International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 03 | March - 2023 Impact Factor: 7.185 ISSN: 2582-3930

DATA SHARING SCHEME FOR BIG DATA BASED ON USER REVOCATION

Mr. Arul raj N K[1], Dr. Isaac Sajan R[2], Mrs. Joselin Kavitha[3], Mr.Adlin Gold[4]

[1] Assistant Professor, ECE, Ponjesly College of Engineering, Nagercoil.
[2] Professor, ECE, Ponjesly College of Engineering, Email: isaacsajan@gmail.com
[3] Assistant Professor ECE, Marthandam College of Engineering & Technology, Kuttakuzhi.
[4] PG Scholar, Ponjesly College of Engineering, Nagercoil.

ABSTRACT

As cloud provides various services number of user stores their data on cloud. Integrity of cloud data is important as number of user shared data on cloud. User revocation is common in such schemes, as user's membership changes for variety of purpose. Previously, user revocation overhead in such schemes was related with the overall different file blocks occupied by a revoked user. Remote information integrity checking permits an information storage server, says a cloud server, to control a character that it's really storing owner's data honestly. Up till now, various Remote information integrity checking protocols are planned, however most of the concept suffer from the difficulty of a fancy key organization, that is, they rely on the expensive public key infrastructure which could hamper the preparation of Remote information integrity checking in observe. This project, have a tendency to propose a replacement construction of identity-based (ID-based) Remote information integrity checking protocol by creating use of key homomorphic cryptanalytic primitive to remove the system complexness and also the value for establishing and managing the general public key authentication framework.

1. INTRODUCTION

1.1 CLOUD COMPUTING

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. The services themselves have long been referred to as Software as a Service (SaaS). The datacenter hardware and software is what will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the general public, call it a Public Cloud; the service being sold is Utility Computing. Use the term Private Cloud to refer to internal datacenters of a business or other organization, not made available to the general public. Thus, Cloud Computing is the sum of SaaS and Utility Computing, but does not include Private Clouds. People can be users or providers of SaaS, or users or providers of Utility Computing.

Cloud computing is regarded as such a computing paradigm in which resources in the computing infrastructure are provided as services over the Internet. The cloud computing benefits individual users and enterprises with convenient access, increased operational efficiencies and rich storage resources by combining a set of existing and new techniques from research areas such as service-oriented architectures and virtualization. Although the great benefits brought by cloud computing are exciting for users, security problems may somehow impede its quick development. Currently, more and more users would outsource their data to cloud service provider (CSP) for sharing. These security matters existing in public cloud motivate the requirement to appropriately keep data confidential. Several schemes exploiting cryptographic mechanisms to settle the security problems have been proposed. The data owners could broadcast their encrypted data to a group of receivers at one time and the public key of the user can be regarded as email, unique id and username. Hence, by using an identity, data owner can share data with other group users in a convenient and secure manner.

a, Cloud type and services providers

Cloud computing technologies promise to change the way businesses operate. Cloud services have made it possible for organizations to operate across institutional boundaries. This has led to overcoming the physical barrier present in isolated systems. Cloud services and virtualization are driving major shifts in IT spending and deployments. Cloud services give companies the flexibility to purchase infrastructure, applications, and services, from third-party providers with the goal of freeing up internal resources and recognizing cost savings.

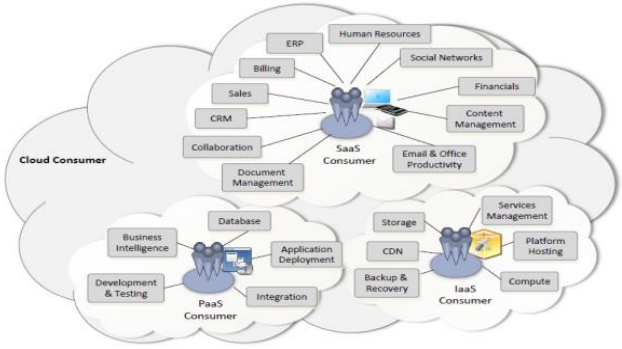

Fig 1.1 Service available for a cloud consumer

Businesses have the flexibility to subscribe to one of the following different type of cloud depending on their needs; public cloud, private cloud, community cloud and hybrid cloud. From this cloud computing types numerous services could be rendered by service providers to consumers. These include software as a service (SaaS), Platform as a service (PaaS) and infrastructure as a service (IaaS).

b, Cloud Computing Adoption Trend

Cloud Computing is a new computing paradigm which has gained a lot of attention over the last few decades. The technology analysts at Gartner see cloud computing as a so-called "emerging technology" which offers IT resources and services of the internet. The driving force has been its effective use of computing resources and management of information. Cloud Computing services allow the flexibility of IT operation outsource, easily adapting to growth or contraction, reduction in IT infrastructure cost and respond quickly to new market conditions. It indicates that a cloud service is implemented by some sort of pool of servers that either share a database subsystem or replicate data. Iyengar describes cloud computing as the mobile and remote data center that users share with your neighbors and strangers alike.

c, Data Protection in the Cloud

Information and data have become one of organization's greatest assets. The amount of data created and held by business is increasing on daily basis and keeping it safe continues to be a major cause for concern. Using cloud services present different security challenge to an organization than traditional IT solutions due to cloud service models and operational models employed and the technology used to enabling them. In an invent that the process or function fail to provide expected results cloud users are left at the mercy of cloud service providers. Information asset thus become widely public and distributed and control is somehow handed over to a third party. The onus lies on the cloud service providers and users to ensure the security of data in the cloud. Some organizations are reluctant to adopt cloud services although it is the best solution they may have to their problems due to cloud security issues.

