ISSN 2395-695X (Print)
ISSN 2395-695X (Online)

International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

Defend Clandestine Rated Multi-keyword Search for Multi-proprietors with Server Side Encryption in cloud computing

Vanishree K.A[1], Santhiya.C[2], Gayathri.V[3] and Dr. M.K. Chandrasekaran PhD[4]

M.E, Department of CSE, Angel College of Engineering and Technology, Tiruppur, Tamilnadu, India[1]

M.E, Department of CSE, Angel College of Engineering and Technology, Tiruppur, Tamilnadu, India[2]

B.E, Department of CSE, Angel College of Engineering and Technology, Tiruppur, Tamilnadu, India[3]

HOD, Department of CSE, Angel College of Engineering and Technology, Tiruppur, Tamilnadu, India[4]

***Abstract*—** In cloud computing, secure searches over encrypted cloud data have motivated numerous research works under the single owner model to obtain privacy. But, most cloud servers in practice do not just serve single owner; instead, they support multiple owners to share their profits which is fetched from cloud server. We proposed a scheme which deals with Defend Clandestine rated Multi-keyword Search in a Multi-proprietor model (DCMSM). To activate cloud servers to perform secure search by novel secure search protocol. The distinct proposal of Privacy Preserving Ranked Search protocol has done the rank the search results and preserves the privacy of appositeness scores between keywords and files. Dynamic secret key generation protocol and a novel data user authentication protocol to prevent the attackers from intruding secret keys and concealing the legal data users submitting searches. But one of the disadvantages in this scheme is client side encryption alone is allowed. We propose a server side encryption model to enhance further security in DCMSM.

Keywords—Cloud computing, rated keyword search, multiple owners, privacy preserving, dynamic secret key, server side encryption.

I. INTRODUCTION

Cloud computing is a dissident technology that altering the way IT hardware and software are designed and purchased. Cloud computing provides plentiful benefits including easy access, decreased costs, quick deployment and flexible resource management, etc. Enterprises of all sizes can boost up the cloud innovation and collaboration. Even though cloud computing lagging on privacy concerns, individuals and enterprise users is hesitate to outsource their sensitive data, including emails, personal health records and government confidential files, to the cloud. This is because only sensitive data are exposed out to a remote cloud; the corresponding data owners lose direct control of these data.

When 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 wants to stay online to generate trapdoors for data users. However, when a huge amount of data owners are presented, asking them to stay online immediately to generate trapdoors would seriously affect the flexibility and usability of the search system. Second, since nobody would be willing to share our secret keys with different data owners, then they would prefer to use their own secret keys to encrypt their secret data. At the same time, it is very challenging to perform a secure, convenient, and efficient search over the data encrypted with dissimilar secret keys. Third, we should ensure efficient user enrollment and revocation mechanisms, so that our system delights in excellent security and scalability which is done only by multiple data owners' involvement.

In our proposed scheme which is deals with Defend Clandestine rated Multi-keyword Search in a Multi-proprietor model (DCMSM). The main aim of this paper is listed as follows:

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

- We propose data user authentication protocol, which prevent the attackers from intruding secret keys and concealing the legal data users submitting searches.
- We systematically construct a novel secure search protocol, allow data owners to encrypt keywords with self-chosen keys and allow authenticated data users to ask without knowing these keys.
- We propose Privacy Preserving Ranked Search, according to their preference which allows data owners to protect the privacy of appositeness scores using different functions while still permitting the cloud server to rank the data files exactly.
- We propose server side encryption to enhance the security in server side from eavesdroppers.

II. DATA USER AUTHENTICATION PROTOCOL

Based on the search result, the legal data users search performance and launching statistical attacks are prevent from intruders this action can be done by administration server's trapdoor decryption, but data users must be authenticated. Traditional authentication methods often follow three steps.

1) Data requester and data authenticator share a secret key.

2) The requester encrypts his personally identifiable information using secret key and sends the encrypted data to the legal users.

3) The legal user decrypts the received data with secret key and authenticates the decrypted data.

Even though this method has two main shortcomings:

1) To unchanged the remaining secret key shared between the requester and the authenticator so that it is easy to incur replay attack.

2) One time the secret key is revealed to attackers, the authenticator cannot differentiate between the legal requester and the attackers; the attackers can make believe to be legal requesters without being detected. In this, we first give an overview of the data user authentication protocol. Then, we commence how to achieve secure and efficient data user authentication.

