International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 04 Issue 06
May 2017

Open Integrity Auditing for Shared Dynamic Cloud Data with Cluster User Revocation

P.Venkatesh ,
PG Scholar, Dept of CSE
Pydah College of Engineering and Technology
Visakhapatnam, AP, India .

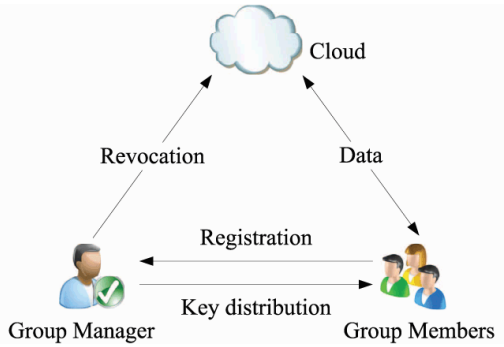
M. Venkatesh, M.Tech .
Assistant Professor , Dept of CSE
Pydah College of Engineering and Technology
Visakhapatnam, AP, India .

Abstract— Outsourcing storage via cloud computing has been the vogue since quite some time now, leading to researchers and analysts constantly work at dynamic remote data auditing for securing big data storage in a cloud computing environment. There have emerged certain strategies dealing with efficient public data integrity auditing for shared dynamic data. However, these strategies seem to gain little success over securing the remote data against the collusion of cloud storage server and revoked group users during user revocation in a practical cloud storage system. In this paper, we lay out a secure anti-collusion data sharing scheme for dynamic groups in the cloud by providing an efficient public integrity auditing scheme with secure group user revocation based on vector commitment and verifier-local revocation group signature. We have designed a concrete scheme that supports public checking and efficient user revocation and other key properties such as confidentiality, 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 more secure and efficient.

Keywords— Open integrity auditing, victor commitment, group signature, cluster user revocation, cloud computing

I. INTRODUCTION

Cloud Computing has become a scalable services consumption and delivery platform in the field of Services Computing. The technical foundations of Cloud Computing include Service-Oriented Architecture (SOA) and Virtualizations of hardware and software. The goal of Cloud Computing is to share resources among the cloud service consumers, cloud partners, and cloud vendors in the cloud value chain. The resource sharing at various levels results in various cloud offerings such as infrastructure cloud (e.g., hardware, IT infrastructure management), software cloud (e.g. SaaS focusing on middleware as a service, or traditional CRM as a service), application cloud (e.g., Application as a Service, UML modelling tools as a service, social network as a service), and business cloud (e.g., business process as a service).



The development of cloud computing motivates enterprises and organizations to outsource their data to third-party cloud service providers (CSPs), which will improve the storage limitation of resource constrain local devices. Recently, some commercial cloud storage services, such as the simple storage service (S3) [1] online data backup services of Amazon and some practical cloud based software Google Drive [2], Dropbox [3], Mozy [4], Bitcasa [5], and Memopal [6], have been built for cloud application. Since the cloud servers may return an invalid result in some cases, such as server hardware/software failure, human maintenance and malicious attack [7], new forms of assurance of data integrity and accessibility are required to protect the security and privacy of cloud user's data.

To overcome the above critical security challenge of today's cloud storage services, simple replication and protocols like Rabin's data dispersion scheme [8] are far from practical application. The formers are not practical because a recent IDC report suggests that data-generation is outpacing storage availability [9]. The later protocols ensure the availability of data when a quorum of repositories, such as k-out-of-n of shared data, is given. However, they do not provide assurances about the availability of each repository, which will limit the assurance that the protocols can provide to relying parties.

- We explore on the secure and efficient shared data integrate auditing for multi-user operation for cipher text database
- By incorporating the primitives of victor commitment, asymmetric group key agreement and group signature, we propose an efficient data auditing scheme while at the same time

providing some new features, such as traceability and countability.

- We provide the security and efficiency analysis of our scheme, and the analysis results show that our scheme is secure and efficient.

II. CLOUD STORAGE MODEL

Cloud storage model contain three entities, namely the cloud storage server, group users and a Third Part Auditor (TPA). Group users consist of a data owner and a number of users who are authorized to access and modify the data by the data owner. The cloud storage server is semi-trusted, who provides data storage services for the group users. TPA could be any entity in the cloud, which will be able to conduct the data integrity of the shared data stored in the cloud server. In our system, the data owner could encrypt and upload its data to the remote cloud storage server. Also, he/she shares the privilege such as access and modify (compile and execute if necessary) to a number of group users. The TPA could efficiently verify the integrity of the data stored in the cloud storage server; even the data is frequently updated by the group users. The data owner is different from the other group users; he/she could securely revoke a group user when a group user is found malicious or the contract of the user is expired.

