ISSN No.: 2455-6130
Volume 1, Issue 3, 2016, pp. 19-26

website: www.ijoetr.com

International Journal of Emerging Trends in Research

Efficient Group User Revocation with Integrity Auditing for Cloud

Abirami R[1*], Eswari K[2], Divya S [3]

[1, 2, 3]*Department of Computer Science and Engineering, Dhanalakshmi College of Engineering, Chennai*

* Corresponding author. E-mail: abiramesh@ymail.com (Abirami R)

Abstract

The cloud computing makes storage out sourcing develops into a rising tendency, which endorses the protected remote information auditing a hot topic that appeared in the investigate literature. Recently some research considers the difficulty of sheltered and competent public data reliability auditing for shared active data. However, these schemes are still not protected against the conspiracy of cloud storage server and revoked group users during user revocation in practical cloud storage system. We understand the collusion attack in the exiting format and offer professional public integrity auditing scheme with secure group user revocation based on vector commitment and verifier-local revocation group signature. Our system supports the public checking and well-organized user revocation and also some nice properties, such as confidently, efficiency, countability and traceability of secure group user revocation. Finally, the security and experimental analysis show that compared with its relevant schemes our scheme is also secure and efficient.

Keywords: Public integrity auditing, dynamic data, vector commitment

1. Introduction

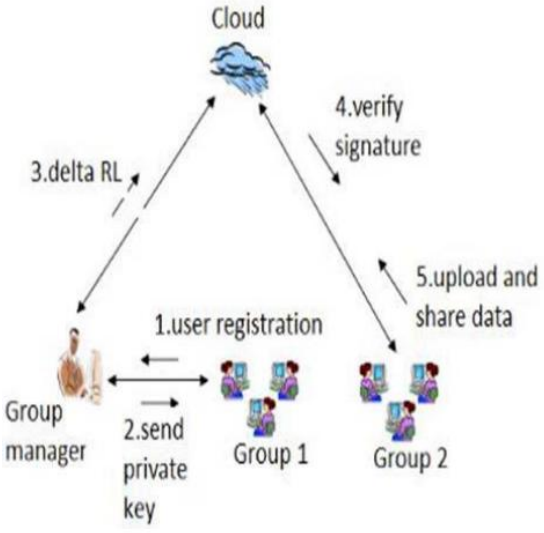
Cloud Storage service are such as easy storage services in online data endorsement services of amazon, and sensible cloud based software mozy, bitcasa, google drive, dropbox, and memopal have been built for cloud use. There is unacceptable result in cloud server like server hardware software breakdown, human maintenance and malicious assault. Rabin data dispersion system implemented for practical request and overcome above challenges. Author in [1-4] provide clarification to integrity and accessibility of remote cloud store. Dynamic scheme means when scheme support data alteration only data owner cloud modify data. The limited dynamic scheme cloud only efficiently supports special field operation. The static scheme not chains data alteration. In publicly verifiable, data integrity check can be performed by data owner and by any third party auditor. Multiple user in group need to share source code they require to access, modify compile and run the shared basis code at any time and place. Remote data auditing is merely data owner can update its data. Ring signature chains multiple user data process. The proxy re-signature is confidential and authenticated channels exist between each pair of entities.

Till today is no solution for above difficulty in public integrity auditing with cluster user modification.

2. Literature work

Group autograph with no revocation. The provably coalition-resistant scalable group name was described by some aouthors in 2000 [5-7]. At that time, the safety of group autograph was not totally unspoken and proper security dentitions were given later on by some reference [8-9] whose model captures all the requirements of group signatures in three properties. In (a relaxation of) this model, Boneh, Boyen and Shacham [10-11] find a construction in the accidental oracle model [12] with signatures shorter than 200 bytes [13]. In the BMW replica, the population of consumer is frozen after the setup phase past which no new member can be added. Dynamic group autograph were independently formal [14]. In these models, pairing-based schemes with relatively short signatures were put forth in [15]. Reference [6] also gave a construction without random oracles by interactive supposition. In the BMW replica [9], Boyen and Waters independently came up with a deferent standard model suggestion using more classical supposition and they subsequently reined their system [12] to obtain constant-size signatures. In the dynamic model [11], one of the references [8] described a system with constant size signatures without chance oracles but this scheme was rather a feasibility effect than an ancient construction. Later on, Groth gave [4] a fairly enceinte realization with signatures consisting of about 50 collection elements in the standard model with the strongest anonymity level. Revocation. In group name, membership revocation has received much notice in the last decade since revocation is central to digital signature schemes. One simple solution is to generate a new group public key and deliver a new signing key to each unrevoked member. However, in large groups, it may be inconvenient to change the public key and send a new secret to signers after they joined the group.

