International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 13
September 2016

Public and Private Integrity Audirting For Data Sharing With Multiple Users Modification

***G.SANDHYA** ****M.SHIRISHA**

*M.TECH student ,Dept of CSE, VAAGDEVI ENGINEERING COLLEGE WARANGAL

**Assistant Professor, Dept of CSE , VAAGDEVI ENGINEERING COLLEGE WARANGAL

Abstract

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 paper, we 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%.

INTRODUCTIONThe continuous of cloud techniques has boosted a number of public loud storage applications. In

particular, more and more cloud storage applications are being used as collaboration platforms, in which data are not only persisted in cloud for storage but also subject to frequent modifications from multiple users. Real-world examples are cloud-based storage synchronization platforms such as Dropbox for Business and Sugarsync , Version Control Systems (VCS) such as Subversion and Concurrent Versions System, which enable multiple team members to work in sync, accessing and modifying same files on cloud servers anywhere anytime. For correct execution of this kind of collaborative applications, one problem is to assure data integrity, i.e., each data modification operation is indeed performed by an authorized group member and the data remains intact and update to date thereafter.

This problem is important given the fact that cloud storage platforms, even well-known cloud platforms, may experience hardware/software failures, human er- *Copyright (c) 2013 IEEE. Personal use of this material is permitted.* we observed that there have been large discrepancies between the numbers of data corruption events reported by users and those acknowledged by service providers, which also causes users to doubt whether or not their data on cloud are truly intact. With the concern on data integrity of cloud storage services, users wish to have a way of auditing the cloud server to ensure that the server stores all their latest data without any corruption. To offer such a service, a series of schemes have been proposed. However, for most of these existing schemes, only the data owner who holds secret keys can modify the data and all other users who share data with the data owner only have read permission. If these solutions are trivially extended to support multiple writers with data integrity assurance, the data owner has to stay online, collecting modified data from other users and regenerating authentication tags for them. Obviously, this kind of trivial extension will introduce a tremendous workload to the data owner, especially in scenarios with a large number of writers (users) and/or a high frequency of data modification operations. Considering practical scenarios wherein all users share data with each other in cloud have both read and write privileges, Wang et al. proposed a public integrity auditing scheme using ring signature-based omomorphic authenticators.

Nevertheless, the scalability of ref is limited by the group size and data size. Moreover, user revocation is not considered in this paper. In order to further enhance previous

work, another attempt was made by Wang et al However, still suffers from a non-trivial computational cost which is linear to the number of modifiers (especially for achieving high error detection probability with small data corruption rate, e.g., less then 0.1%) and the number of checking tasks, and thus is limited in scalability. In addition, user revocation is based on the assumption that the cloud node responsible for updating signatures will not be compromised nor encounter internal errors, which has been proven not always true in practice. As a matter of fact, compromisation and internal errors of the cloud node will lead to the disclosure of users' secrets, and thus causing severe attacks such as user impersonation.

Existing System

- In existing techniques that support multi-user modification. Batch auditing of multiple tasks.
- Only the data woner holds secret keys can modify the data and all other userswho share data with the data owner only have read permission. If these solutions are trivially extended to support multiple writers with data integrity assurance, the data owner has to stay online, collecting modified data from other users and regenerating authentication tags for them.
- Obviously, this kind of trivial extension will introduce a tremendous workload This kind of situation occurs many times, being it internationally or not, with existing cloud storage platforms.
- As our design efficiently supports batch auditing, we can audit all development files at the same time to save cost. Thus, our scheme can be easily applied to existing VCSs to efficient support integrity assurance without changing their original design.

