Cloud Encrypted Data Retrieval Algorithm Based on Root Index Distance

Liu Yang
East China University of Political Science and Law Department of Information Science and Technology, Shanghai
lyang@ecupl.edu.cn

Abstract. In the data-as-a-service model of cloud computing, data of users is stored on servers in the cloud which provides services such as data storage and management. And this form of data management may reduce data security due to security threats to the cloud servers. The full homomorphic encryption data indexing algorithm (HEDR) can effectively solve the data security in the cloud computing environment, but with the increase of data volume, the retrieval efficiency and the accuracy of retrieval results of HEDR will be seriously affected. In this paper, aiming at solving the problem of HEDR, a root index distance (RID) based on the root index distance is proposed. By encrypting and processing the root index, the depth of the index is reduced in the case of irrelevant data volume. This reduces the impact of noise and improves the accuracy of data encryption index results. Finally, the analysis shows that the complexity of the RID algorithm does not increase, while the data security is significantly improved.

1. The first section in your paper

With the rapid development of information technology and the demand for social development today, cloud computing has gradually become a hot topic. It provides a relatively safe, reliable, and easily accessible storage center for a large scale data. The computing power provides technical support for data storage, sharing, security, and analysis. With the continuous "landing" of various cloud computing technologies, people have gradually begun to enter the "cloud era." Cloud computing combines diverse terminal technologies, data storage technologies, distributed computing, and virtualization technologies to integrate various traditional computing resources, storage resources, and software resources to form a cloud with "supercomputing" capabilities. In the cloud, people can use various hardware and software resources and share data conveniently, which saves huge resources and costs.

After the cloud computing technology is widely used in the society, its development has also encountered more and more challenges. For example, it is shown in the investigation report of the Internet Data Center (IDC) on the concerns of users in the cloud computing environment. Among the eight major challenges facing cloud computing, data security is the most concern[1]. The data security problem can be divided into two aspects. In one aspect, cloud data security refers to the loss of stored data due to software and hardware failures of cloud storage resources. At present, this problem mainly adopts methods such as data backup and data verification. To increase the security of data; the other is that the data stored in the cloud will not be stolen by unauthorized parties, this issue is mainly solved by encrypting the data in the cloud. Undoubtedly, data encryption and storage and circulation will separate the data from unauthorized third parties. However, the problem that arises is that data...
encryption greatly increases the user's ability to quickly retrieve data from the cloud\cite{2}\cite{3}. The time and resource consumption, especially if the data volume increases. Therefore, how to effectively encrypt data without increasing excessive search burden becomes another focus of current cloud security research. On the basis of full homomorphic encrypted data indexes, this paper proposes a root index distance (RID) index algorithm based on the root index distance. By encrypting the root index, the depth of the index is reduced and the noise is reduced. The impact of this improves the accuracy of the data encryption index results and is independent of the amount of data. Finally, through analysis, it is proved that the RID algorithm greatly improves the data security without increasing the complexity.

2. Status Analysis for Cloud Data Security

2.1. Cloud Encrypted Data Retrieval

In the past, telecommunications networks were often represented by clouds in the diagram, and later used to represent the abstraction of the Internet and underlying infrastructure. Therefore, cloud computing can even allow you to experience 10 trillion operations per second. With such a powerful computing capability, you can simulate a nuclear explosion, predict climate change and market trends. Users access the data center through computers, laptops, and mobile phones, and perform operations according to their own needs. There are many claims about the definition of cloud computing. For what exactly is cloud computing, at least 100 explanations can be found. It is widely accepted that the National Institute of Standards and Technology (NIST) defines: Cloud computing is a pay-per-use model that provides available, convenient, on-demand network access. Configurable computing resource sharing pool (resources include network, server, storage, application software, service and so on), these resources can be provided quickly, with little management effort or little interaction with the service provider. Cloud computing includes the following levels of services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). This paper analyzes the data security issues of cloud computing systems and mainly considers the data security in the database as a data storage service (DAS) model.

