PRIVATE SEARCHING ON ENCRYPTED CLOUD DATA

DARSANA DINESH
TIFAC CORE of Cyber Security, Amrita Vishwa Vidyapeetham, Coimbatore, India
E-mail : darsana.dinesh88@gmail.com.

Abstract - Cloud computing provides services like software, computation and storage services for which end-user need not worry about knowledge of the physical location and configuration of the system that delivers the services. Data leakage is a major problem in cloud computing and the privacy of user is under serious threat. The goal of the system is that, server shouldn’t have any knowledge on the data it has got and the interests of the user, thus protecting the privacy of the user. Server should not understand the keywords interesting to the user and the documents containing those keywords. We intend to provide privacy by encrypting the data outsourced to the cloud. But a strong encryption will make it difficult to search files. Sensitive data is encrypted using a homomorphic encryption scheme, so that operations on encrypted data is equivalent to the operations on plaintext. This paper suggests a privacy preserving search scheme that is flexible to be applied for a sender-receiver scenario and data outsourcing scenario. The latter scenario is modified for a dynamic group of users. The scheme assures both data privacy and member privacy, by using the Paillier cryptosystem. The paper suggests a new approach using self-blinding property of the cryptosystem, to provide additional security.

Keywords- Cloud computing; privacy; Homomorphic encryption; Paillier cryptosystem; self-blinding

I. INTRODUCTION

Cloud computing is a new computing model that distributes the computing missions on a resource pool which includes a large amount of computing resources. Main services offered by clouds are data storage and data processing through user-defined or publicly available programs. Apart from classical security challenges of shared services raised by third party intruders or malicious users, such as access control, clustering, service abuse, key-management, and denial-of-service, outsourcing of data storage and processing raises new challenges in the face of potentially malicious cloud providers. The main challenge in the particular context of cloud computing is to come up with a scheme that achieves privacy while preserving the efficiency of cloud computing. The motivation is to assure privacy against curious cloud servers, clouds with data centers located in rogue countries or with insufficient security guarantees, and to avoid data leakage in case of operational failures in the cloud. We consider the server to be honest-but-curious. The cloud server has no intention to actively modify the message flow or disrupt any other kind of services, but is curious to infer and analyze the message flow received during the protocol so as to learn additional information. Data are always open to anyone who has access to the server and it becomes worse when the cloud provider is compromised. Sensitive data has to be encrypted before outsourcing, but then, effective data utilization becomes a very challenging task.

The existing system is that the client has to trust the server because all the data are executed in unencrypted form and is entirely under the control of the server. If data are encrypted, attacker can see only the encrypted version of user’s sensitive data. But then, the data must be decrypted each time for any operation on it since the applications that work on these data, need to understand them. To reduce the overhead, there is a need for a mechanism to operate on encrypted programs and data. For this, we use homomorphic encryption, a form of encryption where a specific algebraic operation performed on the plaintext is equivalent to another (possibly different) algebraic operation performed on the ciphertext.

This paper suggests a mechanism to search documents and also preserve the privacy of user. The scheme is designed by focusing on two application scenarios on cloud,

1. A sender encrypts the data and send to receiver via cloud provider. In this case sender encrypts data with public key, and since only receiver knows the private key he retrieves data from cloud and decrypt it, e.g., e-mail.

2. A company will encrypt documents using public key and outsource to cloud. When it needs a document, it queries and decrypts the retrieved document.

The algorithm is designed for the latter scenario, even though it can be used for first scenario too.

Goal of the Scheme

1. Data Privacy against cloud server - A server can’t know anything about keywords and document contents, i.e., encrypted documents must not provide any information about their contents, the ciphertexts for same word in different documents should be different so they leak nothing about their contents, and there must not exist information which is leaked from the query and the results generated in the search process.
2. Data Privacy against leaving member - A leaving member cannot search and retrieve any documents after she has left the group.

3. User Privacy against GM - A GM cannot know anything about the documents a user retrieves.

B. Organization of paper

Section 2 discuss the related works in the field. Section 3 is preliminaries, in which the encryption scheme used by the system and the model of the system are discussed. Section 4 explains the algorithm of the proposed scheme and in section 5 we conclude the paper.