d, Data Security and Privacy in the cloud

Cloud computing opens up the world of computing to a broader range of uses and increases the ease of use by giving access through any internet connection. However, with this increased ease also come drawbacks. Once data has gone into a public cloud, data security and governance control is transferred in whole or part to the cloud provider. Yet cloud providers are not assuming full responsibility of security issues in the cloud. This makes it critical for users to understand the security measures that cloud providers have in place, and it is equally important to take personal precautions to secure organizational data. The chief concern in cloud environments is to provide security around multi-tenancy and isolation, giving customers more comfort besides "trust us" idea of

clouds. There has been survey works reported that classifies security threats in cloud based on the nature of the service delivery models of a cloud computing system.

1.2 DATA SHARING

Data sharing is the important service in the cloud. In data sharing service user share the data with group of user. User does not have physical control when the data is in cloud. Any mistake can cause loss of data. To check integrity of data some scheme is used, when user cheat or leaves the group, the user should be revoked from group. Therefore user revocation is important in cloud storage. The cloud data owner uses his private key to generate signature for file blocks. When user is removed the user private key should also be removed .In previous scheme all signatures generated should get transfer to non-revoked user. In such case the nonrevoked user download all revoked user block resign and upload new one. This cause lots of computation of resources. Once user is removed from group, there is lots of burden of user revocation for large cloud. The situation will be more difficult when membership changes frequently.

The data sharing is one of the most widely used services that the cloud storage provides. With data sharing service, users can share their data in the cloud with a group of users, and reduce the burden of local data storage. Users, however, will lose the physical control over their data when they share them in the cloud. .Any error (the carelessness of human or the failure of hardware/software) might cause loss or damage to the data . In order to check the data integrity, some cloud storage auditing schemes for shared data are proposed .When a group user misbehaves or leaves the group, the user should be revoked from the group. Therefore, user revocation is a common realistic necessity in cloud storage auditing for shared data. In cloud storage auditing schemes, the data owner needs to use his/her private key to generate authenticators (signatures) for file blocks. These authenticators are used to prove that the cloud truly possesses these file blocks.

When a user is revoked, the user's private key should also be revoked. For traditional cloud storage auditing schemes for share data all of authenticators generated by the revoked user should be transformed into the authenticators of one designated non -revoked group user. In this case, this non-revoked group user needs to download all of revoked user's blocks, re-sign these blocks, and upload new authenticators to the cloud.

Obviously, it costs huge amount of computation resource and communication resource due to the large size of shared data in the cloud. In order to solve this problem, recently, some auditing schemes for shared data with user revocation have been proposed. When a user is revoked, the cloud will transform the authenticators of the revoked user's blocks into the authenticators of one non-revoked group user corresponding to these blocks, with a re-signing key.

The goal of cloud storage is to provide powerful out-sourcing data services on demand to the users exploiting highly virtualized infrastructure. There is a triad for the security of the outsourced data in cloud i.e. Confidentiality, Integrity and Availability (CIA). The data stored in the cloud may get corrupted by many reasons. Conventional method RSA needs to download the whole data from the cloud to validate the integrity of data for comparing the hash values of the same. Conventional methods are not suitable for healthcare data, financial data as these methods having a huge amount of data stored in cloud. General schematics comprises of data owner (DO), third party auditor (TPA), cloud storage service (CSS), access control service (ACS) and users showing integrity auditing scheme in cloud environment is given in Fig 1.1.

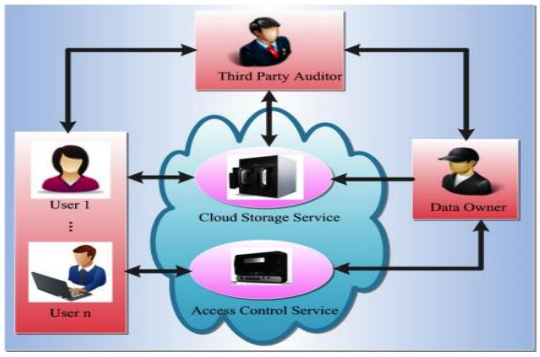

Fig 1.2 General schematics of integrity auditing

DO is responsible for data encryption and metadata generation for all stored files on the cloud and stores it in CSS. On requisition of users to check the integrity of their data stored on cloud, DO send a request to the TPA for the same. The TPA generates a challenge on receiving the request from DO, and sends it to the CSS. Then, CSS sends a response to TPA against that challenge; TPA verifies the proof for the correctness of the data stored on cloud and sends the auditing result to the DO. In the existing public auditing plans, there are two data models such as static and dynamic. In the static model, all the updated data remain untouched during the storage phase (eg. medical data). Whereas in dynamic model, all the updates may be done like modification, and insertion (eg. social media).

That scheme reduced the computation overhead at user side by using multi-threading model. Most of the existing integrity auditing schemes are using bilinear pairing for privacy protection of users, which leads to the delay in verification; hence the computation cost is high. To reduce the overhead at data owner side many researchers proposed different strategies. These strategies ensure the security and privacy of the cloud data but do not support dynamic data operations such as insertion, deletion, append and update in single and multi-user cloud environment. These strategies are more productive only when the respectability of the data is checked by the public verifier. Therefore, in cloud storage dynamic data operations are very frequently used (e.g. twitter). to reduce the computational cost.