In the DAS model, the enterprise outsources data to the cloud service provider for management. The cloud service provider performs storage management and security management of the data, processes the data request of the user, and returns the data required by the user to the user. In order to ensure the security of data, the user encrypts the data and stores it in the cloud server provided by the supplier. When the user needs to query the data, the user needs to send the query condition and request to the cloud server, and then the cloud server pairs the stored encrypted data.

When the encryption algorithm is too complex, the retrieval speed drops quickly and resources are consumed too much. When the encryption algorithm is too simple, the security of the data is seriously reduced. At present, some experts and scholars have begun to study the issue of retrieval of encrypted data. Song et al. proposed a linear retrieval scheme \cite{4} to perform encrypted information retrieval in a one-time method. This method is highly secure, but there is a slow retrieval speed when searching for massive data in cloud services. Problem; Swaminathan proposed a sorting search algorithm \cite{5}, but it does not apply to a query that contains multiple query conditions; the public key encryption search algorithm proposed by Boneh et al. is a kind of security index algorithm based on Boolean model \cite{6}, and it cannot sort according to the query conditions and the relevance of the data to be retrieved.

2.2. Homomorphism Encryption Data Retrieval

Full homomorphic encryption is also known as privacy homomorphism \cite{7} which is referred to as HEDR. It is an encrypted form that allows the user to perform operations on the encrypted data, and the result of the operation is also encrypted. This result is consistent with encrypting the result obtained by performing the same operation in plaintext. This cannot only guarantee the security and privacy of data, but also improve the efficiency of data processing. Therefore, full homomorphic encryption is considered to be the best way to solve cloud computing encrypted data retrieval\cite{8}\cite{9}. 


The fully homomorphic encrypted data retrieval algorithm considers both the user side and the cloud service provider. The user encrypts the search condition and then transmits it to the cloud service provider. The cloud service provider uses the homomorphic property to perform the user encrypted index condition. Operation, and send the encrypted result file to the user, and the user to decrypt the obtained encrypted file. During the entire data transmission process, the cloud service provider or the third party does not know the real data of the user, and thus can better protect the security of the user data. The definition of the full homomorphic encryption system is as follows:

Let \( P = (\text{Gen}, \text{Enc}, \text{Dec}) \), where \( \text{Gen} \) denotes a key generation algorithm for generating a public key and a private key in an encryption system; \( \text{Enc} \) denotes a data encryption algorithm; \( \text{Dec} \) denotes a data decryption algorithm. Let the plaintext space in the encryption system be \( M \), and the cipher text space be \( C \), the full homomorphic encryption system satisfies the following properties:

\[
\text{Dec}_a \left( f_1 \left( \text{Enc}_a \left( a \right), \text{Enc}_a \left( b \right) \right) \right) = a + b \quad (1)
\]

\[
\text{Dec}_a \left( f_2 \left( \text{Enc}_a \left( a \right), \text{Enc}_a \left( b \right) \right) \right) = a \cdot b \quad (2)
\]

Where, \( f_1 \) and \( f_2 \) denote functions that operate on the cipher text space \( C \), "\( + \)" and "\( \cdot \)" denote addition and multiplication operations for the plaintext space group respectively, and \( a, b \in M \).

Data encryption using HEDR generates a superimposable noise \( r \in R_s \) in a relatively small random space \( R_s \). To be able to decrypt the encrypted cipher text correctly, the noise must belong to a large random space \( R_l \), and \( R_s \subseteq R_l \):

\[
\text{Enc}_a \left( a, r \right) + \text{Enc}_a \left( a, r \right) = \text{Enc}_a \left( a + a, r + r \right) \quad (3)
\]

\[
\text{Enc}_a \left( a, r \right) \cdot \text{Enc}_a \left( a, r \right) = \text{Enc}_a \left( a \cdot a, r \cdot r \right) \quad (4)
\]

As shown in Equation (3) and Equation (4), since the introduced noise can be superimposable and unrestricted growth, then the range of "\( + \)" and "\( \cdot \)" must be controlled to avoid excessive noise.