3. Architecture Diagram



Group of consumer are two types of users in a cluster: the original user and a number of cluster users. The original user at first creates shared data in the cloud, and shares it with collected works users. Both the original user and group users are associate of the group. Every member of the cluster is allowed to access and modify shared data. Common data and its verification metadata (i.e., signatures) are both amass in the cloud server. A public verifier, like a third party auditor

providing expert data auditing services or a data user outside the cluster intending to utilize shared data, is able to publicly confirm the integrity of shared data stored in cloud server.

Owner Registration:

A proprietor has to upload its files in a cloud server, he/she should register first. Then only he/she can be capable to do it. For that he wants to fill the facts in the registration form. These details are keep in a database.

Owner Login:

An owner have to login, they must login by filling their email id and password.

User Registration:

A user needs to access the data which is store up in a cloud; he/she must register their details first. These information are maintained in a Database.

User Login:

The user is an official user, he/she can download the folder by using file id which has been stored by data proprietor when it was uploading.

File Upload

File proprietor allowed uploading data on the cloud also for their private or public use. They act as a *Group Manager* for the file they update in the cloud. Both the unique user and group users are clever to access, download and adapt shared data. Shared data is alienated into a number of blocks. A user in the collection can modify a block in shared data by performing arts an insert, delete or update process on the block. File Auditing If and user edited a data then the auditor will watch the user and report to the owner about the reduced data. The group manager will check the changes in the file and if he founds any inconsistency auditor has full rights to relocate from his particular group. The public verifier can audit the honesty of shared data without retrieving the entire data from the cloud, still if some blocks in shared data have been re-signed by the cloud.

Re-assigning

On one hand, once a user is retract from the group, the blocks signed by the revoked user can be capably resigned. More specifically, the proxy is clever to convert a signature of Alice into a signature of Bob on the same block. Mean while, the deputy is not able to learn any confidential keys of the two users, which means it cannot sign any block on behalf of either Alice or Bob.

Group Sharing

Data owner will amass their data in the cloud and split the data among the group associate. Who upload the data have rights to adapt and download their data in the cloud. He can also set privileges to other users in his group to correct or download data.

Access control

Cloud Server permit only the authorized group member to amass their data in the cloud offered by cloud service supplier as Sass and it won’t allow unauthorized group associate to store their data in the cloud.

User Revocation

If a user wishes to revoke from a group their demand regarding revocation will be forwarded to the auditor where auditor will check to it and revoke the consumer from group. The user revocation is secure because only obtainable users are able to sign the blocks in shared data. even with a re-signing key, the cloud cannot make a valid signature for an arbitrary block on behalf of an existing user. In addition, after being cancel from the group, a revoke user is no longer in the user list, and can no longer generate valid signature on shared data.

4. System Model

A structure model for the cloud storage planning, which includes three major network entities: users, a cloud server, and a trusted third party.

• User: an individual or group entity, which possesses its data stored in the cloud for online information storage and computing. Different users may be affiliated with a widespread organization, and are assigned with sovereign authorities on certain data fields.

• Cloud server: an entity, which is managed by a particular cloud overhaul provider or cloud application operative to provide data storage and computing services. The cloud server is regarded as an entity with unrestricted storage and computational resources.

• Trusted third party: an optional and neutral entity, which has advanced capabilities on behalf of the users, to perform data public auditing and dispute arbitration. During cloud data accessing, the user separately interacts with the cloud server without peripheral interferences, and is assigned with the full and independent authority on its own data fields. It is necessary to assurance that the users' outsourced data cannot be unlawful accessed by other users.

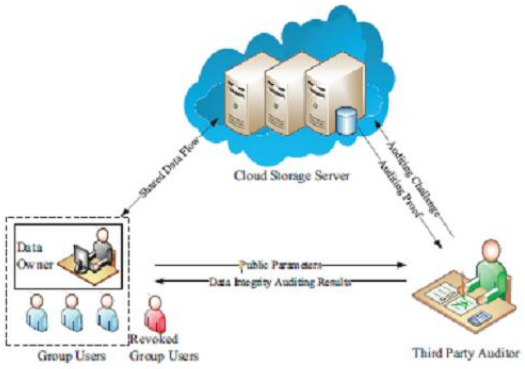


Figure1. The cloud storage model

- Admin login into system first
- Then create groups
- Number of users schedule himself
- Admin add them in dissimilar user then admin upload the file
- Select the group to which file will share then key for group user get generated

File uploaded then user login, using signature decrypted data then revoke user try to access file but they cannot for integrity admin send verification request to TPA then TPA verify data from Cloud Service Provider.