1.3 OVERVIEW

The data sharing is one of the most widely used services that the cloud storage provides. With data sharing service, users can share their data in the cloud with a group of users, and reduce the burden of local data storage. Users, however, will lose the physical control over their data when they share them in the cloud. Any error (the carelessness of human or the failure of hardware/software) might cause loss or damage to the data. In order to check the data integrity, some cloud storage auditing schemes for shared data are proposed. When a group user misbehaves or leaves the group, the user should be revoked from the group.

Therefore, user revocation is a common realistic necessity in cloud storage auditing for shared data. In cloud storage auditing schemes, the data owner needs to use his/her private key to generate authenticators (signatures) for file blocks. These authenticators are used to prove that the cloud truly possesses these file blocks. When a user is revoked, the user's private key should also be revoked. For traditional cloud storage auditing schemes for share data, all of authenticators generated by the revoked user should be transformed into the authenticators of one designated non revoked group user. In this case, this non-revoked group user needs to download all of revoked user's blocks, re-sign these blocks, and upload new authenticators to the cloud. Obviously, it costs huge amount of computation resource and communication resource due to the large size of shared data in the cloud. In order to solve this problem, recently, some auditing schemes for shared data with user revocation have been proposed.

When a user is revoked, the cloud will transform the authenticators of the revoked user's blocks into the authenticators of one non-revoked group user corresponding to these blocks, with a re-signing key. The computation overhead of user revocation is still linear with the total number of file blocks stored by the revoked user in the cloud. Although this method relieves the burden on the non-revoked group user, it transfers the burden to the cloud.

According to a research by Nasuni, there was over 1 exabyte of data stored in the cloud. In reality, people might share extensive amount of file blocks with others on the cloud. Once a user is revoked from the group, the burden of user revocation might be huge, even for the computationally powerful cloud. The matter will be even worse when the membership of the group frequently alters. Therefore, how to design a cloud storage auditing scheme for shared data supporting real efficient user revocation is very worthwhile.

The group's private key derives from two components. One component remains fixed since being issued, and the other component alters with user revocation. Also propose a novel private key update technique to support user revocation. When users are revoked from the group, all of the non-revoked users can update their private keys by this technique to make the cloud storage auditing still work, while the identity information of the group does not need to change. In addition, the revoked users are not able to upload data and authenticators to the cloud any more. In this way, all of the authenticators generated before user revocation do not need to be recomputed. Therefore, the overhead of user revocation is fully independent of the total number of the revoked user's blocks. Even when the amount of data is immense, the group can still complete user revocation very efficiently. Besides, our scheme is based on identity-based cryptography, which eliminates the complicated certificate management in traditional PKI systems, including certificate generation, certificate revocation, certificate renewal, etc.

It proves the correctness and the security of the proposed scheme by concrete analysis. Also justify the performance of the proposed scheme by concrete implementation. In our experiments, evaluate the performance in different phases, and compare our scheme with others in terms of the computation overhead of user revocation. The experiments result shows that our scheme is efficient.

To ensure efficient user revocation in identity-based cloud storage auditing for shared data, our designed scheme should meet the following objectives: Correctness: to ensure that the proof from the cloud can pass the TPA's validation, if the cloud, group users, the group manager and the TPA are honest and obey the specified procedures. Soundness: to ensure that the cloud cannot pass the TPA's verification if it does not store group users' intact data. Secure user revocation: to ensure that the revoked users cannot upload data and the corresponding authenticators to the cloud any more. Efficient user revocation: to ensure that the computation overhead of user revocation is completely independent of the total number of revoked user's blocks. (5) Public auditing: to ensure that the TPA can verify the integrity of shared cloud data on behalf of group users.

2. LITERATURE SURVEY

Cloud computing provides a scalable environment for growing amounts of data and processes that work on various applications and services by means of on-demand self-services. Especially, the outsourced storage in clouds has become a new profit growth point by providing a comparably low-cost, scalable, location-independent platform for managing clients' data. The cloud storage service (CSS) relieves the burden for storage management and maintenance. In this project propose a dynamic audit service for verifying the integrity of an untrusted and outsourced storage. Our audit service is constructed based on the techniques, fragment structure, random sampling, and index-hash table, supporting provable updates to outsourced data and timely anomaly detection. In addition, propose a method based on probabilistic query and periodic verification for improving the performance of audit services. Our experimental results not only validate the effectiveness of our approaches, but also show our audit system verifies the integrity with lower computation overhead and requiring less extra storage for audit metadata[1].

Cloud computing has emerged as a new computing paradigm that offers great potential for storing data remotely. Presently, many organizations have reduced the burden of local data storage and maintenance by outsourcing data storage to the cloud. However, integrity and security of the outsourced data continues to be a matter of major concern for data owners due to the lack of control and physical possession over the data. To deal with this problem, researchers have proposed remote data auditing (RDA) techniques. However, the majority of existing RDA techniques is only applicable for static archived data and is not applicable for auditing or dynamically updating the outsourced data. They are also not applicable to big data storage because of the high computational overhead on the auditor. In this work propose an efficient RDA technique based on algebraic signature properties for a cloud storage system that incurs minimum computational and communication costs[2]. Also present the design of a new data structure-Divide and Conquer Table (DCT)—that can efficiently support dynamic data operations such as append, insert, modify, and delete. Our proposed data structure can be applied for large-scale data storage and will incur minimum computational cost. A comparison between our proposed method and other state-of-the-art RDA techniques shows that our method is secure and highly efficient in reducing the computational and communication costs on the server and the auditor [3].

