Identity Based Reliable Data Sharing with Revoked System in Cloud

Sneha D Raut[1], Prof. Nagaraju Bogiri[2]

[1]K. J. College of Engineering and Management Research, Savitribai Phule, Pune University Pune, Maharashtra
[2]K. J. College of Engineering and Management Research, Savitribai Phule, Pune University Pune, Maharashtra

Abstract
In the storage services of the cloud, users can be store data on the cloud and perform exchange of data with others. Cloud uploaded files contains sensitive information. At the time of file sharing, may realizing the sensitive information of files hide, but this shared file cannot be used by others. The problem is to how to perform data sharing with the sensitive information. For solving such problem, propose a Revocation algorithm to avoid the use of a sensitive file from unauthorized users. First, in the system, by using sanitizer the data blocks are sanitizes and the binded the data blocks and at the end of sanitized data blocks after that sends this sanitized file by sanitizer and its corresponding signature to the cloud. The integrity of sanitized files is verified by using these signatures during the integrity audit phase. TPA then verifies the integrity of the sanitized file. This way, our proposed system provides security to sharing data.
Keywords: Cloud Storage, sensitive information hiding, Revocation Algorithm and Data integrity auditing

1. Introduction

In explosives growth of the data, it is a huge burden on users to store all data on the cloud. However, if data is storing on the cloud, it will be damaged or lost as a result of software, hardware failure, and human error in the cloud. We will continue to propose many remote data integrity audit scheme, for the cloud correctly can save data, to verify whether or not. Generate fist to the needs of the owner of the data on the private cloud of the file before uploading. These signatures audit the health of the phase of the data block with a sense of the cloud of the density. Then, the owner of the data is a signature work within the data block. The accumulated data in cloud will include large number of users, but this does not apply to cloud storage applications. The cloud storage has a very important feature called data sharing. The data stored on cloud may contain some sensitive information [1].

Cloud computing helps companies improve the creation and delivery of IT solutions. Clouds can be divided into three categories Public, private and hybrid cloud depends on accessibility restrictions and deployment models. Many cloud storage audit protocols have been proposed to develop technologies. To reducing the computational load on the client, TPA (Third-Party Auditor) [1] [2] [3] [4], has been introduced, it helps to client to periodically checks the integrity of data on the cloud. All the existing protocols is focused on the cloud failure or fraud, procedure for dealing with client private key disclosure for security weakness and security cloud storage audit is a very important issue for clients. This process includes downloading the entire data from cloud storage, creating a new authentication system, and re-uploading all the data on cloud [3].

Another important issue in cloud storage auditing is the recent review of key is-sues. The problem itself is not self-evident in nature. When a client's private key for the storage auditing is exposed to the cloud, the cloud can easily hide data loss incidents to maintain its reputation [4]. For solving this problem, here propose a Revocation algorithm to avoid the use of a sensitive file from unauthorized users. First, in the system [1], sanitizer sanitizes data blocks and blinded blocks of data blocks, and at the end of sanitized data blocks after that it sends sanitized files by sanitizer and it is correspond to signature on the cloud. The integrity of sanitized files is verified by using these signatures during the integrity audit phase. TPA then verifies integrity of sanitized file. This way, our proposed system provides security to sharing data.

Our contribution: We propose a Revocation algorithm to avoid the use of sensitive files from unauthorized users. This allows the owner at any time to directly revoke any user from cloud servers and to distinguish unauthorized users, because the security model of the cloud service allows you to share data and protect the data of trusted users. User cancellation is the most difficult, clouds in which a single user revocation affects other users who share a common attribute space.

As a result, we are reducing the cost of data sharing services provided by cloud computing. Mobile cloud computing is a mobile smart terminal used from anywhere, when the cloud of convenience, data accessible to the user. The user stores personal data on the cloud server, and it is easy to share only with authorized customers who have access to the data. A lot of traditional systems provide access control for users of the service, a service based on user attributes. Today, the business is usually outsourcing to the cloud with shared encrypted data for users and, moreover, cloud services are new cloud service partners to respond to in its traditional access control system distributed in different geographical areas. In addition, the cloud server was completely reliable, largescale users from different domains did not leak the contents of the data collusion correspondence with these data leaks is difficult, and the user can cancel the scheme for multiple users to access the cloud service. For solving this problem, here design Revocation algorithm for multiple user accessing in cloud service.

