An Accountability Scheme for Cloud Storage Leakage

Yin-dong CHEN*, Xiao-jin XU and Liu ZHANG
College of Engineering, Shantou University, Shantou, China
*Corresponding author

Keywords: Cloud storage server, Accountability, Encrypt, Leakage.

Abstract. With the rapid development of information technology, the privacy and security of cloud data is the problem that the cloud users are facing, when cloud data is leaked, the responsibility is hard to judge, it is easy for cloud storage server to shift responsibility away. The article proposes an accountability scheme, which is to store the information of cloud storage server on the data and then encrypt it multiple times. It is a great way to determine the responsibility of cloud storage server for data leakage. At the same time, the scheme can realize the integrity verification of cloud data.

Introduction

With the development of science and technology, the data security problem has become the primary concern for users [1]. The current research on accountability of cloud storage server is mostly in the stage of proposing a concept, meeting some of the requirements for accountability and building some accountability models [2], the purpose of accountability can be achieved by embedding watermarks in data [3]. We can also use the credential mechanism to achieve the purpose of accountability by looking at the log records of the file operations (the premise is that the log records cannot be tampered with) [4-5]. This paper presents an accountability scheme that adds cloud storage server’s information to the stored data and then encrypt it multiple times, without distortion, the data can be held accountable for the leakage of users’ data. At the same time, the scheme can realize the integrity verification of cloud data.

Relevant Theoretical Basis

Hash Function

Hash function [6] plays an important role in modern cryptography, it can be considered as a single password system, the Hash operation is an irreversible mapping, just an encryption process that cannot be decrypted.

The Digit Summary

The digit summary [7] uses a one-way hash function to encrypt the plaintext into a string of 128-bit ciphertext, which is also called a digital fingerprint, it has a fixed length, the digit summary of different plaintext will be inconsistent.

The Digital Signature

The Hash function is used to make a digit summary of the file, and then the digit summary is encrypted with the private key, which is the digital signature. [8] Then the file and the digital signature are sealed together to get signature result, the signature result is sent to the receiving party. The receiving party decrypt the digital signature with the sender’s public key to get the digit summary. Then make a Hash algorithm with the received file to get a new digit summary. The receiving party compares the two digit summaries, if two digit summaries are the same, the digital signature is verified, or it is not valid.
The Asymmetric Cryptography

The asymmetric cryptography [9] has two keys to form a key pair, one for the key owner, does not involve the distribution problem, the other can be open, based on the open channel can be distributed, greatly improving the distribution Convenience.

Scheme Overview

Problems that Need to be Solved

For users to store data in the cloud storage server, the two parties may have the following questions:

1) Whether the user's data stored in the cloud storage server is complete.
2) Whether the data is tampered or lost by cloud storage server, and so on, if so, whether the cloud storage server will deny the above acts or not.
3) Due to the competition between different cloud storage server, some cloud storage server may disguise as users to frame up other cloud storage server, they can put the data stored in other cloud storage server, and then take the initiative to leak out the data, shift responsibility of leaking data to other cloud storage server. The cloud storage server knows that there is a possibility that users will leak their data, so it can be used as an excuse to deny it when the cloud provider really leaks the user's data. Therefore, when data stored in the cloud is leaking, it is difficult to determine who should be held accountable.

The Cloud Storage Architecture

As is shown in Figure 1. By referring to the basic architecture of the distributed File System [3] GFS (Google File System), this paper presents a cloud storage architecture. The cloud storage architecture contains three types of entities: Client, trusted third party server, and cloud storage server(we use CSS as logogram for cloud storage server and use TTS as logogram for trusted third party server). Clients store vast amounts of data in the cloud, relying on cloud storage server to maintain data and provide services. Client can be a person user or an organization user. TTS is a trusted medium between Client and CSS, and is also a notary for accountability when the cloud data is leaked. CSS can store the user's primary data. In the cloud storage architecture, there are multiple Clients and multiple cloud storage servers, only one trusted third party server. The trusted third party server will verify the files periodically toward cloud storage server to confirm whether cloud storage server has modified the data, which is to verify the integrity of the cloud data.