With the cloud storage services, users can easily form a group and share data with each other. Given the fact that the cloud is not trustable, users need to compute signatures for blocks of the shared data to allow public integrity auditing. Once a user is revoked from the group, the blocks that were previously signed by this revoked user must be re-signed by an existing user, which may result in heavy communication and computation cost for the user. Proxy re-signatures can be used here to allow the cloud to do the re-signing work on behalf of the group. However, a malicious cloud is able to use the re-signing keys to arbitrarily convert signatures from one user to another deliberately. Moreover, collusions between revoked users and a malicious cloud will disclose the secret values of the existing users. In this work propose a novel public auditing scheme for the integrity of shared data with efficient and collusion-resistant user revocation utilizing the concept of Shamir secret sharing. Besides, our scheme also supports secure and efficient public auditing due to our improved polynomial-based authentication tags. The numerical analysis and experimental results demonstrate that our proposed scheme is provably secure and highly efficient [4].

With cloud data services, it is commonplace for data to be not only stored in the cloud, but also shared across multiple users. Unfortunately, the integrity of cloud data is subject to skepticism due to the existence of hardware/software failures and human errors. Several mechanisms have been designed to allow both data owners and public verifiers to efficiently audit cloud data integrity without retrieving the entire data from the cloud server. However, public auditing on the integrity of shared data with these existing mechanisms will inevitably reveal confidential information—identity privacy to public verifiers. In this project propose a novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud. In particular, exploit ring signatures to compute verification metadata needed to audit the correctness of shared data. With our mechanism, the identity of the signer on each block in shared data is kept private from public verifiers, who are able to efficiently verify shared data integrity without retrieving the entire file. In addition, our mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one. Our experimental results demonstrate the effectiveness and efficiency of our mechanism when auditing shared data integrity [5].

In past years, the rapid development of cloud storage services makes it easier than ever for cloud users to share data with each other. To ensure users' confidence of the integrity of their shared data on cloud, a number of techniques have been proposed for data integrity auditing with focuses on various practical features, e.g., the support of dynamic data, public integrity auditing, low communication/computational

audit cost, low storage overhead. However, most of these techniques consider that only the original data owner can modify the shared data, which limits these techniques to client read-only applications. Recently, a few attempts started considering more realistic scenarios by allowing multiple cloud users to modify data with integrity assurance. Nevertheless, these attempts are still far from practical due to the tremendous computational cost on cloud users, especially when high error detection probability is required by the system. In this project propose a novel integrity auditing scheme for cloud data sharing services characterized by multi-user modification, public auditing, high error detection probability, efficient user revocation as well as practical computational/communication auditing performance. Our scheme can resist user impersonation attack, which is not considered in existing techniques that support multi-user modification. Batch auditing of multiple tasks is also efficiently supported in our scheme. Extensive experiments on Amazon EC2 cloud and different client devices (contemporary and mobile devices) show that our design allows the client to audit the integrity of a shared file with a constant computational cost of 340ms on PC (4.6s on mobile device) and a bounded communication cost of 77KB for 99% error detection probability with data corruption rate of 1% [6].

Identity-based encryption (IBE) is a public key cryptosystem and eliminates the demands of public key infrastructure (PKI) and certificate administration in conventional public key settings. Due to the absence of PKI, the revocation problem is a critical issue in IBE settings. Several revocable IBE schemes have been proposed regarding this issue. Quite recently, by embedding an outsourcing computation technique into IBE, Li et al. proposed a revocable IBE scheme with a key-update cloud service provider (KU-CSP). However, their scheme has two shortcomings. One is that the computation and communication costs are higher than previous revocable IBE schemes. The other shortcoming is lack of scalability in the sense that the KU-CSP must keep a secret value for each user. In the article, propose a new revocable IBE scheme with a cloud revocation authority (CRA) to solve the two shortcomings, namely, the performance is significantly improved and the CRA holds only a system secret for all the users. For security analysis, demonstrate that the proposed scheme is semantically secure under the decisional bilinear Diffie-Hellman (DBDH) assumption [7].

The advent of the cloud computing makes storage outsourcing become a rising trend, which promotes the secure remote data auditing a hot topic that appeared in the research literature. Recently some research consider the problem of secure and efficient public data integrity auditing for shared dynamic data. However, these schemes are still not secure against the collusion of cloud storage server and revoked group users during user revocation in practical cloud storage system. In this paper, we figure out the collusion attack in the exiting scheme and provide an efficient public integrity auditing scheme with secure group user revocation based on vector commitment and verifier-local revocation group signature. We design a concrete scheme based on the our scheme definition. Our scheme supports the public checking and efficient user revocation and also some nice properties, such as confidently, efficiency, countability and traceability of secure group user revocation [8].

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In this project propose Panda, a novel public auditing mechanism for the integrity of shared data with efficient user revocation in the cloud. In our mechanism, by utilizing the idea of proxy re-signatures, once a user in the group is revoked, the cloud is able to resign the blocks, which were signed by the revoked user, with a re-signing key.

As a result, the efficiency of user revocation can be significantly improved, and computation and communication resources of existing users can be easily saved. Meanwhile, the cloud, which is not in the same trusted domain with each user, is only able to

convert a signature of the revoked user into a signature of an existing user on the same block, but it cannot sign arbitrary blocks on behalf of either the revoked user or an existing user.