5. Preliminaries

In this section, a short introduction of some cryptographic techniques is given that are used in this paper, including bilinear maps, proxy resignatures, etc.

A. Bilinear Maps

Let G1 and G2 be two multiplicative cyclic groups of leading order p, g be a generator of G1. Bilinear map e is a map e: G1 × G1 → G2 with the following properties: 1) Computability: there exists an efficient algorithm for computing map e. 2) Bilinearity: for all u, v ∈ G1 and a, b ∈ Zp, e(ua, vb) = e(u, v)ab. 3) Nondegeneracy: e(g, g) 6= 1.

B. Complexity Assumptions

Definition 1: Computational Diffie-Hellman (CDH) Problem. For a, b ∈ Zp, given g, ga, gb ∈ G1 as input, output gab ∈ G1.The CDH assumption holds in G1 if it is computationally infeasible to solve the CDH quandary in G1. Definition 2: Discrete Logarithm (DL) Problem. For a ∈ Zp, given g, ga ∈ G1 as input, output a. The DL postulation holds in G1 if it is computationally infeasible to solve the DL problem in G1.

C. Homomorphic Authenticators

Homomorphic authenticators [2], also called homomorphic verifiable tags, allow a public verifier to check the integrity of data stored in the cloud without downloading the entire data. They have been widely used in the previous public auditing mechanisms [2]–[9]. Besides unforgeability, a homomorphic authenticable signature proposal, which denotes a homomorphic authenticator scheme based on signatures, should also satisfy the following properties:

Let (pk, sk) denote the signer's public/private key pair, σ1 denote the signature on block m1 ∈ Zp, and σ2 denote the signature on block m2 ∈ Zp.

• Blockless verifiability: Given σ1 and σ2, two random values α1, α2 in Zp and a block m′ = α1m1 + α2m2 ∈ Zp, a verifier is able to check the correctness of block m′ without knowing m1 and m2.
• Non-malleability: Given m1 and m2, σ1 and σ2, two random values α1, α2 in Zp and a block m′ = α1m1 + α2m2 ∈ Zp, a user, who does not have private key sk, is not able to generate a valid signature σ′ on block m′ by combining σ1 and σ2. Blockless verifiability enables a verifier to audit the correctness of data in the cloud with only a linear combination of all the blocks, while the entire data does not need to be downloaded to the verifier. Non-malleability indicates that an untrusted party cannot produce valid signatures on combined blocks by combining existing signatures.

D. Proxy Re-signatures, first proposed by Blaze et al. [11], allow a semi-trusted proxy to act as a translator of signatures between two users, for instance, Alice and Bob. More specifically, the proxy is capable to translate a signature of Alice into a signature of Bob on the same block. The proxy is not able to learn any private keys of the two users, which means it cannot sign any block on behalf of either Alice or Bob.

DESIGN GOAL

Our recommended scheme should attain the subsequent properties simultaneously:
1) Correctness: the verifier must accept all valid proof information generated by the cloud server;
2) Public Auditing: any entity with public explanations can audit the integrity of communal data without retrieving the data file back from the cloud; 3) Efficient User Revocation: once a user is revoked from the collection, the cloud should be able to help group users update blocks tags generated by the revoked user; 4) Scalability: the data integrity auditing cost on users should be independent or grow practically slow (e.g., logarithmic) to the data size and the number of data modifiers. 5) Security Goals: if the data are corrupted, the cloud servers are not able to produce valid integrity proof information.

6. Cipher Text Database

In cloud storage outsourcing environment, the outsourced figures is generally encrypted database, which is frequently implicitly assumed in the exiting academic research. Actually, our scheme could hold the auditing of database of both plaintext and cipher text database. However, it is not straight forward to widen a scheme to support encrypted database. In order to achieve the privacy of the data record mx, the client can use his/her secret key to encrypt each mx using a encryption scheme. When there is only one client (data owner) in the group, the user only needs to desire a random secret key and encrypt the data using a secure symmetric encryption scheme. Though, when the scheme needs to hold multi-user data modification, while at the same time keeping the shared data encrypted, a shared secret key along with group users will result in single point failure problem. It means that any group user (revoked or leave) leak the shared secret key will break the confidentiality guarantee of the data. To overcome the above problem, we need to adopt a scheme, which could support group users data modification. Luckily, Wu et al. [2] designed an Asymmetric Group Key Agreement scheme (ASGKA). The scheme has a nice property that, instead of a common secret key, only a shared encryption key is negotiated in an ASGKA protocol. Also, in the scheme, the public key can be simultaneously used to confirm signatures and encrypt messages while any signature can be used to decrypt ciphertext under this public key. Using the bilinear pairings, the authors instantiate a one-round ASGKA protocol tightly reduced to the decision Bilinear Diffie Hellman Exponentiation (BDHE) assumption in the standard model. Thus, according to the ASGKA protocol, we consider the case of encrypted database (x, cx), where x is an index and cx is the corresponding cipher value.We provide the detailed changes upon our scheme to support encrypted database.