2.1 Overview

The main idea of the user authentication protocol is illustrated by an example. Let us assume that Alice wants to be authenticated by the administration server, so she starts a conversation with the bob (server). The contents of the conversation are authenticated by bob. Once the contents are authenticated, both Alice and the bob will generate the initial secret key according to the conversation. After the initialization, to be authenticated successfully, then Alice has to grant the historical data of their conversations. If the authentication is done, then both Alice and the administration server will exchange their secret keys according the contents of the conversation. In this manner, the secret keys keep changing dynamically; an attacker cannot start a successful conversation with the administration server, without knowing the exact historical data.

2.2 User Authentication

We first introduce the format of the authentication data dynamic key generation method and the authentication protocol. As shown

Request counter	Last Request Time	Personally Identifiable Data	Random Number	CRC

Table.1: Format of Authentication Data

In table.1, the authentication data consists of five parts:

1) The request counter field records the number of search requests that the data user has submitted.

2) The last request time field requests the legal data user to provide the historical data of his previous request time.

3) 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 added.

4) The random number and CRC field further used to check whether the authentication data has been tampered.

The key point of a successful authentication is to offers

- The dynamically changing secret keys.
- The historical data of the corresponding data user.

After each successful authentication process, the secret key will be changed dynamically according to the previous secret key and some historical data through the successful authentication between the administration server and the data user. Therefore, once an attacker steals a secret key, he can hardly get any benefits and hack the user data in easy manner. On one hand, the attackers cannot even construct a legal authentication data if the attacker knows nothing about the historical data of the legal data user. On the other hand, if the previous key will be expiry, and then the authenticated user performs another successful authentication.

2.3 Illegal Search Detection

In our proposed scheme, the authentication process is sheltered by the dynamic secret key and the historical information. If an attacker has successfully eavesdropped the secret key and then he starts to construct the authentication data. If the attacker has not successfully intrude the historical data, e.g., the request counter, the last request time, they are unable to build the correct authentication data.

Therefore this illegal action will be detected by the administration server immediately. Further, if the attacker has successfully intrude all data means the attacker can correctly construct the authentication data and pretend himself to be without being detected by the administration server. However, once the authenticated user performs his search, since the secret key on the administration server side has altered, there will be contradictory secret keys between the administration server and the authenticated user. Therefore, the data user and administration server will soon detect this illegal action.

2.4 Data User Revocation

Data user revocation does not require decrypting and updating large amounts of data stored on the cloud server. Instead of, the administration server only needs to update the secret data stored on the cloud server. As a result, the previous trapdoors will be expired without the help of the administration server; the officially cancelled data user cannot generate the correct trapdoor. Therefore, once the data user is revokes, then he cannot perform correct searches.

III. MATCHING DIFFERENT-KEY ENCRYPTED KEYWORDS

Many data owners are often involved in practical cloud applications. They would be unwilling to share secret keys with others because of privacy concerns. Instead, they prefer to use their own secret keys to encrypt their sensitive data. When keywords of different data owners are encrypted with different secret keys, the upcoming queries from the data user are how to locate different-key encrypted keywords among multiple data owners. Multiple data owners can enable secure, efficient and convenient searches over encrypted cloud data. We systematically design schemes to achieve the following three requirements:

1) Different data owners use different secret keys to encrypt their keywords.

2) Without knowing these secret keys the authenticated data users can generate their trapdoors.

3) Upon receiving trapdoors, without knowing the exact value of keywords or trapdoors the cloud server can find the corresponding keywords from different data owners encrypted keywords.

3.1 Overview

Assume Alice wants to use the cloud to store her file; she first encrypts her file, and gets the ciphertext. To enable other users to perform secure searches on ciphertext, Alice extracts a keyword, and sends the encrypted keyword to the administration server. The administration server further decrypts the keyword, and submits the keyword to the cloud server. Now Bob wants to search a keyword, he first generates the trapdoor and acknowledges it to the administration server. The administration server decrypts the trapdoor, generates a secret data, and submits trapdoor, secret data to the cloud server. The cloud server will judge whether Bob's search request matches Alice's encrypted keyword by checking whether those are holds or not.

3.2 Keyword Encryption

For keyword encryption, the following conditions should be satisfied:

1) For encrypting the keywords different data owners use their own secret keys.
2) It would be encrypted to different cipher-texts each time for the same keyword.

These properties benefit our scheme for two reasons.

1) Losing the key of one data owner would not lead to the acknowledgement of other owner's data.
2) The cloud server cannot see any relationship among encrypted keywords.

The data owner delivers keywords to the administration server, and further it decrypts the keyword with his secret keys and gets the result. Therefore the administrative server further submits encrypted keyword to the cloud server. Since the administration server only does simple computations on the encrypted data, without knowing the secret keys of data owners they cannot learn any sensitive information from these randomly encrypted data.