By designing a new proxy re-signature scheme with nice properties, which traditional proxy resignatures do not have, our mechanism is always able to check the integrity of shared data without retrieving the entire data from the cloud. Moreover, our proposed mechanism is scalable, which indicates it is not only able to efficiently support a large number of users to share data and but also able to handle multiple auditing tasks simultaneously with batch auditing. In addition, by taking advantages of Shamir Secret Sharing, can also extend our mechanism into the multi-proxy model to minimize the chance of the misuse on re-signing keys in the cloud and improve the reliability of the entire mechanism.

Disadvantages of Existing System

- Especially when the number of re-signed blocks is quite large.
- Existing users may access their data sharing services provided by the cloud with resource limited devices, such as mobile phones.
- Frequent Security Issues

3.1 PROPOSED SYSTEM

In this design, the group's public key is replaced by the group's identity information, which remains unchanged in the whole lifetime. The group's private key derives from two components. One component remains fixed since being issued, and the other component alters with user revocation. Also propose a novel private key update technique to support user revocation.

When users are revoked from the group, all of the non-revoked users can update their private keys by this technique to make the cloud storage auditing still work, while the identity information of the group does not need to change. In addition, the revoked users are not able to upload data and authenticators to the cloud any more. In this way, all of the authenticators generated before user revocation do not need to be recomputed. Therefore, the overhead of user revocation is fully independent of the total number of the revoked user's blocks.

Advantages of Proposed System

- High efficiency on both cloud side and group user side
- Eliminates the complicated certificate management in traditional Public Key Infrastructure (PKI) systems

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

In existing approaches, when group users are revoked, the authenticators of revoked users' blocks will be transformed into those of some designated non-revoked group user to make the cloud storage auditing still work. It will incur huge computation overhead because the number of revoked users' blocks is usually enormous in big data storage scenario. Our basic idea of solving this problem is to update the non-revoked group users' private keys rather than update authenticators for realizing user revocation. One challenge face is how to achieve the integrity checking of the revoked user's data under the condition that the revoked user's authenticators are not updated. In addition, need to be able to detect and refuse the uploading request from the revoked user once he/she is revoked. Propose a secure multi-owner data sharing scheme. It implies that any user in the group can securely share data with others by the untrusted cloud. Our proposed scheme is able to support dynamic groups efficiently. Specifically, new granted users can directly decrypt data files uploaded before their participation without contacting with data owners.

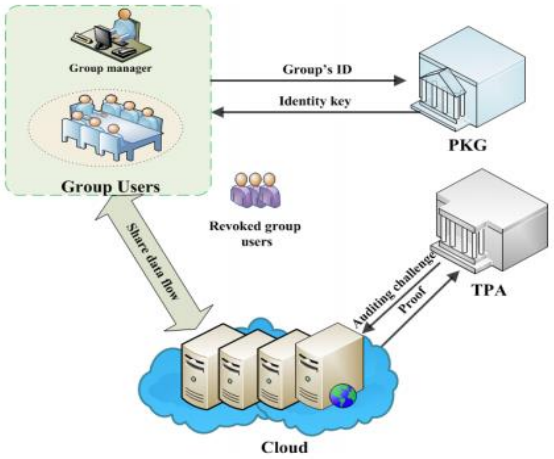

Fig 4.1 System Architecture

User revocation can be easily achieved through a novel revocation list without updating the secret keys of the remaining users. The size and computation overhead of encryption are constant and independent with the number of revoked users. It provides secure and privacy-preserving access control to users, which guarantees any member in a group to anonymously utilize the cloud resource. Moreover, the real identities of data owners can be revealed by the group manager when disputes occur. provide rigorous security analysis, and perform extensive Remote Access to demonstrate the efficiency of our scheme in terms of storage and computation overhead.

In our design, all group users have the same public key and the same private key. The public key is the group's ID, which remains fixed during the entire lifetime. The private key derives from two components, namely, an identity key IDK_{ID} and a partial key $TK_{ID,RN}$. The identity key IDK_{ID} is generated by the PKG, which is related to the group's ID and remains fixed since being issued. The partial key $TK_{ID,RN}$ is generated by the group manager, which is related to the group's ID and the number of user revocations RN, and alters along with user revocation. Group users compute their new private keys SKID,RN by using the identity key IDKID and the partial key TKID,RN . The user revocation is realized by a key update technique. The number of user revocations RN plays an important role in the key update. RN is a value set by the group manager, and also known by group users and the cloud.

4.2 DATAFLOW DIAGRAM

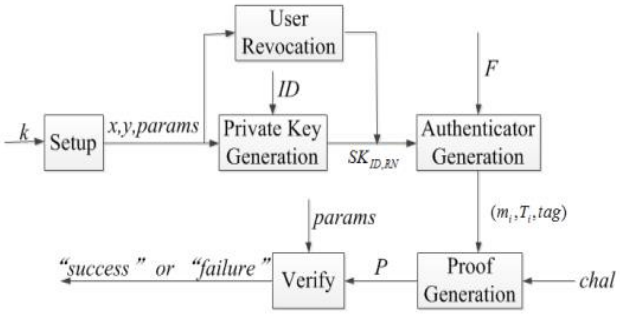

Fig 4.2 Identity-based cloud storage auditing scheme with efficient user revocation

In our design, the number of user revocations RN is integrated into authenticators and the file. When a group user would like to upload data to the cloud, he/she computes authenticators for file blocks according to the current private key and the newest RN, and then uploads them to the cloud. The cloud firstly verifies the validity of the file tag and the authenticators related to the current private key and the newest RN. Because the revoked user cannot use a previous private key to generate the valid authenticators under the newest RN, the data and the authenticators from the revoked user will be refused by the cloud. When the integrity auditing is performed, the TPA needs to retrieve the value of RN which has been integrated into the file tag. The TPA verifies the cloud data integrity using this RN and the group identity. In this way, the integrity auditing of the revoked user's data can still be performed even if the revoked user's authenticators are not updated.

