

April 2013

METHOD TO ACHIEVE SECURITY AND STORAGE SERVICES IN CLOUD COMPUTING

SYED SADDAM HUSSAIN

PG Student, Audisankara College, Nellore, SYEDSADDAMUSSAIN@gmail.com

R.VINOD KUMAR

ME Assistant Professor, Audisankara College India, R.VINOD@gmail.com

Follow this and additional works at: <https://www.interscience.in/ijcns>



Part of the [Computer Engineering Commons](#), and the [Systems and Communications Commons](#)

Recommended Citation

HUSSAIN, SYED SADDAM and KUMAR, R.VINOD (2013) "METHOD TO ACHIEVE SECURITY AND STORAGE SERVICES IN CLOUD COMPUTING," *International Journal of Communication Networks and Security*. Vol. 2 : Iss. 2 , Article 3.

Available at: <https://www.interscience.in/ijcns/vol2/iss2/3>

This Article is brought to you for free and open access by Interscience Research Network. It has been accepted for inclusion in International Journal of Communication Networks and Security by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.

METHOD TO ACHIEVE SECURITY AND STORAGE SERVICES IN CLOUD COMPUTING

SYED SADDAM HUSSAIN¹, R.VINOD KUMAR²

¹PG Student, Audisankara College, Nellore, ²ME Assistant Professor, Audisankara College

Abstract- Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing users' physical possession of their outsourced data, which inevitably poses new security risks toward the correctness of the data in cloud. In order to address this new problem and further achieve a secure and dependable cloud storage service, we propose in this paper a flexible distributed storage integrity auditing mechanism, utilizing the homomorphism token and distributed erasure-coded data. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result not only ensures strong cloud storage correctness guarantee, but also simultaneously achieves fast data error localization, i.e., the identification of misbehaving server. Considering the cloud data are dynamic in nature, the proposed design further supports secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append. Analysis shows the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks)

I. INTRODUCTION:

SEVERAL trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the software as a service (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. The increasing network bandwidth and reliable yet flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside solely on remote data centers. Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management. The pioneer of Cloud Computing vendors, Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) are both well known examples. While these internet-based online services do provide huge amounts of storage space and customizable computing resources, this computing platform shift, however, is eliminating the responsibility of local machines for data maintenance at the same time. As a result, users are at the mercy of their cloud service providers for the availability and integrity of their data. On the one hand, although the cloud infrastructures are much more powerful and reliable than personal computing devices, broad range of both internal and external threats for data integrity still exist. Examples of outages and data loss incidents of noteworthy cloud storage services appear from time to time. On the other hand, since users may not retain a local copy of outsourced data, there exist various incentives for cloud service providers (CSP) to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For example, to increase the profit margin by reducing cost, it is possible for CSP to discard rarely accessed

data without being detected in a timely fashion. Similarly, CSP may even attempt to hide data loss incidents so as to maintain a reputation. Therefore, although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, its lacking of offering strong assurance of data integrity and availability may impede its wide adoption by both enterprise and individual cloud users. In order to achieve the assurances of cloud data integrity and availability and enforce the quality of cloud storage service, efficient methods that enable on-demand data correctness verification on behalf of cloud users have to be designed. However, the fact that users no longer have physical possession of data in the cloud prohibits the direct adoption of traditional cryptographic primitives for the purpose of data integrity protection. Hence, the verification of cloud storage correctness must be conducted without explicit knowledge of the whole data files. Meanwhile, cloud storage is not just a third party data warehouse. The data stored in the cloud may not only be accessed but also be frequently updated by the users, including insertion, deletion, modification, appending, etc. Thus, it is also imperative to support the integration of this dynamic feature into the cloud storage correctness assurance, which makes the system design even more challenging. Last but not the least, the deployment of Cloud Computing is powered by data centers running in a simultaneous, cooperated and distributed manner. It is more advantages for individual users to store their data redundantly across multiple physical servers so as to reduce the data integrity and availability threats. Thus, distributed protocols for storage correctness assurance will be of most importance in achieving robust and secure cloud storage systems. However, such important area remains to be fully explored in the literature. Recently, the importance of ensuring

the remote data integrity has been highlighted by the following research works under different system and security models. These techniques, while can be useful to ensure the storage correctness without having users possessing local data, are all focusing on single server scenario. They may be useful for quality-of-service testing, but does not guarantee the data availability in case of server failures. Although direct applying these techniques to distributed storage (multiple servers) could be straightforward, the resulted storage verification overhead would be linear to the number of servers. As an complementary approach, researchers have also proposed distributed protocols for ensuring storage correctness across multiple servers or peers. However, while providing efficient cross server storage verification and data availability insurance, these schemes are all focusing on static or archival data. As a result, their capabilities of handling dynamic data remains unclear, which inevitably limits their full applicability in cloud storage scenarios

Existing System:

In the past, software had to be installed in an infrastructure close to end users.

