Ensuring Data Integrity in Cloud Computing

[1]N.Praveen Kumarga [2]D.Sireesha

[1,2]D ept. of CSE, Pragati Engineering College, Surampalem, Kak inada, AP, India

Abstract

Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. It moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trust worthy. This unique paradigm brings about many new security challenges, which have not been well understood. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA eliminates the involvement of the client through the auditing of whether his data stored in the cloud are indeed intact, which can be important in achieving economies of scale for Cloud Computing. In this paper we introduce Dynamic Intelligent Server (DIS), which shares CSP work and enhances the dynamic data storage allocation. we follow the methods that support for data dynamics via the most general forms of data operation, such as block modification, insertion, and deletion, is also a significant step toward practicality, since services in Cloud Computing are not limited to archive or backup data only. While prior works on ensuring remote data integrity often lacks the support of either public auditability or dynamic data operations, this paper achieves both. We first identify the difficulties and potential security problems of direct extensions with fully dynamic data updates from prior works and then show how to construct an elegant verification scheme for the seamless integration of these two salient features in our protocol design. In particular, to achieve efficient data dynamics, we improve the existing proof of storage models by manipulating the classic Merkle Hash Tree construction for block tag authentication. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multiuser setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis show that the proposed schemes are highly efficient and provably secure.

Keywords

Data storage, public auditability, data dynamics, cloud computing, DB Server Computation.

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. Meanwhile, the increasing network bandwidth and reliable yet flexible network connections make it even possible that clients can now subscribe high-quality services from data and software that reside solely on remote data centers. Although envisioned as a promising service platform for the Internet, this new data storage paradigm in "Cloud" brings about many challenging design issues which have profound influence on the security and performance of the overall system. One of the biggest concerns with cloud data storage is that of data integrity verification at untrusted servers. For example, the storage service provider, which experiences Byzantine failures occasionally, may decide to hide the data errors from the clients for the benefit of their own. What is more serious is that for saving money and storage space the service provider might neglect to keep or deliberately delete rarely accessed data files which belong to an ordinary client. Consider the large size of the outsourced electronic data and the client's constrained resource capability, the core of the problem can be generalized as how can the client find an efficient way to perform periodical integrity verifications without the local copy of data files. In order to solve the problem of data integrity checking, many schemes are proposed under different systems and security models [2-11]. In all these works, great efforts are made to design solutions that meet various requirements: high scheme efficiency, stateless verification, unbounded use of queries and retrievability of data, etc. Considering the role of the verifier in the model, all the schemes presented before fall into two categories: private auditability and public auditability. Although schemes with private auditability can achieve higher scheme efficiency, public auditability allows anyone, not just the client (data owner), to challenge the cloud server for correctness of data storage while keeping no private information. Then, clients are able to delegate the evaluation of the service performance to an independent third party auditor (TPA), without devotion of their computation resources. In the cloud the clients themselves are unreliable or may not be able to afford the overhead of performing frequent integrity checks. Thus, for practical use, it seems more rational to equip the verification protocol with public auditability, which is expected to play a more important role in achieving economies of scale for Cloud Computing. Moreover, for efficiency consideration, the outsourced

Manuscript received September 5, 2014
Manuscript revised September 20, 2014

data themselves should not be required by the verifier for the verification purpose. Another major concern among previous designs is that of supporting dynamic data operation for cloud data storage applications. In Cloud Computing, the remotely stored electronic data might not only be accessed but also updated by the clients, e.g., through block modification, deletion, insertion, etc. Unfortunately, the state of the art in the context of remote data storage mainly focus on static data files and the importance of this dynamic data updates has received limited attention so far [2-5, 7, 10, 12]. Moreover, as will be shown later, the direct extension of the current Provable Data Possession (PDP) [2] or proof of retrievability (PoR) [3-4] schemes to support data dynamics may lead to security loopholes. Although there are many difficulties faced by researchers, it is well believed that supporting dynamic data operation can be of vital importance to the practical application of storage outsourcing services. In view of the key role of public auditability and data dynamics for cloud data storage, we propose an efficient construction for the seamless integration of these two components in the protocol design. Our contribution can be summarized as follows:

1. We follow the public auditing system of data storage security in Cloud Computing, and propose a protocol supporting for fully dynamic data operations, especially to support block insertion, which is missing in most existing schemes.
2. We follow scheme to support scalable and efficient public auditing in Cloud Computing. In particular, our scheme achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA.
3. We prove the security of our proposed construction and justify the performance of our scheme through concrete implementation and comparisons with the state of the art.
4. We extend the approach to support the performance while achieve efficient data dynamics. we improve the existing proof of storage models