5. SYSTEM IMPLEMENTATION

Systems implementation is the process of defining how the information system should be built, ensuring that the information system is operational and used, ensuring that the information system meets quality standard (i.e., quality assurance).

5.1 SYSTEM MODULES

- Group user
- Group manager
- Cloud
- PKG
- TPA

5.2 MODULE DESCRIPTION

5.2.1 Group User

A group is a collection of entities, where each entity can be either another group or a user. There are multiple group users in a group. Each group user can share data with others through the cloud storage. Group users can join or leave the group. The legal group users are honest and will not leak any private information to others.

The authenticator generation algorithm is run by the group user. It takes as input a file F = (m_1, ..., m_n), the number of user revocation RN and the user's private key $SK_{ID,RN}$, and generates a file tag *tag* and the authenticators T_i(i = 1, ..., n) for blocks m_i . The cloud verifies the correctness of the file tag *tag* and the authenticators.

5.2.3 Group manager

The group manager is a powerful entity. It can be viewed as an administrator of the group. When a user leaves the group, the manager is in charge of revoking this user. The revoked user cannot upload data to the cloud any more.

5.2.4 Cloud

The cloud provides enormous storage space and computing resources for group users. Through the cloud storage, group users can enjoy the data sharing service. The proof generation algorithm is run by the cloud. It takes as input a file F, a set of the corresponding authenticators and an auditing challenge *chal*, and generates a proof P which is used to prove the cloud accurately stores F.

5.2.5 PKG

The PKG is trusted by other entities. It is in charge of generating system public parameters and the identity key of the group according to the group's identity (ID)

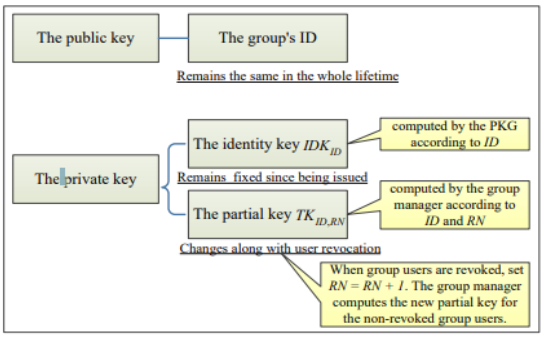

Fig 5.1 The strategy for key generation

The private key generation algorithm is run by the PKG, the group manager and group users. Based on the group's ID, the PKG computes the identity key IDK_{ID} and sends it to group users. Set the number of user revocation RN = 0. The group manager computes the partial key $TK_{ID,RN}$, and sends it to group users. According to the identity key IDKID and the partial key $TK_{ID,RN}$, each group user computes his/her private key SKID,RN. The setup algorithm is run by the PKG. It takes as input a security parameter k. It outputs the master secret key x, the master partial key y and the system public parameters params. The PKG holds the master secret key x itself, and sends the master partial key y to the group manager.

5.2.6 TPA

The TPA is responsible for auditing the integrity of cloud data on behalf of group users. When the TPA wants to audit the data integrity, it will send an auditing challenge to the cloud. After receiving the auditing challenge, the cloud will respond to the TPA with a proof of data possession. Finally, the TPA will verify the data integrity by checking the correctness of the proof. The TPA is a powerful party and it is honest. The proof verification algorithm is run by the TPA. It

takes as input a proof P, and system public parameters, and returns “success” if the proof is valid; or “failure”, otherwise.

5.3 ALGORITHM

5.3.1 AES Encryption

The encryption process uses a set of specially derived keys called round keys. These are applied, along with other operations, on an array of data that holds exactly one block of data the data to be encrypted. This array call the state array.

- Take the following AES steps of encryption for a 128-bit block:
- Derive the set of round keys from the cipher key.
- Initialize the state array with the block data (plaintext).
- Add the initial round key to the starting state array.
- Perform nine rounds of state manipulation.
- Perform the tenth and final round of state manipulation.
- Copy the final state array out as the encrypted data (ciphertext).

Each round of the encryption process requires a series of steps to alter the state array. These steps involve four types of operations called:

- SubBytes
- ShiftRows
- MixColumns
- XorRoundKey

The details of these operations are described shortly, but first need to look in more detail at the generation of the Round Keys, so called because there is a different one for each round in the process.

Round Keys

The cipher key used for encryption is 128 bits long. Where this key comes from is not important here; key hierarchy and how the temporal encryption keys are produced. The cipher key is already the result of many hashing and cryptographic transformations and, by the time it arrives at the AES block encryption, it is far removed from the secret master key held by the authentication server. Now, finally, it is used to generate a set of eleven 128-bit round keys that will be combined with the data during encryption. Although there are ten rounds, eleven keys are needed because one extra key is added to the initial state array before the rounds start. The best way to view these keys is an array of eleven 16-byte values, each made up of four 32-bit words. To start with, the first round key Rkey0 is simply initialized to the value of the cipher key (that is the secret key delivered through the key hierarchy). Each of the remaining ten keys is derived from this as follows. There is a good reason why the sequence of this table suddenly breaks off from 128 to 27. It is because of the way finite fields overflow.