3.4 Trapdoor Generation

Our proposed scheme should satisfy two main conditions to make the data users generate trapdoors securely, conveniently and efficiently. They are,

1) The data user does not need to ask a large amount of data owners for secret keys to generate trapdoors.
2) For the identical keyword, the trapdoor generated each time should be different. To meet this condition, the trapdoor generation is conducted in two steps:

2.1) Based on searching keyword and a random number the data user generates trapdoors.
2.2) Decrypt the trapdoors by the administration server for the authenticated data user.

Assume a data user wants to search keyword, so he encrypts it as follows. Secret keys of data owners are not required during the trapdoor generating process. Moreover, with the help of random variable for the same keyword we can generate two different trapdoors which prevent attackers from knowing the bond among trapdoors. Upon receiving trapdoors, the administration server first generates a random number and then decrypts the trapdoor. Finally, the administration server submits trapdoor to the cloud server.

3.5 Keywords Matching among Different Data Owners

All encrypted files and keywords of different data owners are stored in the cloud server. The administration server will also store a secret data on the cloud server. The cloud will search over the data of all these data owners after receiving a query request. The cloud search request processes with two steps. First, the cloud counterpart the queried keywords from all keywords stored on it and it gets a candidate file set. Second, the cloud ranks files in the candidate file set and finds the most top-k relevant files. We introduce the counterpart strategy here, while leaving the task of introducing the ranking strategy. When the cloud obtains the trapdoor and encrypted keywords. Then he can judge whether an encrypted keyword is located and holds if the process is true.

IV. PRIVACY PRESERVING RANKED SEARCH

However, which is impossible to simply return undifferential files to data users for the following two reasons. First, returning all data user files would cause plentiful communication overhead for the whole system. Second, according to their query the data users would only concern the top-k relevant files. In this, we first clarify an order and privacy preserving encoding scheme. Then we exemplify privacy preserving encoding scheme. Finally, we apply the proposed scheme to encode the appositeness scores and obtain the

top-k search results.

X	1	2	3	4	5
f(x)	100-1000	1100-1800	2000-4200	4300-5000	5100-7000

Table.2: An example of Order Preserving and Privacy Preserving Function

4.1 Order and Privacy Preserving Function

To rank the appositeness score while preserving its privacy, the proposed function should satisfy the following conditions. 1) According to the encoded appositeness scores, the function should preserve the order of data, through this the cloud server determine which file is more relevant to a certain keyword.2) This function should not be cancelled by the cloud server so that it can make comparisons on encoded appositeness scores without knowing their actual values. 3) Dissimilar data owners should have different functions such that revealing the encoded value they would not come forward to the leakage of encoded values of other data owners. In order to satisfy condition 1, we introduce a data processing part which preserves the order of data files. To satisfy condition 2, we introduce a disturbing part which helps prevent the cloud server from revealing this function. To satisfy condition 3, we use the process ID of data owners.

4.2 Additive Order and Privacy Preserving Function

Obviously, from table f(x) has disturbing part is introduced to order preserving to preserve privacy. To evaluate the secure ranked multi-keyword search, the sum of any two encoded appositeness scores should still be ordered and privacy preserved.

4.3 Encoding appositeness scores

It encodes the appositeness score with an additive order and privacy preserving function in our additive order and privacy preserving function family. Through this data owners can individually choose a function from this family to protect the privacy of their appositeness scores.

4.4 Ranking search results

We implement the sum of the appositeness scores between the no. of files and matched keywords for data owner collection as the metric to rank search results. The cloud server ranks the sum of encoded appositeness score with the following two conditions:

(1) Two encoded data belong to the same data owner.
(2) Two encoded data belong to two different data owners. Thus, it is easy for the cloud to return the top-k relevant files to the data user.

V. SERVER SIDE ENCRYPTION

Server-side encryption manages encryption key along with our data, encoding the information once it is uploaded to the provider. In comparison to client-side encryption, this method reduces the complexity of the network environment whilst maintaining the isolation of our data. However the key material can store encrypted and decrypted data together it is employed to protect, make restrict by unauthorized access trying. Providers may agree to keep their data confidential, but can use the data for their personal use alone, for example to enhance search results or deliver advertisements. For this reason, some servers will oppose to use client-side encryption.

The Cloud Security Alliance reported management operate from cloud providers for hosting the data prevents protects from unauthorized disclosure of customer data, and advised customers to retain whole control. This can be accomplished by using client-side encryption methods.

'Zero-knowledge service', known as client-side encryption, allows maintaining the keys (passwords) without server management. The files decryption is only possible by using the key created by the individual user. The law enforcement requests data is protected against, as providers do not have access the encoded

files. With individual-user power comes also responsible for good key management and data backup that should be implemented, as passwords cannot be retrieved when lost.