III. THREAT MODEL AND SECURITY GOALS

Our threat model considers two types of attack:

1) An attacker outside the group (include the revoked group user cloud storage server) may obtain some knowledge of the plain-text of the data. Actually, this kind of attacker has to at least break the security of the adopted group data encryption scheme.

2) The cloud storage server colludes with the revoked group users, and they want to provide a illegal data without being detected.

Actually, in cloud environment, we assume that the cloud storage server is semi-trusted. Thus, it is reasonable that a revoked user will collude with the cloud server and share its secret group key to the cloud storage server. In this case, although the server proxy group user revocation way brings much communication and computation cost saving, it will make the scheme insecure against a malicious cloud storage server who can get the secret key of revoked users during the user revocation phase. Thus, a malicious cloud server will be able to make data m, last modified by a user that needed to be revoked, into a malicious data.

In the user revocation process, the cloud could make the malicious data m′ become valid. To overcome the problems above, we aim to achieve the following security goals in our paper:

- **Security**: A scheme is secure if for any database and any probabilistic polynomial time adversary, the adversary cannot convince a verifier to accept an invalid output.
- **Correctness**: A scheme is correct if for any database and for any updated data m by a valid group user, the output of the verification by an honest cloud storage server is always the value m. Here, m is a cipher-text if the scheme could efficiently support encrypted database.
- **Efficiency**: A scheme is efficient if for any data, the computation and storage overhead invested by any client user must be independent of the size of the shared data.
- **Countability**: A scheme is countable, if for any data the TPA can provide a proof for this misbehaviour, when the dishonest cloud storage server has tampered with the database.
- **Traceability**: We require that the data owner is able to trace the last user who updates the data (data item), when the data is generated by the generation algorithm and every signature generated by the user is valid.

IV. LITERATURE REVIEW

1) Efficient dispersal of information for security, load balancing, and fault tolerance

AUTHORS: M. Rabin

An Information Dispersal Algorithm (IDA) is developed that breaks a file F of length L = 1 F1 into n pieces Fi, l ≤ i ≤ n, each of length 1 Fi1 = L/m, so that every m pieces suffice for reconstructing F. Dispersal and reconstruction are computationally efficient. The sum of the lengths 1 Fi1 is (n/m) · L. Since n/m can be chosen to be close to l, the IDA is space efficient. IDA has numerous applications to secure and reliable storage of information in computer networks and even on single disks, to fault-tolerant and efficient transmission of information in networks, and to communications between processors in parallel computers. For the latter problem

provably time-efficient and highly fault-tolerant routing on the n-cube is achieved, using just constant size buffers.

2) Provable data possession at untrusted stores

AUTHORS: G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song

We introduce a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage system.

We present two provably-secure PDP schemes that are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. Experiments using our implementation verify the practicality of PDP and reveal that the performance of PDP is bounded by disk I/O and not by cryptographic computation.

3) Pors: Proofs of retrievability for large files

AUTHORS: A. Juels and B. S. Kaliski

We define and explore proofs of retrievability (PORs). A POR scheme enables an archive or back-up service (prover) to produce a concise proof that a user (verifier) can retrieve a target file F, that is, that the archive retains and reliably transmits file data sufficient for the user to recover F in its entirety.

A POR may be viewed as a kind of cryptographic proof of knowledge (POK), but one specially designed to handle a large file (or bitstring) F. We explore POR protocols here in which the communication costs, number of memory accesses for the prover, and storage requirements of the user (verifier) are small parameters essentially independent of the length of F. In addition to proposing new, practical POR constructions, we explore implementation considerations and optimizations that bear on previously explored, related schemes.

In a POR, unlike a POK, neither the prover nor the verifier need actually have knowledge of F. PORs give rise to a new and unusual security definition whose formulation is another contribution of our work.

We view PORs as an important tool for semi-trusted online archives. Existing cryptographic techniques help users ensure the privacy and integrity of files they retrieve. It is also natural, however, for users to want to verify that archives do not delete or modify files prior to retrieval. The goal of a POR is to accomplish these checks without users having to download the files themselves. A POR can also provide quality-of-service guarantees, i.e., show that a file is retrievable within a certain time bound.

4) Dynamic provable data possession

AUTHORS: C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia

We consider the problem of efficiently proving the integrity of data stored at untrusted servers. In the provable data possession (PDP) model, the client preprocesses the data and then sends it to an untrusted server for storage, while keeping a small amount of metadata. The client later asks the server to prove that the stored data has not been tampered with or deleted (without downloading the actual data). However, the original PDP scheme applies only to static (or append-only) files.

We present a definitional framework and efficient constructions for dynamic provable data possession (DPDP), which extends the PDP model to support provable updates to stored data. We use a new version of authenticated dictionaries based on rank information. The

price of dynamic updates is a performance change from O(1) to O(logn) (or O(nɛlog n), for a file consisting of n blocks, while maintaining the same (or better, respectively) probability of misbehavior detection. Our experiments show that this slowdown is very low in practice (e.g. 415KB proof size and 30ms computational overhead for a 1GB file). We also show how to apply our DPDP scheme to outsourced file systems and version control systems (e.g. CVS).

