipher text in time logarithmic in the size of the database. The question is what security one can expect. To answer this, we need a definition of privacy for deterministic encryption.

A definition. One possibility is to just ask for one wayness, but we would like to protect partial information about the plaintext to the maximum extent possible. To gauge what this could be, we note two inherent limitations of deterministic encryption. First, no privacy is possible if the plaintext is known to come from a small

space. Indeed, knowing that c is the encryption under public key pk of a plaintext z from a set Z, the adversary can compute the encryption cz of z under pk for all z ∈ Z, and return as the decryption of c the z satisfying cz = c. We address this by only requiring privacy when the plaintext is drawn from a space of large min-entropy. Second, and more subtle, is that the cipher text itself is partial information about the plaintext. We address this by only requiring non-leakage of partial information when the plaintext and partial information do not depend on the public key. This is reasonable because in real life public keys are hidden in our software and data does not depend on them. We provide a semantic-security style definition of privacy for deterministic encryption that takes these issues into account. While certainly weaker than the classical notions met by randomized schemes, our notion, which we call PRIV, is still quite strong. The next question is how to achieve this new notion.

Efficiently searchable encryption. We introduce the notion of efficiently searchable encryption (ESE) schemes. These are schemes permitting logarithmic time(fast) search. Encryption may be randomly, but there is a deterministic function of the plaintext that can also be computed from the cipher text and serves as a "tag," permitting the usual (fast) comparison-based search. Deterministic encryption schemes are a special case and the notion of security remains the same. The benefit of the generalization is to permit schemes with more flexible privacy to search-time trade-offs. Specifically, we analyze a scheme from the database literature that we call "Hash-and- Encrypt." It encrypts the plaintext with a randomized scheme but also includes in the cipher text a deterministic, collision-resistant hash of the plaintext. We prove that this scheme is PRIV secure in the RO model when the underlying encryption scheme is IND-CPA. With this scheme, loss of privacy due to lack of entropy in the plaintext space can be compensated for by increasing the probability of hash collisions. The trade-of is that the receiver then also gets "false positives" in response to a search query and must spend the time to sift through them to obtain the true answer. This technique is known as bucketization in the database literature, but its security was not previously rigorously analyzed.

Advantage:

- The construction achieves security when min-entropy of the data is high enough to preclude a dictionary attack by the adversary against the scheme.
- The benefit of the generalization is to permit schemes with more flexible privacy to search-time trade-offs.

4.4 Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions

We identify and fill some gaps with regard to consistency (the extent to which false positives are produced) for public-key encryption with keyword search (PEKS). We done computational and statistical relaxations of the existing notion of perfect consistency, show that the scheme of is computationally consistent, and provide a new scheme that is statistically consistent. We also provide a transform of an anonymous IBE scheme to a secure PEKS scheme that, unlike the previous one, guarantees consistency. Finally we suggest three extensions of the basic notions considered here, namely anonymous HIBE, public-key encryption with temporary keyword search, and identity-based encryption with keyword search. There has recently been interest in various forms of \searchable encryption". In this paper, we further explore one of the variants of this goal, namely public-key encryption with keyword search (PEKS) as introduced by Boneh, Di Crescenzo, Ostrovsky and Persiano [7]. We begin by discussing consentiency- related issues and results, and then consider the connection to anonymous identity based encryption (IBE) and ¯nally discuss some extensions. Any cryptographic primitive must meet two conditions. One is of course a security condition. The other, which we will here call a consistency condition, ensures that the primitive full's its function. For example, for public-key encryption, the security condition is privacy. (This could be formalized in many ways, e.g. IND-CPA or INDCCA.) The consistency condition is that decryption reverses encryption, meaning that if M is encrypted under public key pk to result in cipher text C, then decrypting C under the secret key corresponding to pk results in M being returned. It is natural to ask if BDOP-PEKS meet some consistency condition that is weaker than theirs but still adequate in practice. To answer this, we provide some new definitions. Somewhat unusually for a consistency condition, we formulate consistency more like a security condition, via an experiment involving an adversary. The deference is that this adversary is not very adversarial": it is supposed to reject some kind of worst case but not malicious behaviour. However this turns out to be a difficult line to draw , definition ally, so that some subtle issues arise.