In the cloud computing environment, although the HEDR algorithm is more suitable for cloud computing environment, it has the following two defects:

1) The amount of data in the cloud computing environment is huge, so the number of bits generated when indexing data blocks under this environment will also become larger and larger, eventually leading to excessive data index overhead, making the efficiency of encrypted data indexes.

2) As the amount of data increases, the HEDR algorithm will generate errors due to the presence of noise. When the cumulative error exceeds a certain threshold, the final result will become deviant or even wrong, so the amount of data to be encrypted and the noise must be controlled.

3. Principle and Implementation of RID Algorithm

3.1. Principle

For the characteristics of encrypted data retrieval in cloud environment, in order to avoid the defects of the above HEDR algorithm, a search algorithm RID based on the root index distance is proposed. The specific principles are as follows:

When the user queries the cloud server for the encrypted data file, it first calculates the index of the encrypted root, and the distance between other index files and the index root, and sends the whole to the cloud server; the cloud server calculates the user data query condition according to the encryption property. The corresponding root index value, combined with the data transmitted by the user, to obtain the corresponding encrypted data block stored in the cloud computing end for the index value corresponding to the final ciphertext data; the client uses the private key to encrypt the data block returned by the cloud server. Decrypt and recover the plaintext file. During the entire process, because the data is encrypted, the server cannot know the user's query condition information, nor can it directly contact the user's original data file stored on the cloud, ensuring data security.
3.2. Implementation

Assume that the user's file data block size is \( n \), the cloud server side stores the data block is \( d = d_0d_1d_2 \ldots d_{i-1} \), the user establishes the data index is \( x \in \{ x_0, x_1, \ldots, x_{n-1} \} \in \{0,1\}^m \), according to the data index classification, establishes the directory tree, and calculates the root index distance \( l_i \). When the user performs data query, the public key encrypted by the data, the index of the root directory of the query, and the distance of the index \( l_j \) from the root index are sent to the cloud server. For any \( i(0 \leq i \leq n-1) \), the cloud server does not know the plaintext of the data and retrieve the plaintext. The formula for calculating the encrypted data block on the data storage \( i \) is as follows:

\[
e_i = \begin{cases} 
Enc_{pk}(0), & i \neq x \\
Enc_{pk}(d_i), & i = x 
\end{cases}
\tag{5}
\]

Based on the properties of additive homomorphism, we have \( Enc_{pk}(d_i) = e_0 + e_1 + \ldots + e_{n-1} \), and return the final result as the user's search to the client \( Enc_{sk}(d_i) \). The client uses the private key \( sk \) to decrypt the encrypted data \( Enc_{pk}(d_i) \) and obtain the final plaintext \( d_i \). The RID algorithm execution process can be divided into three steps as shown in Figure 1, the specific process is as follows:

Step 1: A module is generated through the key algorithm \( Gen \), the user generates the public and private key pairs \( (pk, sk) \), and the root index \( {x_0, x_1} \) in the file data block. The root index will be encrypted: \( Enc_{pk}(x_0) \). When the user submits a data query request to the cloud server, the public key \( pk \), the encrypted root index \( x_0 \), and the index of the other file are sent together with the distance \( l_j \) from the root node to the cloud server.