Although the algorithm for deriving the round keys seems rather complicated, you will notice that no difficult computations have been performed and it is not at all computationally intensive. Also note that, after the first, each key is generated sequentially and based on the previous one.

Computing the Rounds

Having described how the round keys are derived, can now return to the operations used in computing each round. Earlier mentioned that four operations are required called:

- SubBytes
- ShiftRows
- MixColumns
- XorRoundKey

Each one of these operations is applied to the current state array and produces a new version of the state array. In all but the rarest cases, the state array is changed by the operation. The details of each operation are given shortly.

SubBytes

This operation is a simple substitution that converts every byte into a different value. AES defines a table of 256 values for the substitution. You work through the 16 bytes of the state array, use each byte as an index into the 256-byte substitution table, and replace the byte with the value from the substitution table. Because all possible 256 byte values are present in the table, you end up with a totally new result in the state array, which can be restored to its original contents using an inverse substitution table. The contents of the substitution table are not arbitrary; the entries are computed using a mathematical formula but most implementations will simply have the substitution table stored in memory as part of the design.

MixColumns

This operation is the most difficult, both to explain and perform. Each column of the state array is processed separately to produce a new column. The new column replaces the old one. The processing involves a matrix multiplication. If you are not familiar with matrix arithmetic, don't get to concerned. it is really just a convenient notation for showing operations on tables and arrays.

5.3.2 AES Decryption

As you might expect, decryption involves reversing all the steps taken in encryption using inverse functions:

- InvSubBytes
- InvShiftRows
- InvMixColumns
-

XorRoundKey doesn't need an inverse function because XORing twice takes you back to the original value. InvSubBytes works the same way as SubBytes but uses a different table that returns the original value. InvShiftRows involves rotating left instead of right and InvMixColumns uses a different constant matrix to multiply the columns.

The order of operation in decryption is:

- XorRoundKey
- InvShiftRows
- InvSubBytes
- Perform nine full decryption rounds:
- XorRoundKey
- InvMixColumns
- InvShiftRows
- InvSubBytes
- Perform final XorRoundKey

The same round keys are used in the same order.

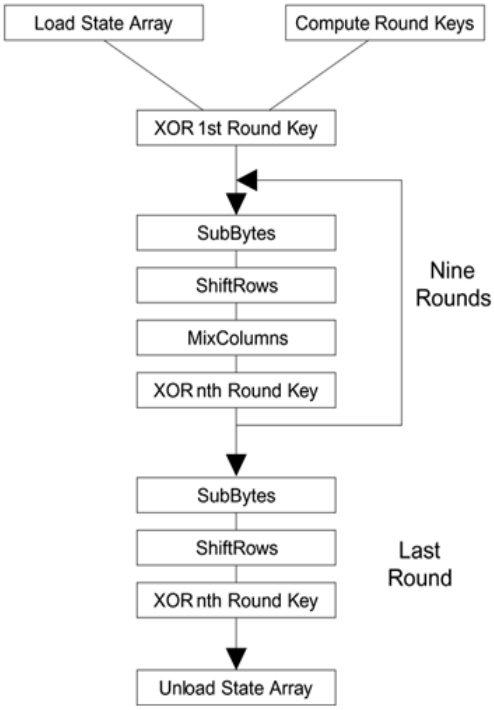

Fig 5.2 AES Encryption

6. PERFORMANCE ANALYSIS

Network security combines multiple layers of defenses at the edge and in the network. Each network security layer implements policies and controls.

The bit length for the existing system PANDA Plus is =82%. While comparing with our proposed system the Storage Auditing Scheme is 110%.

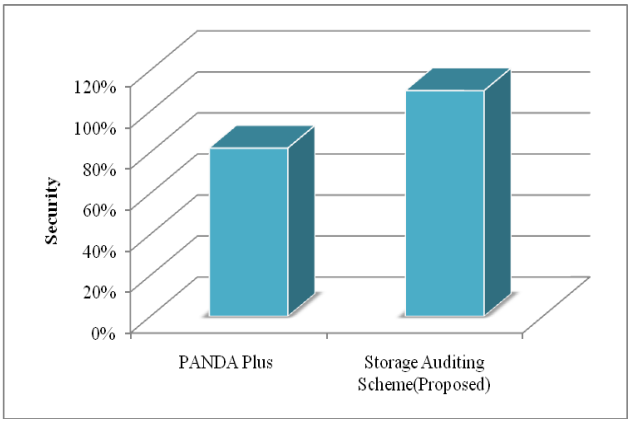

Fig 6.1 Security

Encryption user time is the consumption of encryption time required to perform during computational process.

The bit length for the existing system is PANDA Plus =6.4. While comparing with our proposed system the Storage Auditing Scheme is 5.6.

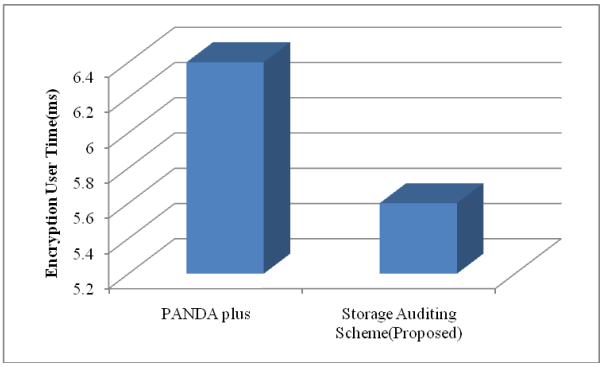

Fig 6.2 Encryption User Time

7. CONCLUSION

In this project propose an identity-based cloud storage auditing scheme for shared data, which supports real efficient user revocation. In our scheme, the cloud or the non-revoked user does not need to re-sign any file blocks of the revoked user. The overhead of user revocation in our scheme is fully independent of the number of the revoked user's blocks. Security proof and experimental results show that our proposed scheme is secure and efficient.