1) In the Setup phase, the scheme has to run the key agreement of ASGKA for the group users. Then, the database DB = (i,mi) is encrypted by the group key gpk of data owner. Finally, the stored database is a ciphertext database DB = (i, ci).

2) In the second step of the Update phase, a group user firstly decrypts the record ci using the ASGKA secret key gsk[∗] to get plaintext database DB = (i,mi). Then, update the data to m′ i, and later encrypt the data with the public key gpk of ASGKA scheme to get the new encrypted database DB = (i, c′ i). 2) Public Auditing: Any entity with public keys can audit the integrity of shared data without retrieving the data file back from the cloud; 3) Efficient User Revocation: once a user is revoked from the group, the cloud should be able to help group users update blocks tags generated by the revoked user;

4) Scalability: the data integrity auditing cost on users should be independent or grow practically slow (e.g., logarithmic) to the data size and the number of data modifiers.

5) Security Goals: if the data are corrupted, the cloud servers are not able to produce valid integrity proof information; any illegitimate user shall not be able to impersonate valid users and generate legitimate tags behalf of valid users.

7. Proposed Systems

It is proposed a scheme to realize efficient and secure data integrity auditing for share dynamic data with multi-user modification. The scheme vector commitment , Asymmetric Group Key Agreement (AGKA) and group signatures with user revocation are adopt to get the data integrity auditing of remote data. Next to the public data auditing, the combining of the three primitive enable our scheme to outsource cipher text database to remote cloud and support secure group users revocation to public dynamic data. We provide security analysis of our scheme, and it shows that our scheme offer data confidentiality for group users and it is also secure against the collusion attack from the cloud storage server and revoked group users. Moreover, the performance analysis shows that, compared with its relevant schemes, our scheme is also efficient in different phases.

8. Conclusion

In this paper, securely share the data file between the dynamic groups. Without revealing their identity members in the identical group can share the data efficiently. Cryptography is used for over all safety. When compared to other algorithm key size is very small, it is not able to hack easily. It is used for efficient revocation without updating private keys of residual users. In future, concentrate on key organization, how to revoke the private keys from the group members.

References

[1] Amazon. (2007) Amazon simple storage service (amazon s3). Amazon. [Online]. Available: http://aws.amazon.com.

[2] Google. (2005) Google drive. Google. [Online]. Available: http://drive.google.com

[3] Dropbox. (2007) A file-storage and sharing service. Dropbox. [Online]. Available: http://www.dropbox.com/

[4] Mozy. (2007) An online, data, and computer backup software. EMC. [Online]. Available: http://www.dropbox.com/

[5] Bitcasa. (2011) Inifinite storage. Bitcasa. [Online]. Available: http://www.bitcasa.com/

[6] Memopal. (2007) Online backup. Memopal. [Online]. Available: http://www.memopal.com/

[7] M. A. et al., "Above the clouds: A berkeley view of cloud computing," Tech. Rep. UCBEECS, vol. 28, pp. 1–23, Feb. 2009.

[8] M. Rabin, "Efficient dispersal of information for security," Journal of the ACM (JACM), vol. 36(2), pp. 335–348, Apr. 1989.

[9] J. G. et al. (2006) The expanding digital universe: A forecast of worldwide information growth through 2010. IDC. [Online].Available: Whitepaper

[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, Virginia, USA, Oct. 2007, pp. 598–609.

[11] A. Juels and B. S. Kaliski, "Pors: Proofs of retrievability for large files," in Proc. of ACM CCS, Virginia, USA, Oct. 2007, pp. 584–597.

[12] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: theory and implementation," in Proc. of CCSW 2009, llinois, USA, Nov. 2009, pp. 43–54.

[13] Y. Dodis, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in Proc. of TCC 2009, CA, USA, Mar. 2009, pp. 109–127.

[14] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Proofs of retrievability via hardness amplification," in Proc. of ESORICS 2009, Saint-Malo, France, Sep. 2009, pp. 355–370.

[15] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. of ACM CCS, Illinois, USA, Nov. 2009, pp. 213–222.