Step 2: After the cloud server receives the search condition \( (pk, a_i, l_j) \) from the user query data, the public key \( pk \) is used to perform the following operations for \( x_0 \):

For \( \forall i(0 \leq i \leq n-1) \), perform the following calculations according to the properties of addition and multiplicative homomorphism of the full homomorphic encryption system:

\[
\beta_i = Enc_{pk}(eq_i(x)) \\
\beta_i = Enc_{pk}(x_0^i) \cdot Enc_{pk}(x_1^i) \cdot \ldots \cdot Enc_{pk}(x_{n-1}^i) \\
= Enc_{pk} \left( \prod_{i=0}^{n-1} x_i^i \right) 
\tag{6}
\]

where \( i \in \{ i_0, i_1, \ldots, i_{n-1} \} \in \{0,1\}^m \), if \( \beta_i = 1 \), then

\[
Enc_{pk}(x_0^i) = a_i = Enc_{pk}(x_0) 
\tag{7}
\]

If not:

\[
Enc_{pk}(x_0^i) = Enc_{pk}(1) + a_i Enc_{pk}(-1) \\
= Enc_{pk}(1 - x_0) 
\tag{8}
\]

The cloud server performs the following calculation based on the file index distance \( l_j \) submitted by the user:

\[
\beta_j = \beta_i + l_j \\
e_j = \beta_j \cdot Enc_{pk}(d_j) 
\tag{9, 10}
\]

where, \( e_j \) satisfies the formula (5). According to the properties of additive homomorphism, we will have:
The cloud server returns the cryptographic result of the index calculated by equation (11) to the user $Enc_{pk}(d_i)$.

Step3: The user obtains the final data plaintext based on the obtained file cipher text data $d_i$ and the equation (12).

$$d_i = Dec_{sk}(Enc_{pk}(d_i))$$

The server retrieves encrypted data blocks $Enc_{pk}(d_i)$ of user-required data based on the public key uploaded by the user while retrieving data, the root index encrypted by the private key, and the distance of other data $f_j$ with respect to the root index. After returning $Enc_{pk}(d_i)$ to the user, the user decrypts the encrypted data block $Enc_{pk}(d_i)$ using the private key he holds, and finally obtains the required data. During the entire process, the data and retrieval information stored in the cloud server and in the transmission process are all in cipher text. The cloud server and other third parties cannot obtain real user plaintext information, effectively reducing the noise impact of the retrieval data and ensuring the data security in a cloud server environment.

4. RID algorithm analysis

When retrieving the data in the cloud system, the retrieval conditions are passed to the cloud server, but the retrieval of the plaintext will leak the user’s data privacy, and the retrieval of the cipher text will increase the resource cost of the server and affect the accuracy of the retrieved data. The following section analyzes the RID algorithm from the aspects of security and algorithm performance.

4.1. Security Analysis

Before analyzing the security of the RID algorithm, it is assumed that the full homomorphic encryption algorithm running on the cloud server is correct, and the plaintext attack can be selected. The user and the cloud server are all honest, and the user and the cloud server will not actively divulge the keys that are in hand and perform the operations strictly in accordance with the algorithm’s flow.

In the RID algorithm, its security is based on the correctness of the calculation of $Enc_{pk}(eq(x))$ and the correctness of the full homomorphic encryption algorithm. At the same time, when calculating the index depth $\lceil \log_2 m \rceil = \log_2 n$ of $Enc_{pk}(eq(x))$, the RID algorithm only encrypts the root index, and accordingly reduces the depth of the index and reduces the effect of noise. While calculating the value of $e_0 + e_1 + \cdots + e_{n-1}$, although it will increase the computational load of the cloud server due to the noise reduction operation, it does not affect the accuracy of the data encryption result. Therefore, the RID algorithm retains the security of the HEDR encryption algorithm in response to the plaintext attack, and at the same time, encrypts the root index, reduces the depth of the index, reduces the impact of noise, and improves the accuracy of the data encryption index result.

4.2. Performance Analysis

When a user initiates a retrieval request for data from a cloud server, there are operations such as encryption and decryption on the local server. Similarly, there are calculation requirements on the server side. The performance of the RID algorithm is considered from both the server side and the client side.