- 1) In an existing system, we doesn't measure the correctness of cloud storage. In case any of the corruption might be happened on cloud mean we can't get the original information's everything going to be lost.
- 2) There are no recover methods present their existing systems. It is more advantages for individual users to store their data redundantly across multiple physical servers so as to reduce the data integrity and availability threats. Thus, distributed protocols for storage correctness assurance will be of most importance in achieving robust and secure cloud storage systems.

Proposed System:

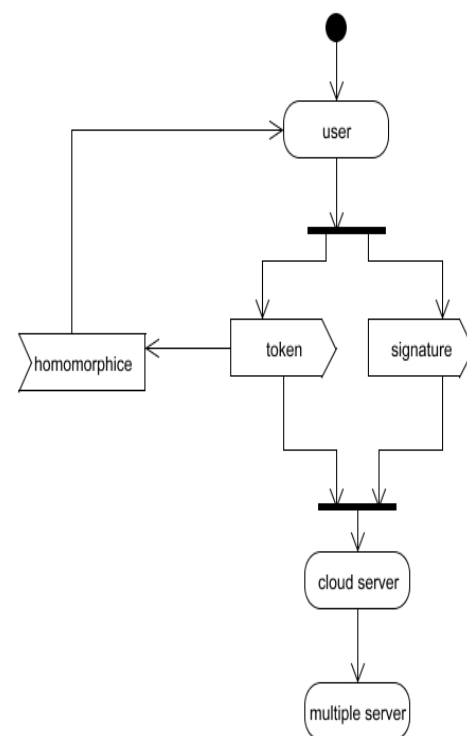
- 1) If any of a cloud get affected means with the help of the third party auditor We easily recover the data's from cloud.
- 2) This So that we measure cloud correctness based on integrity auditing mechanism it helps to secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append.
- 3) With the help of that we ensure our data's always going to be a right one if any mismatch occurs means we easily find out were ever corruption are made based on that we recover the from cloud as much as possible.

System Features

To ensure the security and dependability for cloud data storage under the aforementioned adversary model, we aim to design efficient mechanisms for dynamic data verification and operation and achieve

the following goals: Storage correctness: to ensure users that their data are indeed stored appropriately and kept intact all the time in the cloud. Fast localization of data error: to effectively locate the malfunctioning server when data corruption has been detected. Dynamic data support: to maintain the same level of storage correctness assurance even if users modify, delete or append their data files in the cloud. Dependability: to enhance data availability against Byzantine failures, malicious data modification and server colluding attacks, i.e. minimizing the effect brought by data errors or server failures. Lightweight: to enable users to perform storage correctness checks with minimum overhead.

Activity Diagram



CONCLUSION

In this paper, we investigate the problem of data security in cloud data storage, which is essentially a distributed storage system. To achieve the assurances of cloud data integrity and availability and enforce the quality of dependable cloud storage service for users, we propose an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append. We rely on erasure-correcting code in the file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., whenever data corruption has been detected during the storage correctness verification across the

distributed servers, we can almost guarantee the simultaneous identification of the misbehaving server(s). Considering the time, computation resources, and even the related online burden of users, we also provide the extension of the proposed main scheme to support third-party auditing, where users can safely delegate the integrity checking tasks to thirdparty auditors and be worry-free to use the cloud storage services. Through detailed security and extensive experiment results, we show that our scheme is highly efficient and resilient to Byzantine failure, malicious data modification attack, and even server colluding attacks.