2. Literature review

The author proposes techniques for auditing integrity of remote data that enables you to share data with confidential information hidden. In this used sanitizer for the block data corresponds for the confidential information of file, and these blocks of data on the valid signature for the sanitized File are used to encrypt the file. Using this signature to, verify sanitized file integrity in the integrity auditing phase. These techniques are capable of ensuring the storage and exchange of files in the cloud and also to hide confidential information. This technique is based on Cryptography-based identity [1].

For group user-author proposes the cloud storage audit scheme. It is used to reducing computational load on user side. This method introduces TPM (Third-Party Media). It performs lengthy operations on behalf of the user. Where, TPA is responsible to generate authenticators to the user and TPA on behalf of user to verifying the integrity of data. In that way, data is protected from TPA. Simple operation to blind data Phase data uploads, data audit [2].

In this article, the author focuses on auditing of cloud storage. The author study on the client's key damage for exposure: how to reduce in the cloud storage auditing and set solution on these problems. In this paper, the definition of auditing protocol and the security model are formulated using the key Exposure resilience and this protocol is proposed. In this they design binary tree structure and a leading traversal technique are adapted to update the client's secret key. In development of such a new certification construction will help to ensure the security and reliability of block less certification [3].

In this article, the author will explain how to update the key as a key update, and how to ensure the transparency of the new paradigm proposed by the client, called cloud storage auditing and verification of outsourcing keys. In this key renewal is safely entrusted to authorized parties, and burden of keys renewal on client is particularly acute. It leverages the existing many auditing designs and serves as the authorized party in this design, to ensure that the storage auditing. In this design, the TPA is to retain the encrypted client's private key, all these tedious work on behalf of the client these remarkable features are all carefully designed to make the entire audit process of the key exposure resistance as transparent for the client. Also formulates definition of this paradigm in the security models. Detailed design instantiations that can minimize safety certification and performance simulation are safe and efficient [4].

Cloud storage services have been widely adopted by more and more institutions, through the user's convenience per share data. However, the anonymity of unconstrained identities leads to new problems, that is since members of the group changed shared data without maliciously identified. Malicious modifications share data to hold the usability sharing data of wreck, and also the attribution of the identity. In this paper, the author proposes efficient public auditing of the preservation of privacy and identity traceability of group members at the same time. The system also allows you to achieve data during your privacy authentication generation by using the blind signature technology. Using the proposed method, the auditing system for the actual scenario is designed further [5].

They present a new public audit technique for integrity of the shared data with the user's efficient cancellation in mind. For utilize idea of quitting the proxy's, you can resign all blocks on the behalf on existing user upon user revocation, and then re-sign the existing user. In addition, even if a portion of the shared data is re-signed by the cloud, the public verification function, without obtaining all entire data on the cloud, it is possible to protect the shared data. In addition, the organization provides a comprehensive audit of the audit work at the same time [6].

Data per user group and share in cloud storage services. Given the fact that the cloud cannot be trusted, when users need to calculate the signature of a block of shared data to enable public integrity audit, the user has signed a block previously from the group of revoked, this revoked users sign updates and existing users[7].

This paper proposes an proxy oriented identity-based data upload.The formal definition of a system model is the security model. Next, they design a specific ID-PUIC protocol using a bilinear combination [8].

3. Proposed Approach

3.1 Problem Statement

The files on the cloud may contain sensitive information. The problem is how to perform data sharing with sensitive information. In order to solve this problem, we propose a Revocation algorithm to avoid the use of a sensitive file from unauthorized users.

3.2 Proposed System Overview

A detailed description of the proposed system architecture is as follows:

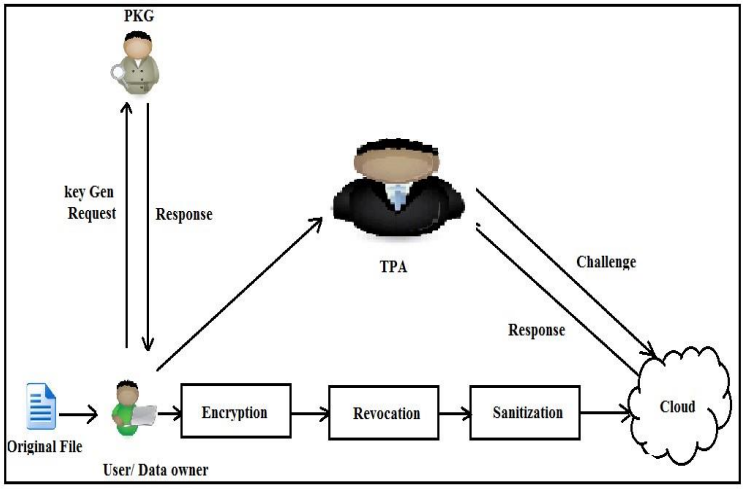

Figure 1. Proposed System Architecture

The cloud: On the cloud, users save a lot of data. Users upload data and share data with other users.

