ISSN 2395-695X (Print)
ISSN 2395-695X (Online)
International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST)

Vol. 2, Special Issue 10, March 2016

DATA INTEGRITY AND AUTO HEALING USING MULTIPLE CHUNKS IN CLOUD

[1]R.Sudhakar,[2]Nithya Devi, [3]K.Sripriya
[1]Assistant Professor, [2,3]UG Scholar, Nandha College of Technology, Erode

ABSTRACT

Increasingly more and more organizationsare opting for outsourcing data to remote cloud service providers(CSPs). Customers can rent the CSPs storage infrastructureto store andretrieve almost unlimited amount of data bypaying fees metered in gigabyte/month. For an increased level ofscalability, availability, and durability, some customers may wanttheir data to be replicated on multiple servers across multipledata centers. The more copies the CSP is asked to store, themore fees the customers are charged. Therefore, customers needto have a strong guarantee that the CSP is storing all data copiesthat are agreed upon in the service contract, and all these copiesare consistent with the most recent modifications issued by thecustomers.

Providing simultaneous public audit ability and data dynamics for remote data integrity check in Cloud Computing. The existing method construction is deliberately designed to meet these two important goals while efficiency being kept closely in mind. To achieve efficient data dynamics, we improves the existing proof of storage models by manipulating the classic storage system is construction for chunk based tag authentication.

The chunk based codes and construct chunk based-DIP codes, which allow clients to remotely verify the integrity of random subsets of long-term archival data under a multiserver setting. The chunk based codes preserve fault tolerance and repair traffic saving in cloud. Also, we assume only a thin-cloud interface meaning that servers only need to support standard read/ write functionalities. The auto healing method is used to change the corrupted data in the cloud system.

1. INTRODUCTION

Cloud computing, or **the cloud**, is a colloquial expression used to describe a variety of different types of computing concepts that involve a large number of computers connected through a real-time communication network such as the Internet. Cloud computing is a term without a commonly accepted unequivocal scientific or technical definition. In science, cloud computing is a synonym for distributed computing over a network and means the ability to run a program on many connected computers at the same time. The phrase is also, more commonly used to refer to network-based services which appear to be provided by real server hardware, which in fact are served up by virtual hardware, simulated by software running on one or more real machines. Such virtual servers do not physically exist and can therefore be moved around and scaled up (or down) on the fly without affecting the end user - arguably, rather like a cloud.

All Rights Reserved © 2016 IJARBEST

The popularity of the term can be attributed to its use in marketing to sell hosted services in the sense of application service provisioning that run client server software on a remote location. Cloud computing relies on sharing of resources to achieve coherence and economies of scale similar to a utility (like the electricity grid) over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services.

The cloud also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically re-allocated per demand. This can work for allocating resources to users. For example, a cloud computer facility, which serves European users during European business hours with a specific application (e.g. email) while the same resources are getting reallocated and serve North American users during North America's business hours with another application (e.g. web server). This approach should maximize the use of computing powers thus reducing environmental damage as well since less power, air conditioning, Rackspace , etc. is required for a variety of functions. The term "moving to cloud" also refers to an organization moving away from a traditional CAPEX model (buy the dedicated hardware and depreciate it over a period of time) to the OPEX model (use a shared cloud infrastructure and pay as you use it). Proponents claim that cloud computing allows companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand.

1.1 CLOUD COMPUTING

While cloud computing makes these advantages more appealing than ever, it also brings new and challenging security threats toward users' outsourced data. Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time. Second, there do exist various motivations for CSP to behave unfaithfully toward the cloud users regarding their outsourced data status. For examples, CSP might reclaim storage for monetary reasons by discarding data that have not been or are rarely accessed, or even hide data loss incidents to maintain a reputation. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the success of cloud architecture.

To fully ensure the data integrity and save the cloud users' computation resources as well as online burden, it is of critical importance to enable public auditing service for cloud data storage, so that users may resort to an independent third-party auditor (TPA) to audit the outsourced data when needed. The TPA, who has expertise and capabilities that users do not, can periodically check the integrity of all the data stored in the cloud on behalf of the users, which provides a much more

All Rights Reserved © 2016 IJARBEST

easier and affordable way for the users to ensure their storage correctness in the cloud. Moreover, in addition to help users to evaluate the risk of their subscribed cloud data services, the audit result from TPA would also be beneficial for the cloud service providers to improve their cloud-based service platform, and even serve for independent arbitration purposes. In a word, enabling public auditing services will play an important role for this nascent cloud economy to become fully established; where users will need ways to assess risk and gain trust in the cloud.