5) Privacy-preserving public auditing for data storage security in cloud computing

AUTHORS: C. Wang, Q. Wang, K. Ren, and W. Lou

Cloud computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. By data outsourcing, users can be relieved from the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the possibly large size of outsourced data makes the data integrity protection in Cloud Computing a very challenging and potentially formidable task, especially for users with constrained computing resources and capabilities. Thus, enabling public auditability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of outsourced data when needed. To securely introduce an effective third party auditor (TPA), the following two fundamental requirements have to be met: 1) TPA should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user; 2) The third party auditing process should bring in no new vulnerabilities towards user data privacy. In this paper, we utilize and uniquely combine the public key based homomorphic authenticator with random masking to achieve the privacy-preserving public cloud data auditing system, which meets all above requirements. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis shows the proposed schemes are provably secure and highly efficient.

V.SYSTEM ANALYSIS

EXISTING SYSTEM:

- For providing the integrity and availability of remote cloud store, some solutions and their variants have been proposed. In these solutions, when a scheme supports data modification, we call it dynamic scheme, otherwise static one (or limited dynamic scheme, if a scheme could only efficiently support some specified operation, such as append). A scheme is publicly verifiable means that the data integrity check can be performed not only by data owners, but also by any third-party auditor. However, the dynamic schemes above focus on the cases where there is a data owner and only the data owner could modify the data.
- To support multiple user data operation, Wang et al. proposed data integrity based on ring signature.
- To further enhance the previous scheme and support group user revocation, Wang et al. designed a scheme based on proxy re-signatures.
- Another attempt to improve the previous scheme and make the scheme efficient, scalable and collusion resistant is Yuan and Yu, who designed a dynamic public integrity auditing scheme with group user revocation. The authors designed polynomial authentication tags and adopt proxy tag update techniques in their scheme, which make their scheme support public checking and efficient user revocation.

DISADVANTAGES OF EXISTING SYSTEM:

- User revocation problem is not considered and the auditing cost is linear to the group size and data size.
- However, the scheme assumed that the private and authenticated channels exist between each pair of entities and there is no collusion among them. Also, the auditing cost of the scheme is linear to the group size.
- However, in Yuan and Yu scheme, the authors do not consider the data secrecy of group users. It means that, their scheme could efficiently support plaintext data update and integrity auditing, while not ciphertext data. In their scheme, if the data owner trivially shares a group key among the group users, the defection or revocation any group user will force the group users to update their shared key. Also, the data owner does not take part in the user revocation phase, where the cloud itself could conduct the user revocation phase. In this case, the collusion of revoked user and the cloud server will give chance to malicious cloud server where the cloud server could update the data as many time as designed and provide a legal data finally.

PROPOSED SYSTEM:

- The deficiency of above schemes motivates us to explore how to design an efficient and reliable scheme, while achieving secure group user revocation. To the end, we propose a construction which not only supports group data encryption and decryption during the data modification processing, but also realizes efficient and secure user revocation.
- Our idea is to apply vector commitment scheme over the database. Then we leverage the Asymmetric Group Key Agreement (AGKA) and group signatures to support ciphertext data base update among group users and efficient group user revocation respectively.
- Specifically, the group user uses the AGKA protocol to encrypt/decrypt the share database, which will guarantee that a user in the group will be able to encrypt/decrypt a message from any other group users. The group signature will prevent the collusion of cloud and revoked group users, where the data owner will take part in the user revocation phase and the cloud could not revoke the data that last modified by the revoked user.

ADVANTAGES OF PROPOSED SYSTEM:

- We explore on the secure and efficient shared data integrate auditing for multi-user operation for ciphertext database.
- By incorporating the primitives of victor commitment, asymmetric group key agreement and group signature, we propose an efficient data auditing scheme while at the same time providing some new features, such as traceability and countability.
- We provide the security and efficiency analysis of our scheme, and the analysis results show that our scheme is secure and efficient.

VI.IMPLEMENTATION

MODULES:

- Cloud server
- Group of users
- Public verifier
- Auditing Module

MODULES DESCRIPTION:

Cloud server

In the first module, we design our system with Cloud Server, where the datas are stored globally. Our mechanism, Oruta, should be designed to achieve following properties:

(1) Public Auditing: A public verifier is able to publicly verify the integrity of shared data without retrieving the entire data from the cloud.

(2) Correctness: A public verifier is able to correctly verify shared data integrity.

(3) Unforgeability: Only a user in the group can generate valid verification metadata (i.e., signatures) on shared data.