Sanitizer: The sanitizer, which is responsible for sanitizing data blocks that correspond to sensitive information in files (such as personal and organizational information), is responsible for sanitizing data blocks.

The Public Key Generation (PKG): Other entities rely on the generation of Public Key. The functionality of this PKG is the generation of public attributes and private key for users according to the identity ID.

Revocation: Whenever data owner wants to share reliable data with some selected authorized user from the set of all authorized user of organization, we use revocation algorithm to revoke the particular set of user as input then select the encrypted reliable data and time stamp used for at what time period revoked user is removed from organization, then we update the list of authorized and sent it to the server with encrypted reliable data.

The Third Party Auditor (TPA): TPA is a public verifier. It checks integrity of data stored on the cloud on behalf of users.

On cloud, users save a lot of data. The sanitizer [1], which is responsible for sanitizing data blocks that correspond to sensitive information in files (such as personal and organizational information), is responsible for sanitizing data blocks. Other entities rely on the generation of Public Key. The functionality of this PKG is the generation of public parameters and private key for the user according to their identity ID. Whenever data owner wants to share reliable data with some selected authorized user from the set of all authorized user of organization, we use revocation algorithm to revoke the particular set of user as input then select the encrypted reliable data and time stamp used for at what time period revoked user is removed from organization, then we update the list of authorized and sent it to the server with encrypted reliable data. TPA is a public verifier.
It checks the integrity of data stored in the cloud on behalf of users.

In this system first blinds sensitive information data from files and then generate the corresponding signature for those blinding files. Here, signatures are using to the verifying integrity of the sanitized files during the integrity audit phase. In the revocation system, we can add some user those who cannot able to download that sensitive file called revoked user. When user makes a request to download the file from the cloud: cloud firstly checks requested user if the revoked user if it is revoked user then the cloud will not allow downloading that file. If it is not revoked user then he/she can download the file using the private key. In addition, we add the validation period to the file.

3.3 Algorithm

Algorithm 1: Setup

I/P = {K};
Where,
K= Security Attributes
O/P = {Sk, Y, Pa}
Where,
Sk = Secret Key
Pk = Public Key
Pa= System Public Attributes

Algorithm 2: Extract

Input= {Pa, Sk, ID};
Where,
Pa = System Public Attributes
Sk = Master Secret Key
ID= User Identity
Output= {UID};
Where,
UID= User private key

The user verifies the correctness of UID and it accepts the private key, if it passed to verification.

Algorithm 3: Signature generation algorithm

The probabilistic algorithm runs by User ID. User private key UID, User signing private key Sk and a file F, File identifier name n as input, . It outputs a blinded file F*.
Input= {F, UID, sk, n}
Where,
F= Original file UID= User Private Key sk = User Signing
Private Key n= File Identifier Name
Output= {F*} \\ blinded file corresponding signature

Algorithm 4: Sanitization

The input is blinded files F* and its signature set and the outputs is sanitized file F1 and its signature.
It's run by sanitization.
Input= {F*, Φ1}; Where,
F*= blinded file
Φ1= signature set
Φ1{F1,Φ1 } Where,
F1= sanitized file
Φ1= corresponding signature set

Algorithm 5: Proof Generation Algorithm

This algorithm runs by cloud.
Input= {F1, Φ, C}
Where,
F1= sanitized file
Φ= corresponding signature set
C= Auditing challenge
Output= {P};
Where,
P= Auditing proof

Algorithm 6: Verification Algorithm

Input={C, pa, P}
Where,
C= Auditing challenge
Pa= system public parameters
P= Auditing proof
Output= {Fp}
Where,
Fp= correctness of proof P