II. Related Work

Recently, much of growing interest has been pursued in the context of remotely stored data verification [2-10, 12-15]. Ateniese et al. [2] are the first to consider public auditability in their defined "provable data possession" model for ensuring possession of files on untrusted storages. In their scheme, they utilize RSA- based homomorphic tags for auditing outsourced data, thus public auditability is achieved However, the system imposes a priori bound on the number of queries and does not support fully dynamic data operations, i.e., it only allows very basic block operations with limited functionality, and block insertions cannot be supported. In [13], Wang et al. consider dynamic data storage in a distributed scenario, and the proposed challenge-response protocol can both determine the data correctness and locate possible errors. Similar to [12], they only consider partial support for dynamic data operation. Juels and Kaliski [3] describe a "proof of retrievability" model, where spot-checking and error-correcting codes are used to ensure both "possession" and "retrievability" of data files on archive service systems. Specifically, some special blocks called "sentinels" are randomly embedded into the data file F for detection purpose, and F is further encrypted to protect the positions of these special blocks. However, like [12], the number of queries a client can perform is also a fixed priori, and the introduction of precomputed "sentinels" prevents the development of realizing dynamic data updates To support updates, especially for block insertion, they eliminate the index information in the "tag" computation in Ateniese's PDP model [2] and employ authenticated skip list data structure to authenticate the tag information of challenged or updated blocks first before the verification procedure. However, the efficiency of their scheme remains unclear. Although the existing schemes aim at providing integrity verification for different data storage systems, the problem of supporting both public auditability and data dynamics has not been fully addressed. How to achieve a secure and efficient design to seamlessly integrate these two important components for data storage service remains an open challenging task in Cloud Computing. We revise the paper a lot and add more technical details as compared to [1]. In this paper we introcing the Dynamic Intelligent Server , which enhances the performance of CSP we follow two basic solutions (i.e., the MAC-based and signature- based schemes) for realizing data auditability and discuss their demerits in supporting public auditability and data dynamics. Second, we generalize the support of data dynamics to both PoR and PDP models and discuss the impact of dynamic data operations on the overall system efficiency both. In particular, we emphasize that while dynamic data updates can be performed efficiently in PDP models more efficient protocols need to be designed for the update of the encoded files in PoR models. For completeness, the designs for distributed data storage security data auditing scheme for the single client and explicitly include a concrete description of the multiclient data auditing scheme. We also redo the whole experiments and present the performance comparison between the multiclient data auditing scheme and the individual auditing scheme

III. Proposed Work

We present enhanced performance server called Dynamic intelligent Server which follows security

protocols for cloud data storage service with the aforementioned research goals in mind. We start with some basic solutions aiming to provide integrity assurance of the cloud data and discuss their demerits. Then, we present our protocol which supports public auditability and data dynamics. We also show how to extent our main scheme to support batch auditing for TPA upon delegations from multiusers. In This paper, we introduce Dynamic Intelligent Server DIS which enhances the performance of cloud storage server, DIS process the storage sever to compute the data storage allocations of outsourced data from the heterogeneous client data files .The DIS server dynamically works with the cloud storage server by internally maintaining the Cloud service metrics. The proposed DIS server enhances the performance of overall Cloud computing services. The Intelligent server continuously monitoring the datastores and various heterogeneous client data flows.

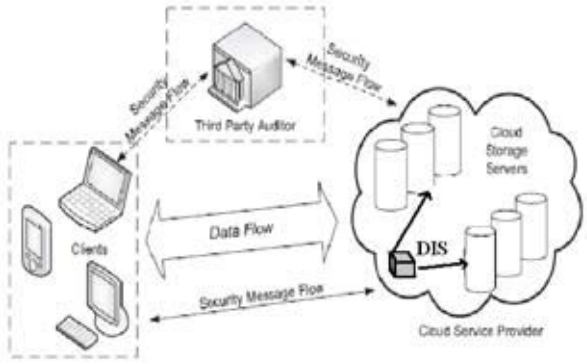


Fig. 1: Cloud Data Storage rchitecture

A. System Model

A representative network architecture for cloud data storage is illustrated in fig. 1. Three different network entities can be identified as follows:

1. Client

An entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations;

2. Cloud Storage Server (CSS) and Dynamic Intelligent Server (DIS)

An entity, which is managed by Cloud Service Provider (CSP), has significant storage space and computation resource to maintain the clients' data and enhances performance of CSP;

3. Third Party Auditor

An entity, which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request.

B. Design Goals

Our design goals can be summarized as the following:

1. Public auditability for storage correctness assurance: to allow anyone, not just the clients who originally stored the file on cloud servers, to have the capability to verify the correctness of the stored data on demand.
2. Dynamic data operation support: to allow the clients to perform block-level operations on the data files while maintaining the same level of data correctness assurance. The design should be as efficient as possible so as to ensure the seamless integration of public auditability and dynamic data operation support.
3. Blockless verification: no challenged file blocks should be retrieved by the verifier (e.g., TPA) during verification process for efficiency concern.
4. Enhanced performance during cloud computing using DI Server.