1) Client complexity: The amount of computation required by the client is that encrypting the root index $(Enc_{pk}(x_0), Enc_{pk}(x_1), \cdots, Enc_{pk}(x_{m-1}))$ used by data query $x \leftrightarrow (x_0, x_1, \cdots, x_{m-1}) \in \{0,1\}^n$ $(m=\lceil \log_2 n \rceil)$. From the above formula, we can see that during the entire data encryption process, the encryption process of the root index has nothing to do with the data size, so the complexity of its
calculation is \( O(1) \), and it will not increase the resource overhead as the data size increases.

2) Server-side complexity: The index of server-side data needs to index all the files on the server. For any \( \forall i \leq n-1 \), the server-side is mainly based on the index distance parameters \( l_{ij(0 \leq i \leq n-1)} \) of other files submitted by the client to calculate \( e_i = Enc_{p,i}(d_i) \cdot Enc_{p,i}(eq_i(x)) \). The complexity of this part is \( O(n \log n) \). In the end, the complexity of \( Enc_{p,i}(d_i) = e_i + e_{i+1} + \cdots + e_{n-1} \) is \( O(n) \), therefore, the server-side computational complexity is \( O(n \log n) \). We can see that, RID algorithm increases the security of data flow without increasing the complexity of the algorithm.

5 Conclusion
For the special environment of cloud computing, this paper proposes a cloud-encrypted data retrieval algorithm RID based on the root index distance. Through the comprehensive analysis of the performance of RID algorithm, it shows that this RID algorithm did not increase the complexity, effectively eliminated the influence of noise and improved the security of data retrieval. The RID algorithm has very important significance for the retrieval of massive cloud-encrypted data. In the follow-up further research, the security of the algorithm in the event of a plaintext attack will be studied.

Acknowledgments
This research was partly funded by the Fund for “Integration of Cloud Computing and Big Data, Innovation of Science and Education”, project number 2017B06106; and by rule of law and theoretical research, ministerial level scientific research projects from Ministry of justice, project number17SFB3001; by scientific research project from East China University of politics and law.

References
[1] Feng Dengguo, Zhang Min, Zhang Hao, et al. Research on cloud computing security[J]. Journal of Software, 2011, 22(1): 71-83
[2] Liu Mingjie, Wang An. An Overview of the Research and Application of Fully-Hystirical Encryption[J]. Journal of Computer Research and Development, 2014,51(12):2593-2603
[3] Yu Zhibin, Zhou Yanhui. Keyword-based cloud encryption data privacy protection system [J]. Computer Science, 2015, 6A: 365-369
[4] SONG D, WAGNER D, PERRIG A. Practical Techniquesfor Searches on Encrypted Data[C]//Proceedings of the IEEE Symposium on Security and Privacy(S&P’00), May 14-17, 2000, Berkeley, CA,USA. Piscataway, NJ. USA:IEEE, 2000:44-55.
[5] SWAMINATHAN A, MAO Y, SU G M, et al. Confidentiality-Preserving Rank-Ordered Search [C]//Proceedings of the 2007 ACM Workshop on Storage Security and Survivability(StorageSS’07), Oct 29, 2007, Alexandria, VA, USA. New York, NY, USA: ACM, 2007:7-12.
[6] BONEH D, CRESCENZO G, OSTROVSKY R, et al. Public Key Encryption With Keyword Search [C]//Advances in Cryptology. Proceedings of the 23rd Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT’04), May 2-6, 2004, Interlaken, Switzerland. LNCS 3027.Berlin, Germany: Springer-Verlag, 2004:506-522.
[7] Atallah M J, Du Wenliang. Secure multi-party computational geometry[G]//Proceedings of the 7th International Workshop on Algorithms and Data Structures. Berlin: Springer, 2001: 165-179.
[8] Li Lang, Yu Xiaozhong, Yang Yaqiong, et al. Review of research progress on homomorphic encryption [J]. Journal of Computer Applications, 2016, 32(11): 3209-3214.
[9] Xu Peng, Liu Chao, Le Xueming. Fully homomorphic encryption algorithm based on integer polynomial rings [J]. Computer Engineering, 2012, 38(24.)