Algorithm 7: Revocation

Revocation algorithm to revoke the particular set of user as input then select the encrypted reliable data and time stamp used for at what time period revoked user is removed from organization, then we update the list of authorized and sent it to the server with encrypted reliable data.
Input: List of the user for revocation
Output: Revoke user from an organization

Process:
Start
Take encrypted File and TimeStamp
Remove selected user from the organization for
The selected Timestamp
Update list of user
Upload updated list of user to a server with
Encrypted
End

3.4 Mathematical Model

System S can be defined as:

1) User

S = {U, K, C, DA, P}
Where,
U = {UI, UF, UO} is A User
UI= {UI1, UI2} ,A set of input
UI1-User Authentication details
UI2-File Data

UF= {UF1, UF2, UF3, UF4}, A set of functions

UF1=User Registration
UF2=User Login
UF3=key Generation Request
UF3=File Encryption
UF4=File Upload
UF5=Verification Request

UO= {UO1, UO2, UO3}\\ A set of output
UO2=File Blocks {b1, b2..,}
UO3= Verification result

2. KDC:

KI= {UA, KF, KO}
KI= {UA} User Attributes
KF= {KF1, KF2, KF3}
KF1=public Key Generation
KF2=Private Key Generation
KF3=Generated Key Response
KO= {KO1, KO2}
KO1=Public and Private Key

3. Cloud:

CI= {CI1, CI2}, A set of input
CI1-Block Data
CI2-Challenge Message
CF= {CF1 CF2} \\ A set of functions
CF1=Save Blocks and its hash
CF2=Generate Responce

CO=Output of cloud
CO= {CO1, CO2}
CO1=Set of Blocks = {b1, b2.., bn}
CO2=Generated Response of File block

4. Data Auditor:

DAI= {DAI1, DAI2}\\ A set of input
DAI1-User Registration Details
DAI2-Block Details

DAF= {DAF1, DAF2}\\ A set of functions
DAF1=Generate Challenge
DAF2=Verify Proof

DAO= {DAO1, DAO2} \\A set of output
DAO1=Challenge Message
DAO2=Verification Message

5. Revocation:

P = {PI, PF, PO}
L= {L1, L2}\\ A set of Input
L1= List of user
L2= Revoke list of user

R= {R1, R2} \\ A set Of Function
R1 = Revoke User
R2 = Secret key updating

O = {OP} \\ A set Of Output
OP= List of revoked user

4. Results and Discussion

4.1 Experimental Setup

This system is built use of Java framework and uses Net Bean IDE as a development tool. We use a number of files in different size.

4.2 Result and Analysis

The figure 2 shows that auditing computation overhead of the proof verification, proof generation and challenge generation, these three procedures linearly increases with number of challenged blocks. The verification procedure cots takes longer than the other two phases, and the challenge generation procedure cost takes the shortest time among the three procedures. In this graph, all the blocks will be challenged, and these three stages will require a large amount of computational overhead.

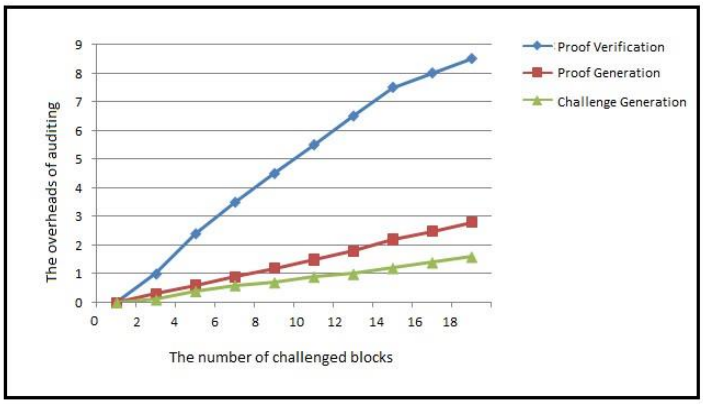

Figure 2. Computation Overhead in Auditing Phase

Table 1 shows, the time comparison between the existing and proposed system algorithm. Figure 3 shows the time require to access data of the proposed techniques with existing techniques. It shows that the data accessing time of the proposed Revocation algorithm is less because of data sharing with selected authorized users and there is no traffic in the data accessing process. Suppose, the hundreds of users send a request to the admin to access the file, the admin gives the permission only authorized 60 users. So the less time requires accessing the file and there is no traffic in the data accessing process.