C. Construction

To effectively support public auditability without having to retrieve the data blocks themselves, we resort to the homomorphic authenticator technique [2, 4]. Homomorphic authenticators are unforgeable metadata generated from individual data blocks, which can be securely aggregated in such a way to assure a verifier that a linear combination of data blocks is correctly computed by verifying only the aggregated authenticator. In our design, we propose to use PKC-based homomorphic authenticator (e.g., BLS signature [4] or RSA signature-based authenticator [2]) to equip the verification protocol with public auditability. In the following description, we present the BLS-based scheme to illustrate our design with data dynamics support. As will be shown, the schemes designed under BLS construction can also be implemented in RSA construction. In the discussion of Section 3.4, we show that direct extensions of previous work [2, 4] have security problems, and we believe that protocol design for supporting dynamic data operation is a major challenging task for cloud storage systems. Now we start to present the main idea behind our scheme. We assume that file F (potentially encoded using Reed-Solomon codes [18]) is divided into

n blocks. The procedure of our protocol execution is as follows: Integrity Verification, Dynamic Data Operation with Integrity Assurance, Batch Auditing for Multiclient Data.

Table 1: Protocols for Default Integrity Verification

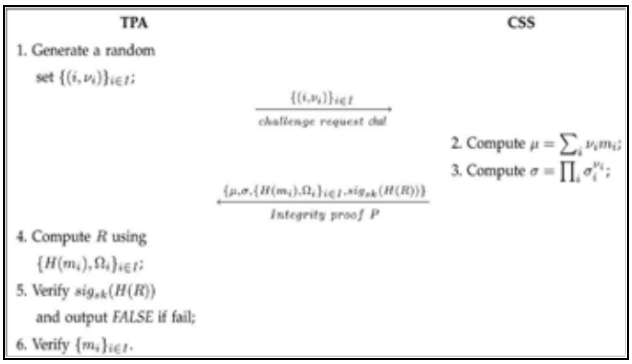
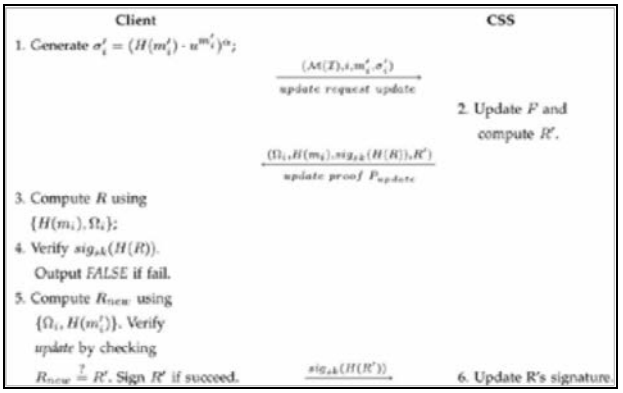


Table 2: The Protocol for Provable Data Update (Modification and Insertion)



IV. Performance Analysis

We list the features of our proposed scheme in Table 3 and make a comparison of our scheme and the state of the art. The scheme in [14], extends the original PDP [2], to support data dynamics using authenticated skip list. Thus, we call it DPDP scheme thereafter. For the sake of completeness, we implemented both our BLS and RSA-based instantiations as well as the state-of-the-art scheme [14], in Linux. Our experiment is conducted using C on a system with an Intel Core 2 processor running at 2.4 GHz, 768 MB RAM, and a 7200 RPM Western Digital 250 GB Serial ATA drive with an 8 MB buffer. We used Dynamic Intelligent Server have computational capabilities over the cloud computing server Which shows the overall performance gain over previous system.

The block size for RSA-based instantiation and scheme in [14] is chosen to be 4 KB From Table 4, it can be observed that the overall performance of the three schemes are comparable to each other. Due to the smaller block size (i.e., 20 bytes) compared to RSA-based instantiation, our BLS-based instantiation is more than two times faster than the other two in terms of server computation time.