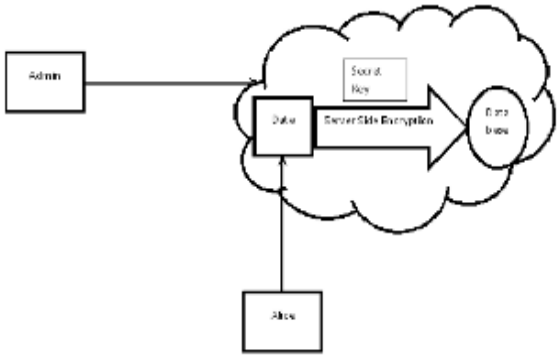

Fig 3 Server Side Encryption

5.1 Client side Encryption process

Without the blind trust in the cloud storage provider, the user could choose to encrypt their files one-by-one and only upload these secure files to the cloud. This works well for process like backup, but this is only possible for as long as the user does not want to share their files. As the amount of data and the number of people involved in the sharing mount, this way becomes intractable. The administrator with group management, invitation of new group members, and official cancellation of expired permissions quickly becomes complex. This approach is inflexible enough for collaborative use.

5.2 Benefits of Server side Encryption

- Access Control Mechanism is implemented.
- Enhancing security in server side. So that, Server side also trusts worthy.
- User can store their data in object level, from this unauthorized person is restricted to view their secrets.

VI. SECURITY ANALYSES

It define as follows,

6.1 Data Files

Before uploading the data files are protected by symmetric encryption. Since the encryption algorithm is not breakable and also the cloud server cannot know the data.

6.2 Keywords

We originate the security goals achieved by DCMSM with the semantically secure against the chosen keyword attack under the selective security model and achieve keyword secrecy in the random oracle model.

6.3 Trapdoors

The cloud server must solve the discrete logarithm problem with large prime factor if they want to know the actual value of the trapdoor, and differentiate two trapdoors. Through this the privacy of trapdoor is protected as long as the discrete logarithm problem is rigid.

6.4 Appositeness Scores

Appositeness scores are encoded with two Additive Order and Privacy Preserving Functions and we analyze

the security of additive order and privacy preserving functions. By using this technique and data owner ID to get the value and form the N values. Through this it is infeasible to break the additive order and privacy preserving family function and encoded relevance score also preserved.

6.4 Server Side Encryption

Several cloud storage middleware are presented with just double clicks on the Internet. Most of them even provide free-of-charge services for domestic users. Most solutions promise provides security for our data in the cloud, almost none of them can protect the cloud storage, provider itself peering into our data. This is because we apply server-side encryption through this user has no need to control over how the encryption is done and who has access the decryption keys. In other words, the user has to reliance the cloud storage provider for being honest and not revealing their private data.

CONCLUSION

When comparing with previous works, our schemes enable authenticated data users to achieve secure, convenient, and efficient searches over multiple data owner's data. To activate cloud servers to perform secure search, we methodically build a novel secure search protocol. We propose another technique as Privacy Preserving Ranked Search for the purpose of rank the search results and preserve the privacy of appositeness scores between keywords and files. By introducing a modish dynamic secret key generation protocol and a novel data user authentication protocol to prevent the attackers from intruding secret keys and concealing the legal data users submitting searches. To enhance security in server side we propose sever side encryption. As our future work, we will consider the difficulty of secure fuzzy keyword search in a multi-owner paradigm.

REFERENCES

[1] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W.Lou, Privacy-preserving public auditing for secure cloud storage," Computers, IEEE Transactions on, vol. 62, no. 2, pp. 362-375, 2013.

[2] D.Song, D.Wagner, and A.Perrig, "Practical techniques for searches on encrypted data," in Proc. IEEE International Symposium on Security and Privacy (S&P'00), Nagoya, Japan, Jan.2000, pp. 44-55.

[3] R.Curtmola, J.Garay, S.Kamara, and R.Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proc. ACM CS'06, VA, USA, Oct. 2006, pp.79-88.

[4] N.Cao,C. Wang, M. Li, K. Ren, and W. Lou, "Privacy- preserving multi-keyword ranked search over encrypted clouddata,"inProc.IEEEINFOCOM'11,Shanghai, China,Apr.2011,pp. 829-837.

[5] N.Cao, C.Wang, M. Li, K. Ren, and W. Lou,"Privacy- preserving multi-keyword ranked search over encrypted cloud data," Parallel and Distributed Systems, IEEE Transactions on,vol. 25, no. 1, pp. 222-233, 2014.

[6] T. Jung, X. Y. Li, Z. Wan, and M. Wan, "Privacy preserving cloud data access with multi-authorities," in Proc. IEEE INFOCOM'13, Turin, Italy, Apr. 2013, pp. 2625-2633.