II. RELATED WORK

Searchable encryption was first introduced by Song, Wagner, and Perrig [1]. But they proposed a symmetric key setting, which is not possible in sender-receiver scenario. Goh [2] presented a scheme using Bloom filters to build a secure index for each file. But, Goh's scheme induces false positives inevitably and requires that all parties share all keys. Many searchable symmetric encryption schemes are proposed for secure keyword search [3]. But these schemes restrict the user to a single person. It's more unreliable in multi-user scheme and impractical in a sender-receiver scenario like in emails. Boneh et al. [5] presented the first public-key based searchable encryption scheme. In their construction, anyone with the public key can write to the data stored on the server, but only authorized users with the private key can search.

Based on Goh's scheme [2], Park et al. [4] proposed search schemes for groups, which can deal with membership changes without re-encrypting data. The main idea is to use one-way hash chain in reverse order to generate group session keys, encryption keys and index generation keys. In these schemes, all the group members use identical secret keys to make secure indices and trapdoors, and a set of new group keys must be generated for each session. Therefore, the schemes bring a risk to the key management; and the size of a query would become larger as the number of sessions increases. It should be noted that, if one of the leaving members reveal her group key to the server, the server can decrypt all the documents encrypted previously because a user can know all of the previous group keys by hashing the current group key repeatedly.

Peishun et al. [6] discuss a searchable protocol on multi-users sharing encrypted data, but the shortcomings of conjunctive keyword retrieval exists since documents are retrieved only if all the keywords in the query are satisfied. It demands the user to keep memory on all the keywords of a document. This is not possible when the user's interest is not on a document, but on a topic. So we suggest an asymmetric method of privacy preserving keyword searching scheme for dynamic group of users.

III. PRELIMINARIES

A. System Model

The system involves a dynamic group of users. Users store their encrypted documents together with the trapdoor on an untrusted server. All the documents can be accessed by each user from the group. The group may be dynamic, i.e., a person may join and leave the group. For a leaving member, all documents stored on the server should be no longer accessible to the member including her own documents. A joining member not only can store her encrypted documents on the server that would be shared by other users, but also is able to retrieve all the previous and current documents that are owned by the group members.

GM plays an important role as it manages the group members, their joining and leaving operations, group key generation and distribution, and all other operations related to maintenance of secret keys. First, GM setups the system and assigns an authentication code to each user. Then, every user encrypts her documents using the public key and generates trapdoor. When a user wishes to retrieve the documents containing some keywords, she uses the group public keys to generate ciphertexts of keywords and sends it along with her authentication code to the server. Next, the server verifies the authentication code to confirm that the user is legitimate. Then, for the legitimate user, server will select documents and multiplies the encrypted query with the trapdoor of the document and search for a match in the document. After that, the server sends all matched documents to the user. On receiving the documents, the user multiplies them with her secret value and sends the encrypted data to GM, who uses decryption key to decrypt the documents and return them to the user. Finally, the user gets the desired documents by subtracting her secret value from the received data.

B. Building Blocks

1) Paillier cryptosystem

The encryption scheme used by the proposed scheme is Paillier encryption scheme. Paillier cryptosystem is a homomorphic cryptosystem and poses some special properties. It poses the property of self-blinding, by which ciphertext can be changed without changing the plaintext [7]. Hence, the same keyword in different documents appear as different ciphertexts. This adds to the security of the scheme. Another property is that paillier cryptosystem is homomorphic under addition. Multiplication of ciphertexts is equivalent to addition of plaintexts. This property is used to add member privacy for the members of a dynamic group.
The public key is a 2k-bit RSA modulus \( N = pq \), where \( p, q \) are two large safe primes and the secret key is \((p, q)\). The plain-text space is \( \mathbb{Z}_N \) and the cipher-text space is \( \mathbb{Z}_N^2 \). To encrypt a message \( m \in \mathbb{Z}_N \), one chooses \( r \in \mathbb{Z}_N^* \) uniformly at random and computes the cipher-text as \[ E_{PK}(m, r) = g^m r^N (mod \ N^2), \]

where \( g = (1 + N) \) has order \( N \) in \( \mathbb{Z}_N^* \). To decrypt, one first computes \[ c_1 = c \mod N, \]
and then computes \( r \) from the equation, \[ r = c_1^{N-1} \mod \Phi(N). \]

Finally, one can compute \( m \) from the equation, \[ cr^{-N} (mod \ N^2) = 1 + mn. \]

The properties of Paillier's public-key cryptosystem are,
\[ D[E(m_1) \cdot E(m_2) (mod \ N^2)] = m_1 + m_2 (mod \ N) \]

\[ D[E(m)^k (mod \ N^2)] = km \mod N \]

Self-blinding property: change the cipher text without changing the value of the original plaintext \[ D[E(m), g^{x \cdot k} \mod n^2] = nx + m \equiv m \mod n \]

2) Dynamic Accumulators

Dynamic accumulators are cryptographic authentication primitives for optimally verifying set-membership relations. Given a set \( X \) of elements, an accumulator can be used to compute an accumulation value \( A(X) \), subject to which there exist short witnesses for any element in \( X \) that has been accumulated to \( A(X) \) [9]. For each element, it's witness can be used to provide a proof that the corresponding element is a member of \( X \). Insertion or deletion of any element from set \( X \) result in corresponding updates on the accumulation values and the element witnesses.