Table 3: Performance Comparison under Different Tolerance Rate of File Corruption for 1GB File

	Our BLS-based instantiation		Our RSA-based instantiation		[14]
Metric \ Rate-ρ	99%	97%	99%	97%	99%
Sever comp. time (ms)	2.45	0.11	10.81	0.55	14.13
Verifier comp. time (ms)	806.01	284.17	779.10	210.47	782.56
Comm. cost (KB)	239	80	223	76	280

V. Conclusion

To ensure cloud data storage security, it is critical to enable a TPA to evaluate the service quality from an objective and independent perspective. Public auditability also allows clients to delegate the integrity verification tasks to TPA while they themselves can be unreliable or not be able to commit necessary computation resources performing continuous verifications. Another major concern is how to construct verification protocols that can accommodate dynamic data files. In this paper, we explored the problem of providing simultaneous public auditability and data dynamics for remote data integrity check in Cloud Computing and the performance of the Cloud Storage Servers. Our construction is deliberately designed to meet these two important goals while efficiency being kept closely in mind. To achieve efficient data dynamics, we improve the existing proof of storage models by manipulating the classic Merkle Hash Tree construction for block tag authentication. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multiuser setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure.

References

[1] IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 5, May 2011

[2] Q. Wang, C. Wang, J. Li, K. Ren, W. Lou,"Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing", Proc. 14th European Symp. Research in Computer Security (ESORICS '09), pp. 355-370, 2009.

[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, D. Song,"Provable Data Possession at Untrusted Stores", Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 598-609, 2007.

[4] A. Juels, B.S. Kaliski Jr.,"Pors: Proofs of Retrievability for Large Files", Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, 2007.
[5] H. Shacham 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.
[6] K.D. Bowers, A. Juels, A. Oprea,"Proofs of Retrievability: Theory and Implementation", Report 2008/175, Cryptology ePrint Archive, 2008.
[7] M. Naor, G.N. Rothblum,"The Complexity of Online Memory Checking", Proc. 46th Ann. IEEE Symp. Foundations of Computer Science (FOCS '05), pp. 573-584, 2005.
[8] E.-C. Chang, J. Xu,"Remote Integrity Check with Dishonest Storage Server", Proc. 13th European Symp. Research in Computer Security (ESORICS '08), pp. 223-237, 2008
[9] M.A. Shah, R. Swaminathan, M. Baker,"Privacy-Preserving Audit and Extraction of Digital Contents", Report 2008/186, Cryptology ePrint Archive, 2008.
[10] A. Oprea, M.K. Reiter, K. Yang,"Space-Efficient Block Storage Integrity," Proc. 12th Ann. Network and Distributed System Security Symp. (NDSS '05), 2005.
[11] T. Schwarz and E.L. Miller,"Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage", Proc. 26th IEEE Int'l Conf. Distributed Computing Systems (ICDCS'06), p. 12, 2006.
[12] Q. Wang, K. Ren, W. Lou, Y. Zhang,"Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance", Proc. IEEE INFOCOM, pp. 954-962, Apr. 2009.
[13] G. Ateniese, R.D. Pietro, L.V. Mancini, G. Tsudik, "Scalable and Efficient Provable Data Possession", Proc. Fourth Int'l Conf. Security and Privacy in Comm. Networks (SecureComm '08), pp. 1-10,2008.
[14] 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), 2009.
[15] C. Erway, A. Kupcu, C. Papamanthou, R. Tamassia, "Dynamic Provable Data Possession", Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), 2009.
[16] K.D. Bowers, A. Juels, A. Oprea,"Hail: A High-Availability and Integrity Layer for Cloud Storage", Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), pp. 187-198, 2009.
[17] D. Boneh, B. Lynn, H. Shacham,"Short Signatures from the Weil Pairing", Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '01), pp. 514-532, 2001.
[18] R.C. Merkle, "Protocols for Public Key Cryptosystems", Proc. IEEE Symp. Security and Privacy, pp. 122-133, 1980.
[19] S. Lin, D.J. Costello,"Error Control Coding", second ed., Prentice- Hall, 2004.
[20] M. Bellare, P. Rogaway,"Random Oracles are Practical: A Paradigm for Designing Efficient Protocols", Proc. First ACM Conf. Computer and Comm. Security (CCS '93), pp. 62-73, 1993.
[21] D. Boneh, C. Gentry, B. Lynn, H. Shacham,"Aggregate and Verifiably Encrypted Signatures from Bilinear Maps", Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic techniques (Eurocrypt '03), pp. 416-432, 2003.

N.Praveen Kumar is a student of Pragati Engineering College, surampalem,kakinada. Presently he is pursuing his M.Tech (C.S.E) from this college and he received his Bachelor of Information technology from Pragati Engineering College, surampalem,kakinada in the year 2010. His area of interest includes Computer Networks,cloud computing and Object oriented Programming languages in Computer Applications.

D. Sireesha is well known Author and excellent teacher, Received M.Tech (CSE) from JNTU university is working as Associate Professor in Pragati Engineering College. She has several years of teaching experience in various engineering colleges. To his credit couple of publications both national and international conferences /journals . Her area of Interest includes Data Warehouse and Data Mining, information security, image processing and cloud computing.