Advantage:

- One advantage of this approach is that it naturally gives rise to a hierarchy of notions of consistency, namely perfect, statistical and computational.
- The advantage of any (even computationally unbounded) adversary is negligible.
- The advantage of any polynomial-time adversary is negligible.

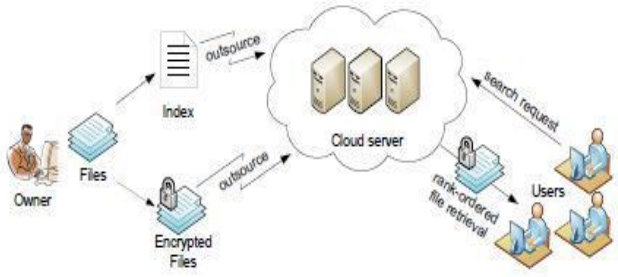
4.5 Enabling Secure and Efficient Ranked

Keyword Search over Outsourced Cloud Data Cloud computing economically enables the paradigm of data service outsourcing. However, to protect data privacy,

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 6, June 2017

sensitive cloud data has to be encrypted before outsourced to the commercial public cloud, which makes effective data utilization service a very challenging task. Although traditional searchable encryption techniques allow users to securely search over encrypted data through keywords, they support only Boolean search and are not yet sufficient to meet the effective data utilization need that is inherently demanded by large number of users and huge amount of data files in cloud. In this paper, we define and solve the problem of secure ranked keyword search over encrypted cloud data. Ranked search greatly enhances system usability by enabling search result relevance ranking instead of sending undifferentiated results, and further ensures the file retrieval accuracy. Specifically, we explore the statistical measure approach, i.e. relevance score, from information retrieval to build a secure searchable index, and develop a one-to-many order-preserving mapping technique to properly protect those sensitive score information. The resulting design is able to facilitate efficient server side ranking without losing keyword privacy. Thorough analysis shows that our proposed solution enjoys "as-strong-as-possible" security guarantee compared to previous searchable encryption schemes, while correctly realizing the goal of ranked keyword search. Extensive experimental results demonstrate the efficiency of the proposed solution.



Architecture for search over encrypted cloud data

Cloud computing is the long dreamed vision of computing as a utility, where cloud customers can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources [3]. The benefits brought by this new computing model include but are not limited to: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc [4]. As Cloud Computing becomes prevalent, more and more sensitive information are being centralized into the cloud, such as emails, personal health records, company finance data, and government documents, etc. The fact that data owners and cloud server are no longer in the same trusted domain may put the outsourced unencrypted data at risk [5]: the cloud server may leak data information to unauthorized entities [6] or even be hacked [7]. It follows that sensitive data has to be encrypted prior to outsourcing for data privacy and combating unsolicited accesses.

However, data encryption makes effective data utilization a very challenging task given that there could be a large amount of out sourced data files. Besides, in Cloud Computing, data owners may share their outsourced data with a large number of users, who might want to only retrieve certain specific data files they are interested in during a given session. One of the most popular ways to do so is through keyword-based search. Such keyword search technique allows users to selectively retrieve files of interest and has been widely applied in plaintext search scenarios Unfortunately, data encryption, which restricts user's ability to perform keyword search and further demands the protection of keyword privacy, makes the traditional plaintext search methods fail for encrypted cloud data. Therefore, how to enable a searchable encryption system with support of secure ranked search, is the problem tackled in this paper. Our work is among the first few ones to explore ranked search over encrypted data in Cloud Computing. Ranked search greatly enhances system usability by returning the matching files in a ranked order regarding to certain relevance criteria (e.g., keyword frequency), thus making one step closer towards practical deployment of privacy-preserving data hosting services in the context of Cloud Computing. To achieve our design goals on both system security and usability, we propose to bring together the advance of both crypto and IR community to design the ranked searchable symmetric encryption scheme, in the spirit of "as-strong as- possible" security guarantee. Specifically, we explore the statistical measure approach from IR and text mining to embed weight information (i.e. relevance score) of each file during the establishment of searchable index before outsourcing the encrypted file collection. As directly outsourcing relevance scores will leak lots of sensitive frequency information against the keyword privacy, we then integrate a recent crypto primitive [12] order preserving symmetric encryption (OPSE) and properly modify it to develop a one-to-many order preserving mapping technique for our purpose to protect those sensitive weight information, while providing efficient ranked search functionalities.