Extend this work with cluster Management with Forward Secrecy & Backward Secrecy by Time period & Recovery of File once knowledge Integrity Checking Fault Occur.

REFERENCES

[1] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.
[2] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," In Proc. of IEEE Cloud 2012, pp. 295-302, 2012.
[3] B. Wang, B. Li, and H. Li, "Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud," In Proc. of International Conference on Applied Cryptography and Network Security, pp. 507-525, 2012.
[4] G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu, and R. Hao. "Enabling Public Auditing for Shared Data in Cloud Storage Supporting Identity Privacy and Traceability," Journal of Systems and Software, vol. 113, pp. 130-139, 2016.
[5] W. Shen, J. Yu, H. Xia, H. Zhang, X. Lu and R. Hao, "Light-weight and Privacy-preserving Secure Cloud Auditing Scheme for Group Users via the Third Party Medium," Journal of Network and Computer Applications, vol. 82, pp.56-64, 2017.
[6] B. Wang, B. Li, and H. Li, "Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud," IEEE Transactions on Services Computing, vol. 8, no. 1, pp. 92-106, 2015.
[7] J. Yuan and S. Yu, "Public Integrity Auditing for Dynamic Data Sharing With Multiuser Modification," IEEE Transactions on Information Forensics and Security, vol. 10, no. 8, pp. 1717-1726, Aug. 2015.
[8] Y. Luo, M. Xu, S. Fu, D. Wang, and J. Deng, "Efficient Integrity Auditing for Shared Data in the Cloud with Secure User Revocation," IEEE Trustcom/BigDataSE/ISPA, pp. 434-442, 2015.

[9] Goran Candrli ˇ C, "How Much Is Stored in the Cloud?", ´ online at http://www.globaldots.com/how-much-is-storedin-the-cloud/.
[10] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," In Proc. of ACM CCS 2007, pp. 598- 610, 2007.
[11] A. Juels and B. S. Kaliski Jr, "Pors: Proofs of Retrievability for Large Files," In Proc. of 14th ACM conference on Computer and communications security, pp. 584-597, 2007.
[12] H. Shacham and B. Waters, "Compact Proofs of Retrievability," In Proc. of ASIACRYPT 2008, pp. 90-107, 2008.
[13] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," In Proc. of 4th international conference on Security and privacy in communication netowrks, pp. 1-10, 2008.
[14] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," IEEE Transactions on Parallel and Distributed Systems, vol.22,no.5, pp. 847-859, 2011.
[15] Y. Zhu, H.G. Ahn, H. Hu, S.S. Yau, H.J. An, and C.J. Hu, "Dynamic Audit Services for Outsourced Storages in Clouds," IEEE Transactions on Services Computing, vol. 6, no. 2, pp. 409-428, 2013.
[16] D. Cash, A. Kupc¸ ¨ u, and D. Wichs, "Dynamic Proofs of ¨ Retrievability via Oblivious Ram," In Proc. 32nd Intl Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT 13), pp. 279-295, 2013.
[17] K. Yang and X. Jia, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing," IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No. 9, pp. 1717-1726, 2013.
[18] M. Sookhak, A. Gania, M. K. Khanb, and R. Buyyac, "Dynamic Remote Data Auditing for Securing Big Data Storage in Cloud Computing," Information Science, vol. 380, pp. 101-116, 2017.
[19] L. Rao, H. Zhang, and T. Tu, "Dynamic Outsourced Auditing Services for Cloud Storage Based on Batch-LeavesAuthenticated Merkle Hash Tree," IEEE Transactions on Services Computing, Available online 26 May 2017 DOI: 10.1109/TSC.2017.2708116.
[20] C. Wang, S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Transactions on Computers, vol. 62, No. 2, pp. 362-375, 2013.
[21] J. Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling Cloud Storage Auditing with Key-Exposure Resistance," IEEE Transactions on Information Forensics and Security. vol. 10, no. 6, pp. 1167-1179, Jun. 2015.
[22] J. Yu, K. Ren, and C. Wang, "Enabling Cloud Storage Auditing with Verifiable Outsourcing of Key Updates," IEEE Transactions on Information Forensics and Security, vol. 11, no.5, pp. 1362-1375, 2016.
[23] J. Yu and H. Wang, "Strong Key-Exposure Resilient Auditing for Secure Cloud Storage," IEEE Transactions on Information Forensics and Security, vol. 12, no.8, pp. 1931-1940, 2017.
[24] J. Yu, H. Rong, H. Xia, H. Zhang, X. Cheng, and F. Kong, "Intrusion-resilient identity-based signatures: Concrete scheme in the standard model and generic construction," Information Sciences, vol. 442-443, pp. 158-172, 2018.
[25] J. Yu, R. Hao, H. Zhao, M. Shu, and J. Fan, "IRIBE: Intrusion-resilient identity-based encryption," Information Sciences, vol. 329, pp. 90-104, 2016.