Table 1. Time Comparison

Algorithm	Time in ms
Data access time without Revocation	90
Data access time with Revocation	40

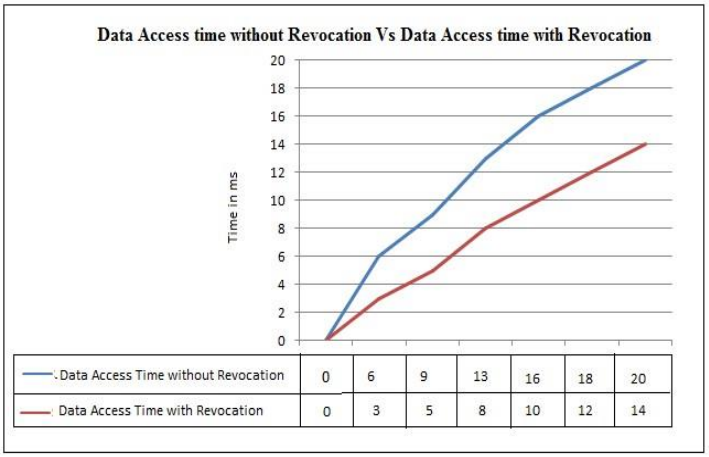

Figure 3. Data access time without Revocation Vs Data access time with Revocation

Table 2 shows, the performance Comparison. Figure 3 shows that, if number of users increased performance of the system with revocation and without revocation. The graph shows that the performance of the system with revocation is better than the performance of system without revocation.

Table 2. Performance Comparison

System	No of Users
Performance of system without Revocation	85
Performance of system with Revocation	90

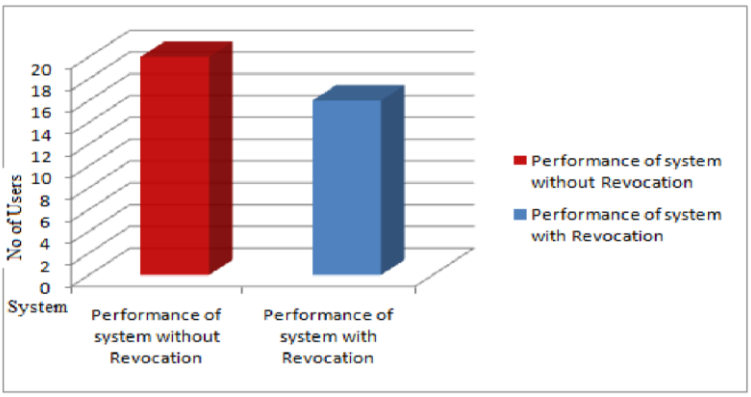

Figure 4. Performance graph

5. Conclusion

We propose a Revocation algorithm for reliable data sharing that supports data sharing with the sensitive information hiding. This supports data sharing with selected authorized users within organization. Data accessing time of the proposed Revocation algorithm is less because of data sharing with selected authorized users and there is no traffic in data accessing process. The results show that the proposed system provides security to data sharing.

References

6.1 Journal Article

[1] jia yu, wentingshen, jing qin, ronghao, and jiankun hu, "enabling identity-based integrity auditing and datasharing with sensitive information hiding forsecure cloud storage", [2018].
[2] h. zhang, x. lu, w. shen, h. xia, j. yu, and r. hao, "light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium,", [2017].
[3] k. ren, c. wang, j. yu and v. varadharajan, "enabling cloud storage auditing with key-exposure resistance," IEEE [2015].
[4] k. ren, j. yu and c. wang, "enabling cloud storage auditing with verifiable outsourcing of key updates," [2016].
[5] w. shen, g. yang, q. su, z. fu, j. yu, and r. hao, "enabling public auditing for shared data in cloud storage supporting identity privacy and traceability," [2016].
[6] b. li, b. wang and h. li, "panda: public auditing for shared data with efficient user re-vocation in the cloud," [2015].
[7] d. wang, m. xu, s. fu y. luo and j. deng, "efficient integrity auditing for shared data in the cloud with secure user revocation," [2015].
[8] d. he, h. wang, and s. tang, "identity-based proxy oriented data uploading and remote data integrity checking in public cloud," [2016].
[9] d. xie, j. li, j. li and z. cai, "secure auditing and deduplicating data in cloud," [2016].
[10] d. koo, j. hur, y. shin and k. kang, "secure data deduplication with dynamic ownership management in cloud storage," [2016].