(4) Identity Privacy: A public verifier cannot distinguish the identity of the signer on each block in shared data during the process of auditing.

Group of users

- There are two types of users in a group: the original user and a number of group users. The original user initially creates shared data in the cloud, and shares it with group users. Both the original user and group users are members of the group. Every member of the group is allowed to access and modify shared data. Shared data and its verification metadata (i.e., signatures) are both stored in the cloud server. A public verifier, such as a third party auditor providing expert data auditing services or a data user outside the group intending to utilize shared data, is able to publicly verify the integrity of shared data stored in the cloud server.
- Owner Registration: In this module an owner has to upload its files in a cloud server, he/she should register first. Then only he/she can be able to do it. For that he needs to fill the details in the registration form. These details are maintained in a database.
- Owner Login: In this module, owners have to login, they should login by giving their email id and password.
- User Registration: In this module if a user wants to access the data which is stored in a cloud, he/she should register their details first. These details are maintained in a Database.
- User Login: If the user is an authorized user, he/she can download the file by using file id which has been stored by data owner when it was uploading.

Public verifier

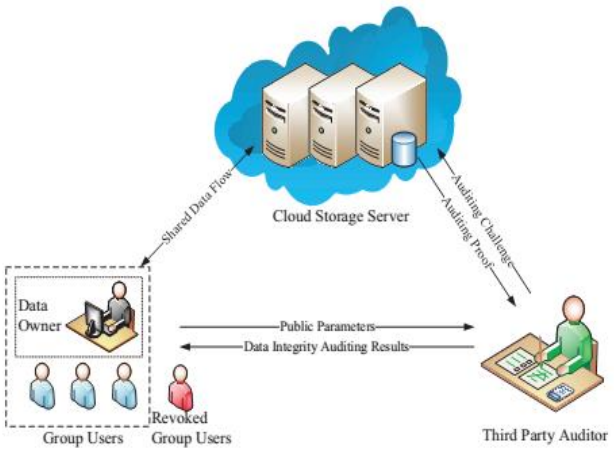
- When a public verifier wishes to check the integrity of shared data, it first sends an auditing challenge to the cloud server. After receiving the auditing challenge, the Cloud server responds to the public verifier with an auditing proof of the possession of shared data.
- Public verifier checks the correctness of the entire data by verifying the correctness of the auditing proof. Essentially, the process of public auditing is a challenge and- response protocol between a public verifier and the cloud server

Auditing Module

- Third party auditor TPA (maintainer of clouds) should register first. This system allows only cloud service providers. After third party auditor gets logged in, He/ She can see how many data owners have uploaded their files into the cloud. Here we are providing TPA for maintaining clouds.
- We only consider how to audit the integrity of shared data in the cloud with static groups. It means the group is pre-defined before shared data is created in the cloud and the membership of users in the group is not changed during data sharing.
- Original user is responsible for deciding who is able to share her data before outsourcing data to the cloud. Another interesting problem is how to audit the integrity of shared data in the cloud with dynamic groups — a new user can be added into the group and an existing group member can be revoked during data sharing — while still preserving identity privacy.

VII.SYSTEM DESIGN

SYSTEM ARCHITECTURE:



VIII.CONCLUSION

The primitive of verifiable database with efficient updates is an important way to solve the problem of verifiable outsourcing of storage. We propose 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 achieve the data integrity auditing of remote data. Beside the public data auditing, the combining of the three primitive enable our scheme to outsource ciphertext database to remote cloud and support secure group users revocation to shared dynamic data. We provide security analysis of our scheme, and it shows that our scheme provide data confidentiality for group users, and it is also secure against the collusion attack from the cloud storage server and revoked group users. Also, the performance analysis shows that, compared with its relevant schemes, our scheme is also efficient in different phases.

XI.REFERENCES

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

[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]http://www.ijracse.com/olvolume1issue10/JerripothuRavindraBabu-MangalagiriVenkatesh-DrRameshChallagundla-2.pdf

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

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

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

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

[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.

[16] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in

cloud computing," in *Proc. of IEEE INFOCOM 2010*, CA, USA, Mar. 2010, pp. 525–533.

[17] J. Yuan and S. Yu, "Proofs of retrievability with public verifiability and constant communication cost in cloud," in *Proc. of International Workshop on Security in Cloud Computing*, Hangzhou, China, May 2013, pp. 19–26.

[18] E. Shi, E. Stefanov, and C. Papamanthou, "Practical dynamic proofs of retrievability," in *Proc. of ACM CCS 2013*, Berlin, Germany, Nov. 2013, pp. 325–336.

[19] Cloud9. (2011) Your development environment, in the cloud. Cloud9. [Online]. Available: https://c9.io/

[20] Codeanywhere. (2011) Online code editor. Codeanywhere. [Online]. Available: https://codeanywhere.net/