REFERENCES

- [1] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS '09), pp. 1-9, July 2009.
- [2] Amazon.com, "Amazon Web Services (AWS)," <http://aws.amazon.com>, 2009.
- [3] Sun Microsystems, Inc., "Building Customer Trust in Cloud Computing with Transparent Security," https://www.sun.com/offers/details/sun_transparency.xml, Nov. 2009.
- [4] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.
- [5] M. Arrington, "Gmail Disaster: Reports of Mass Email Deletions," <http://www.techcrunch.com/2006/12/28/gmail-disasterreports-of-mass-email-deletions>, Dec. 2006.
- [6] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors," <http://www.techcrunch.com/2008/07/10/mediamaxthelinkup-closesits-doors>, July 2008.
- [7] Amazon.com, "Amazon S3 Availability Event: July 20, 2008," <http://status.aws.amazon.com/s3-20080720.html>, July 2008.
- [8] S. Wilson, "Appengine Outage," http://www.cio-weblog.com/50226711/appengine_outage.php, June 2008.
- [9] B. Krebs, "Payment Processor Breach May Be Largest Ever," http://voices.washingtonpost.com/securityfix/2009/01/payment_processor_breach_may_b.html, Jan. 2009.
- [10] A. Juels and B.S. Kaliski Jr., "PORs: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, Oct. 2007.
- [11] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 598-609, Oct. 2007.
- [12] M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop Hot Topics in Operating Systems (HotOS '07), pp. 1-6, 2007.
- [13] M.A. Shah, R. Swaminathan, and M. Baker, "Privacy-Preserving Audit and Extraction of Digital Contents," Cryptology ePrint Archive, Report 2008/186, <http://eprint.iacr.org>, 2008.
- [14] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. Fourth Int'l Conf. Security and Privacy in Comm. Netowrks (SecureComm '08), pp. 1-10, 2008.
- [15] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proc. 14th European Conf. Research in Computer Security (ESORICS '09), pp. 355-370, 2009. WANG ET AL.: TOWARD SECURE AND DEPENDABLE STORAGE SERVICES IN CLOUD COMPUTING 231
- [16] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), pp. 213-222, 2009.
- [17] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (Asiacrypt '08), pp. 90-107, 2008.
- [18] K.D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Proc. ACM Workshop Cloud Computing Security (CCSW '09), pp. 43-54, 2009.
- [19] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-Replica Provable Data Possession," Proc. IEEE 28th Int'l Conf. Distributed Computing Systems (ICDCS '08), pp. 411-420, 2008.
- [20] Y. Dodis, S. Vadhan, and D. Wichs, "Proofs of Retrievability via Hardness Amplification," Proc. Sixth Theory of Cryptography Conf. (TCC '09), Mar. 2009.
- [21] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 5, pp. 847-859, 2011.
- [22] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, preprint, 2012, doi:10.1109/TC.2011.245.
- [23] K.D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Proc. ACM Conf. Computer and Comm. Security (CCS '09), pp. 187-198, 2009.
- [24] T. Schwarz and E.L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS '06), pp. 12-12, 2006.
- [25] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard, "A Cooperative Internet Backup Scheme," Proc. USENIX Ann. Technical Conf. (General Track), pp. 29-41, 2003.
- [26] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance and Proactive Recovery," ACM Trans. Computer Systems, vol. 20, no. 4, pp. 398-461, 2002.
- [27] L. Carter and M. Wegman, "Universal Hash Functions," J. Computer and System Sciences, vol. 18, no. 2, pp. 143-154, 1979.
- [28] J. Hendricks, G. Ganger, and M. Reiter, "Verifying Distributed Erasure-Coded Data," Proc. 26th ACM Symp. Principles of Distributed Computing, pp. 139-146, 2007.
- [29] J.S. Plank and Y. Ding, "Note: Correction to the 1997 Tutorial on Reed-Solomon Coding," Technical Report CS-03-504, Univ. of Tennessee, Apr. 2003.
- [30] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. IEEE INFOCOM, Mar. 2010.
- [31] C. Wang, K. Ren, W. Lou, and J. Li, "Towards Publicly Auditible Secure Cloud Data Storage Services," IEEE Network Magazine, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.

- [32] R.C. Merkle, "Protocols for Public Key Cryptosystems," Proc. IEEE Symp. Security and Privacy, 1980.
- [33] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance," Proc. IEEE INFOCOM, Apr. 2009.
- [34] J.S. Plank, S. Simmerman, and C.D. Schuman, "Jerasure: A Library in C/C++ Facilitating Erasure Coding for Storage Applications - Version 1.2," Technical Report CS-08-627, Univ. of Tennessee, Aug. 2008.
- [35] M. Bellare, R. Canetti, and H. Krawczyk, "Keying Hash Functions for Message Authentication," Proc. 16th Ann. Int'l Cryptology Conf. Advances in Cryptology (Crypto '96), pp. 1-15, 1996.
- [36] M. Bellare, O. Goldreich, and S. Goldwasser, "Incremental Cryptography: The Case of Hashing and Signing," Proc. 14th Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO '94), pp. 216-233, 1994.
- [37] D.L.G. Filho and P.S.L.M. Barreto, "Demonstrating Data Possession and Uncheatable Data Transfer," Cryptology ePrint Archive, Report 2006/150, <http://eprint.iacr.org>, 2006.