Proposed System

In this we have a way of auditing the cloud server to ensure that the server stores all their latest data without any corruption. To offer such a service, a series of schemes have been proposed. However, for most of these existing schemes, only the data owner . In cloud have both read and write privileges, Wang proposed a public integrity auditing scheme using ring signature-based homomorphic authenticators. Nevertheless, the scalability of ref.Last but not least, our proposed scheme allows aggregation of integrity auditing operations for multiple tasks (files)

through our batch integrity auditing technique, which promote our scheme in terms of auditing efficiency and data corruption detection probability. The TPA refers to any party that checks the integrity of data being stored on the cloud. As our proposed scheme allows public integrity auditing, the TPA can actually be any cloud user as long as he/she has access to the public keys.

To design an efficient public integrity auditing scheme supporting multi-user modification and efficient user revocation simultaneously, we need to overcome the following major (not necessarily complete list of challenges: 1) Aggregation of individually generated authentication tags. Specifically, any user with read and write privilege should be able to modify the data and generate new authentication tags without the help of the data owner. In this context, how to aggregate tags from different users becomes a challenge problem, because the tags are signed with individual users' secret keys which are different from each other. Without aggregation, a data integrity verifier has to process tags from different modifiers one by one for an auditing task, and thus limiting the scalability of scheme. A straightforward solution to this problem is to let all users share the same secret key,so all authentication tags are in the same format and can be easily aggregated. Nevertheless, this kind of solution is limited by another challenge: 2) efficient and secure user revocation.

User revocation is a challenging issue in most security systems and usually involves disabling user secret keys. Upon user revocation, all authentication tags generated by the revoked user should be updated,and this heavy task is usually delegated to the cloud by disclosing partial secrets to it. This method, however, can lead to disclosure of secret keys of valid users once the cloud server node colludes with a revoked user. Public auditing. In practical systems, data integrity auditing can be performed not only by data owners or other group users but also by a third-party auditor or any general user who has public keys of the system. As a result, just a constant size of integrity proof information and a constant number of computational operations are needed for the verifier, no matter how large the audited file is and how many writers are associated with the data blocks. Moreover, with the novel proxy authentication tag update technique, our scheme allows secure delegation of user revocation operations to the cloud and can defeat impersonation attacks from illegitimate users. By uniquely incorporating Shamir's Secret Sharing

scheme , we further enhance the reliability of our design during the user revocation procedure. Last but not least, our proposed scheme allows aggregation of integrity auditing operations for multiple tasks (files)through our batch integrity auditing technique, which promote our scheme in terms of auditing efficiency and data corruption detection probability. Thorough analysis and extensive experimental results on Amazon EC2 cloud and various client devices (PC and mobiledevices) demonstrate the scalability and efficiency of our design.

Literature survey:

The problem of data integrity auditing in cloud have been extensively studied in past years by a number of Proof of Retrievability (POR) and Proof of Data Possession (PDP) schemes. In ref. concepts of POR and PDP were first proposed separately using RSA based homomorphic authentication tags. The efficiency of POR scheme was later enhanced by Shachametal based on the BLS (Boneh-Lynn-Shacham) signature . To further enhance the efficiency of data integrity auditing, batch integrity auditing was introduced by Wang et al.. Recently, Xu et al. and Yuan et al. proposed private and public POR schemes respectively with constant communication cost by using a nice algebraic property of polynomial. To support dynamic operations in verification, Atenieseetal proposed another private PDP scheme with symmetric encryption. A Public integrity auditingwith dynamic operations is introduced by Wang et al. based on the Merkle Hash Tree. Based on the rank information, Erwayetal. also achieve the dynamic PDP. Zhu et al. later utilized the fragment structure to save storage overhead of authentication tags with the support of dynamic data. A private POR scheme with the support of dynamic data is recently proposed by Cash et al. by utilizing Oblivious RAM. Although many efforts have been made to guarantee the integrity of data on remote server, most of them only consider single data owner who has the system secret key and is the only party allowed to modify the shared data on cloud. In order to improve the previous works to support multiple writers, Wangetal. first proposed a public integrity auditing scheme for shared data on cloud based on ring signature-based homomorphic authenticators. In their scheme, user revocation is not considered and the auditing cost grows with group size and data size. Recently, Wang et al. enhanced their previous public integrity verification scheme with the support of user revocation. However, if the cloud node responsible for tag update is compromised during user revocation process, attackers can discover

the secret keys of all other valid users. What is more, verification cost of the TPA (can also be users) in ref. [21] is significantly influenced by the error detection probability requirement and is also linear to the number of data modifier. Batch verification is not supported in their design. Therefore,this scheme is limited in its scalability.