Related Instructions

1) When the data transmits in the communication channel, the data has the possibility of being stolen and tampered with, such a channel is not credible, we can use the digital signature technology [8] to ensure that the data will not be tampered in the process of transmission, so that the communication channel can be trusted. In order to facilitate the discussion of the scheme, the proposed scheme is based on the trusted of communication channel. 2) In the scheme, the cloud storage server negotiates with trusted third party server to generate important information ID_B, we
use ID_B to represent the information of the cloud storage server, which can prove that the file has been leaked from the cloud storage server. When a cloud storage file is leaking, we can take responsibility for cloud storage server by ID_B. 3) In the scheme, the trusted third party server negotiates with Client and cloud storage server to generate tags MsgA_1, MsgA_2, ... and MsgB_1, MsgB_2, ... . We use MsgA_1, MsgA_2, ... to distinguish different Clients and use MsgB_1, MsgB_2, ... to distinguish different cloud storage servers(In the cloud storage architecture, there are multiple Clients and multiple cloud storage servers, only one trusted third party server). These tags do not have the role of accountability. 4) In the scheme, the data encryption algorithm, the information of the cloud storage server ID_B and the public key of asymmetric encryption have been exposed, but the private key is not exposed. 5) In the scheme, the trusted third party server will verify the files periodically toward cloud storage server to confirm whether cloud storage server has modified the data, which is to verify the integrity of the cloud data.

**Scheme Realization**

**Prestorage Stage**

As is shown in Figure 2.
1) The Client sends request for storage to the CSS.
2) The CSS sends request for storage to the TTS.
3) The Client sends request for storage to the TTS.
4) The TTS sends response to the Client when the TTS receives storage requests from both Client and CSS.

When the Client receives response from TTS, the storage stage begins.

![Figure 2. Prestorage stage.](image)

![Figure 3. Storage stage.](image)

**Storage Stage**

As is shown in Figure 3.
1) The Client sends the raw data to the TTS.
2) F_1 = k_cs(Data, ID_B)
3) F_2 = k_ap(F_1)
4) F_3 = (F_2, MsgA_1)
5) F_4 = Hash(F_3)
6) F_5 = k_cs(F_3, F_4, ID_B)
7) F_6 = Hash(F_3)
8) Compare(F_4, F_6)

![Figure 4. Integrity verification stage.](image)

![Figure 5. Accountability stage 1.](image)
2) The TTS attaches the information of CSS ID_B to the raw data, and encrypts them with its own private key k_{cs} to get file F_1.
3) The TTS encrypts file F_1 with Client’s public key k_{ap} to get file F_2.
4) The TTS attaches the tag of Client MsgA_1 to the file F_2 to get file F_3.
5) The TTS generates a hash value of the file F_3 with a hash function to get file F_4.
6) The TTS encrypts F_3, F_4, ID_B with its own private key k_{cs} to get file F_5.
7) The TTS sends file F_5 to CSS.

**Integrity Verification Stage**

As is shown in Figure 4.
1) The TTS sends request for file F_5 to the CSS.
2) The CSS sends file F_5 to the TTS.
3) The TTS decrypt file F_5 with the TTS’s public key k_{cp} to get file F_3, F_4, ID_B.
4) The TTS generates a hash value of the file F_3 with a hash function to get file F_6.
5) The TTS compares the content of file F_4 and the content of file F_6, if the content are the same, the content of file F_5 is complete, so the raw data is complete.

**Accountability Stage 1**

As is shown in Figure 5. When the file F_4 stored in the CSS was leaked, TTS is able to get the CSS’s information ID_B from the file F_5 to prove the responsibility of the CSS, and to hold the CSS accountable.

The proof process is as follows:
1) The TTS decrypts the file F_5 with its own public key k_{cp} to get file F_3, F_4, ID_B.

![Figure 6. Accountability stage 2.](image)

![Figure 7. Retrieve stage.](image)

**Accountability Stage 2**

As is shown in Figure 6. The cloud storage server can decrypt files F_3, F_4, ID_B(file F_4 is a hash value and ID_B is a tag, they have not practical significance when they be leaked). When the file F_3 stored in the CSS was leaked, TTS and Client is able to get the CSS’s information ID_B from the file F_3 to hold the CSS accountable.

The proof process is as follows:
1) The TTS get tag MsgA_1 and file F_2 from file F_3.
2) The TTS sends file F_2 to Client(A_1) according to the tag MsgA_1.
3) Client decrypts the file F_2 with its own private key k_{as} to get file F_1.
4) Client decrypts the file F_1 with TTS’s public key k_{cp} to get raw data and ID_B.

**Retrieve Stage**

As is shown in Figure 7.
1) The Client sends request for file F_5 to the CSS.
2) The CSS sends file F_5 to the Client.
3) The Client decrypts file F_5 with TTS’s public key k_{cp} to get file F_3, F_4, ID_B.
4) The Client get file F_2 and tag MsgA_1 from file F_3.
5) The Client decrypts file $F_2$ with its own private key $k_{ats}$ to get file $F_1$.
6) The Client decrypts file $F_1$ with TTS’s public key $k_{cp}$ to get raw data and $ID_B$.