In this scheme we use RSA-based accumulator as discussed in [6]. For a set of elements \( X = \{x_1, \ldots, x_m\} \), the accumulator function is

\[ y_i = f(y_{i-1}, x_i), \]

where \( f \) is a one-way function: \( f(u, x) = u^x \mod n \) for suitably-chosen values of the seed \( u \) and RSA modulus \( n \). The accumulated value is \[ v = u_1 \cdot \ldots \cdot u_m \mod n, \]

and the witness for the element \( x_i \) is \[ w_i = u_1^x \cdot \ldots \cdot u_{i-1}^x \cdot u_{i+1}^x \cdot \ldots \cdot u_m^x \mod n. \]

The elements must be prime and the seed \( u \) must be from a group of quadratic residues.

IV. ALGORITHM

A sender-receiver scenario involves three parties: sender, receiver and cloud server. Sender encrypts the data using receiver's public key and send to the server along with the trapdoor. Receiver queries by sending encrypted search keyword to the server. Cloud server sends back matching encrypted documents and receiver decrypts them using the private key known only to him.

A data outsourcing scenario is applicable for a company, educational institution or any other dynamic group. It also involves three parties a trusted group manager (GM), members in the dynamic group, cloud server. GM setups the system and distributes an authentication code to every member. A member encodes her data and sends to server along with the index. Member queries using encrypted keyword and sends it along with her authentication code to the server. For the legitimate member, the server finds the matching documents and returns them to the member. Finally, the member interacts with GM to get the plaintext data.

Though the scheme can be used for both sender-receiver scenario and data outsourcing scenario, the scheme discussed below is designed for later model. For sender-receiver scenario, after the step 8, receiver just decrypt the retrieved documents using private key, known only to the receiver. Based on the algorithm discussed below, we define the problem as, members in a dynamic group retrieve their encrypted data from an untrusted server based on keywords and without any loss of data confidentiality and member's privacy.

1) System Setup

a) KeyGen : Key generation algorithm generates public and private keys of the underlying cryptosystem. In Paillier's encryption scheme, \( n = pq \) and \( sk = (p, q) \) where \( n \) is public key and \( sk \) is the private key \((p, q)\) are two large prime numbers).

b) Define a pool of PIN numbers, \( M = \{e \in \text{primes} : e \neq p \text{ and } e \neq q\} \)

c) Choose a random quadratic residue, \( u \mod n \) (\( u \neq 1\))

2) Group Authentication Process

The group \( G \) has \( N \) members \( \{U_1, U_2, \ldots, U_N\} \)

For every member \( U_i \), GM selects \( d_i \in \mathbb{M} \) as \( U_i \)'s PIN number

\[ \text{GM picks up } d' \in \mathbb{M} \setminus D, \text{ where } D = \{d_1, \ldots, d_N\} \]

and computes \( U_i \)'s secure code

\[ c_i = u^d \cdot d^{i+1} \cdot d^{N-i} \cdot N \mod n \]

GM sends STC to server and puts \( d' \) in \( D \)
c) Member can thus get d' by subtracting r from it

V. CONCLUSION

Cloud computing is service oriented and so protection of privacy is a necessity. For data privacy since public key cryptosystem is used, the scheme is applicable for both sender-receiver scenario and data outsourcing scenario. Both the documents and the keywords are encrypted. Server knows nothing about the specified keyword and the document retrieved. Thus, server has no knowledge on the interests of the retriever. Even anyone who monitors the traffic will not be able to deduce anything about the documents or the interests of searcher since they are encrypted. Moreover, ciphertext patterns are not leaked since each keyword is encrypted as different ciphertexts in different files (due to the property of self-blinding). Private key is known only to the GM and hence privacy is not affected even if any member leave the group and thus, the scheme can be applied for a dynamic group. Any leaving member of the group is not able to search and retrieve data after her revocation. Although a member interacts with the group manager, member privacy guarantees that the group manager knows nothing about the data the member retrieves. Thus, member privacy is also protected in the proposed scheme.

REFERENCES