BIBLIOGRAPHY

Good Teachers are worth more than thousand books, we have them in Our Department

References Made From:

1. User Interfaces in C#: Windows Forms and Custom Controls by Matthew MacDonald.
2. Applied Microsoft® .NET Framework Programming (Pro-Developer) by Jeffrey Richter.
3. Practical .Net2 and C#2: Harness the Platform, the Language, and the Framework by Patrick Smacchia
4. Data Communications and Networking, by Behrouz A Forouzan
5. Computer Networking: A Top-Down Approach, by James F. Kurose.
6. Operating System Concepts, by Abraham Silberschatz.
7. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep. USB-EECS-2009-28, Feb 2009
8. "The apache cassandra project," http://cassandra.apache.org/.
9. L. Lamport, "The part-time parliament," ACM Transactions on Computer Systems, vol. 16, pp. 133–169, 1998.
10. N. Bonvin, T. G. Papaioannou, and K. Aberer, "Cost-efficient and differentiated data availability guarantees in data clouds," i n Proc. of the ICDE, Long Beach, CA, USA, 2010
11. O. Regev and N. Nisan, "The popcorn market. online markets for computational resources," Decision Support Systems, vol. 28, no. 1-2, pp. 177 – 189, 2000.
12. A. Helsinger and T. Wright, "Cougaar: A robust configurable multi agent platform," in Proc. of the IEEE Aerospace Conference, 2005.
13. J. Brunelle, P. Hurst, J. Huth, L. Kang, C. Ng, D. C. Parkes,

M. Seltzer, J. Shank, and S. Youssef, "Egg: an extensible and economics-inspired open grid computing platform," in Proc. of the GECON, Singapore, May 2006.

14. J. Norris, K. Coleman, A. Fox, and G. Candea, "Oncall: Defeating spikes with a free-market application cluster," in Proc. of the International Conference on Autonomic Computing, New York, NY, USA, May 2004.
15. C. Pautasso, T. Heinis, and G. Alonso, "Autonomic resource provisioning for software business processes," Information and Software Technology, vol. 49, pp. 65–80, 2007.
16. M. Wang and T. Suda, "The bio-networking architecture: a biologically inspired approach to the design of scalable, adaptive, and survivable/available network applications," in Proc. of the IEEE Symposium on Applications and the Internet, 2001.
17. N. Laranjeiro and M. Vieira, "Towards fault tolerance in web services compositions," in Proc. of the workshop on engineering fault tolerant systems, New York, NY, USA,2007
18. C. Engelmann, S. L. Scott, C. Leangsuksun, and X. He, "Transparent symmetric active/active replication for servicelevel high availability," in Proc. of the CCGrid, 2007.
19. J. Salas, F. Perez-Sorrosal, n.-M. M. Pati and R. Jim´enez-Peris, "Ws-replication: a framework for highly available web services," in Proc. of the WWW, New York, NY, USA, 2006.

Sites Referred:

http://www.sourcefordgde.com
http://www.networkcomputing.com/
http://www.ieee.org
http://www.emule-project.net/

AUTHOR 1:-

**G.SANDHYA completed her B tech in RAMAPPA ENGINEERING COLLEGE in 2012 and completed M-Tech in VAAGDEVI ENGINEERING COLLEGE

AUTHOR 2:-

**M.SHIRISHA is working as Assistant Professor in Dept of CSE, VAAGDEVI ENGINEERING COLLEGE