**Scheme Analysis**

**Determine the Responsibility for the Behavior of Cloud Storage Server Leaking File**

The cloud storage server can decrypt files $F_5$ by trusted third party server’s public key $k_{cp}$ to get file $F_3$, $F_4$, $ID_B$ (cloud storage server cannot decrypt file $F_2$, because cloud storage server cannot get Client’s private key $k_{ats}$ to decrypt file $F_2$). There is no practical significance for the leakage of file $F_4$ and $ID_B$, but if the file $F_3$ or $F_5$ is leaked by cloud storage server, we can get message $ID_B$ by decrypting file $F_3$ or $F_5$, so as to investigate the responsibility of cloud storage server.

**Prevent Cloud Storage Server from Tampering with File F5**

If the cloud storage server tampers with the file $F_3$ after he received the file $F_5$, the trusted third party server will find that there is no hash relationship between the content of file $F_3$ and the content of file $F_5$ when he checks the files periodically toward cloud storage server, so trusted third party server will discover that cloud storage server has tampered with the file $F_5$.

**Prevent Cloud Storage Server from Counterfeiting File F5**

If the cloud storage server causes the file $F_5$ to be damaged for some reason, cloud storage server may want to counterfeit another file $F_5'$ to pass the check of trusted third party server. However, the production of file $F_5$ requires trusted third party server’s private key $k_{cs}$, cloud storage server cannot get $k_{cs}$ to counterfeit another file $F_5'$.

**Prevent Client from Framing up Cloud Storage Server**

Due to the competition between different cloud storage server, some other cloud storage server may disguise as Client to store data in cloud storage server. Suppose that trusted third party server’s algorithm for encrypting raw data is public, Client will try to counterfeit the file $F_3$ or $F_5$ and leak them out intentionally, which will cause the Client to be held accountable. However, the production of file $F_3$ and $F_5$ requires trusted third party server’s private key $k_{cs}$, Client cannot get $k_{cs}$, so file $F_3$ or $F_5$ cannot be counterfeited. The cloud storage server cannot be framed.

**Secure User Data**

Even if the file $F_3$ or $F_5$ is leaked, the plaintext of the file can’t be obtained by others, which protects the security of user’s data.

**The Efficiency of Data Transmission is High**

When Client retrieves the file $F_5$ from cloud storage server, the Client can decrypt the file $F_5$ directly to obtain the raw data. The data does not need to pass through trusted third party server in the process of transmission, which has high transmission efficiency.

**Summary**

The article proposes an accountability solution, which is to store the information of cloud storage server on the data and then encrypt it multiple times. With the help of trusted third party server, when data is leaked, the scheme can determine the responsibility for the behavior of cloud storage server leaking file. At the same time, the scheme can prevent cloud storage server from counterfeiting file or tampering with file and prevent Client from framing up cloud storage server. Even if the file is leaked, the plaintext of the file can’t be obtained by others, which protects the security of user’s data. The efficiency of data transmission in the scheme is high.
Acknowledgement

This paper is supported by the National Natural Science Foundation of China (No. 61103244, 61702318), the Guangdong Natural Science Foundation (No. 2015A030313433), the Characteristic Innovation project in Higher Education of Guangdong (No. 2015KTSCX036, 2016KQNCX056), the Science and Technology Planning Project of Guangdong Province (No. 2016B010124012, 2016B090920095), the Engineering and Technology Research Center of Guangdong Higher Education Institutes (No. GCZX-A1306).

References

[1] SA. Hande, SB. Mane. An Analysis on Data Accountability and Security in Cloud [C]. International Conference on Industrial Instrumentation and Control, 2015:713-717.

[2] Fan Zhen-hua. Researches on Data Integrity Verification and Accountability Mechanisms of Cloud Storage [D]. Hebei: Hebei University, 2014. (in Chinese)

[3] Ting-ting Liu. Research on Key Technologies of Data Security towards Cloud ComPuting [D]. Henan: Information Engineering University, 2013. (in Chinese)

[4] Wang Yan. Design and Implementation of a Secure and Accountable Cloud Storage Scheme Based on Trusted Third Party [D]. Beijing: Beijing University of Posts and Telecommunications, 2012. (in Chinese)

[5] Chen Bing-quan. Research on Cloud storage system accountability mechanism [J]. Information Technology and Informatization, 2015, (1): 57-60. (in Chinese)

[6] Chen Ai-lin, Li Bin-yong. Research on Privacy Protection By One-to-Many Reliable Communication Based on RSA and Hash [J]. Cyberspace Security, 2017, 08 (08-09):28-31. (in Chinese)

[7] Wang Gui-hong. E-commerce security control [J]. Computer & Network, 2014, 60-63. (in Chinese)

[8] Wu Jun-xi. Application of encryption technology in network communication [J]. Network Security Technology & Application, 2015, 1:47-50. (in Chinese)

[9] Lin Ning, Zhang Liang. Preliminary exploration of information encryption technology [J]. Information & Communications, 2015, 156 (12):149-150.