Our contribution can be summarized as follows

1) For the first time, we define the problem of secure ranked keyword search over encrypted cloud data, and provide such an effective protocol, which fulfils the secure ranked search functionality with little relevance score information leakage against keyword privacy.

2) Thorough security analysis shows that our ranked searchable symmetric encryption scheme indeed enjoys "as-strong-as-possible" security guarantee compared to previous SSE schemes.

3)We investigate the practical considerations and enhancements of our ranked search mechanism, including the efficient support of relevance score dynamics, the authentication of ranked search results, and the reversibility of our proposed one-to many order-preserving mapping techniques

4) Extensive experimental results demonstrate the effectiveness and efficiency of the proposed solution.

Advantage:

- A lot of computation overhead when file collection changes, is a significant advantage in our scheme.
- The motivated ranked key search over encrypted file to achieve economies of scale in Cloud Computing.

5 CONCULSION

In this paper, we explore the problem of secure multi-keyword search for multiple data owners and multiple data users in the cloud computing environment. Different from prior works, our schemes enable authenticated data users to achieve secure, convenient, and efficient searches over multiple data owners' data. To efficiently authenticate data users and detect attackers who steal the secret key and perform illegal searches, we propose a novel dynamic secret key generation protocol and a new data user authentication protocol. To enable the cloud server to perform secure search among multiple owners' data encrypted with different secret keys, we systematically construct a novel secure search protocol. To rank the search results and preserve the privacy of relevance scores between keywords and files, we propose a novel additive order and privacy preserving function family. Moreover, we show that our approach is computationally efficient, even for large data and keyword sets. As our future work, on one hand, we will consider the problem of secure fuzzy keyword search in a multi owner paradigm. On the other hand, we plan to implement our scheme on the commercial clouds, we motivate and solve the problem of supporting efficient ranked keyword search for achieving effective utilization of remotely stored encrypted data in Cloud Computing. We first give a basic scheme and show that by following the same existing searchable encryption framework, it is very inefficient to achieve ranked search. We then appropriately weaken the security guarantee, resort to the newly developed crypto primitive OPSE, and derive an efficient one-to-many order-preserving mapping function, which allows the effective RSSE to be designed..

6 REFERENCES:

[1] Cong Wang, Student Member, IEEE, Ning Cao, Student Member, IEEE, Kui Ren, Senior Member, IEEE, Wenjing Lou, Senior Member, IEEE, "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data", IEEE Transactions on Parallel and Distributed Systems Vol.23 No.8 Year 2012.

[2] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," in Proc. of ICDCS'10, 2010.

[3] 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.

[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Commun. ACM, vol. 53, no. 4, pp. 50–58, 2010.

[5] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," IEEE Trans. Comput., vol. 62, no. 2, pp. 362–375, Feb. 2013.

[7]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. Full version of current paper. Available at IACR Cryptology ePrint Archive, http://eprint.iacr.org.

[8]M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In 29th ACM STOC. ACM Press, 1997.

[9]Fuzzy keyword search over encrypted data in cloud computing" by T. Balamuralikrishna.

[10]Implementation of Fuzzy keyword search over encrypted data in cloud computing" by D. VASUMATHI.

[11]Fuzzy keyword search over encrypted data in cloud computing", Illinois Institute of Technology, ISSN: 2321- 8134.

[12]Z. Xu, W. Kang, R. Li, K. Yow, and C. Xu, "Efficient multikeyword ranked query on encrypted data in the cloud," in Proc. IEEE 19th Int. Conf. Parallel Distrib. Syst.,Singapore,Dec. 2012, pp. 244–251.

[13] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," inProc. IEEE INFOCOM, San Diego, CA, USA, Mar. 2010, pp. 1–5.