2. PROPOSED SYSTEM

Providing simultaneous public audit ability and data dynamics for remote data integrity check in Cloud Computing. The existing method 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 control the cloud storage usingchunk based-DIP codes construction for chunk tag authentication. To support efficient handling of multiple auditing tasks, we further explore the technique of digital signature to extend our main result into a multiuser setting, where TPA can perform multiple auditing tasks in the cloud system. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure.

The chunk based-DIP codes which allow the clients to remotely verify the integrity of random subsets of long-term archival data under a multiserver setting in cloud. The chunk based-DIP codes preserve fault tolerance and repair traffic saving as in chunk based-DIP codes. Also, we assume only a thin-cloud interface meaning that servers only need to support standard read/ write functionalities. This adds to the portability of chunk based-DIP codes and allows simple deployment in general types of storage services.

2.1Advantage

- Chunk based file system
- Auto healing for the corrupted file information
- Do not download full file for auditing
- Chunk based signature is handling
- Less maintains cost

3. PROPOSED SYSTEM ALGORITHMS

- User interface design
- Cloud storage
- FMSR
- Third party auditor
- Cloud client

3.1 USERINTERFACE DESIGN

The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals—what is often called user-centered design. Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design may be utilized to support its usability. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

Interface design is involved in a wide range of projects from computer systems, to cars, to commercial planes; all of these projects involve much of the same basic human interactions yet also require some unique skills and knowledge.

3.2CLOUD STORAGE

Cloud Storage is a model of networked computer data storage where data is stored on multiple virtual servers, generally hosted by third parties, rather than being hosted on dedicated servers. Hosting companies operate large data centers; and people who require their data to be hosted buy or lease storage capacity from them and use it for their storage needs. The data center operators, in the background, virtualize the resources according to the requirements of the customer and expose them as virtual servers, which the customers can themselves manage. Physically, the resource may span across multiple servers.

3.3FUNCTIONAL MINIMUM-STORAGE REGENERATING (FMSR)

The implementation of functional minimum-storage regenerating (FMSR) codes and constructs FMSR-DIP codes, which allow clients to remotely verify the integrity of random subsets of long-term archival data under a multiserver setting. FMSR-DIP codes preserve fault tolerance and repair traffic saving as in FMSR codes. This method DIP scheme is developed. FMSR codes belong to maximum distance separable (MDS) codes. An MDS code is defined by the parameters ðn; kÞ, where k < n. It encodes a file F of size jFj into n pieces of size jFj=k each. Anðn; kÞ-MDS code states that the original file can be reconstructed from any k out of n pieces (i.e., the total size of data required is jFj). An extra feature of FMSR codes is that a specific piece can be reconstructed from data of size less than jFj. FMSR codes are built on regenerating codes which minimize the repair traffic while preserving the MDS property.

3.4THIRD PARTY AUDITOR

TPA in possession of the public key can act as a verifier. We assume that TPA is unbiased while the server is untrusted. For application purposes, the clients may interact with the cloud servers via CSP to access or retrieve their pre-stored data. More importantly, in practical scenarios, the client may frequently perform block-level operations on the data files. The most general forms of these operations we consider in this paper are modification, insertion, and deletion. 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.

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. Blockless verification: no challenged file blocks should be retrieved by the verifier (*e.g.*, TPA) during verification process for efficiency concern.

3.5CLOUD CLIENT

A cloudclient consists of computer hardware and/or computer software that relies on cloud computing for application delivery, or that is specifically designed for delivery of cloud services and that, in either case, is essentially useless without it. Examples include some computers, phones and other devices, operating systems and browsers.

4. CONCLUSION

In this paper, we explored the problem of providing simultaneous public auditability and data dynamics for remote data integrity check in Cloud Computing. 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.

Given the popularity of outsourcing archival storage to the cloud, it is desirable to enable clients to verify the integrity of their data in the cloud. We design and implement a DIP scheme for the FMSR codes under a multiserver setting. We construct FMSR-DIP codes, which preserve the fault tolerance and repair traffic saving properties of FMSR codes. To understand the practicality of FMSRDIP codes, we analyze the security strength via mathematical modeling and evaluate the running time overhead via testbed experiments. We show how FMSR-DIP codes trade between performance and security under different parameter settings. The source code of the implementation of our FMSRDIP codes is available at: http://ansrlab.cse. cuhk.edu.hk/ software/fmsrdip.

REFERENCES

[1] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon, “RACS: A Case for Cloud Storage Diversity,” Proc. First ACM Symp. Cloud Computing (SoCC ’10), 2010.

[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “A View of Cloud Computing,” Comm. ACM, vol. 53, no. 4, pp 50-58, 2010.

[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, and D. Song, “Remote Data Checking Using Provable Data Possession,” ACM Trans. Information and System Security, vol. 14, article 12, May 2011.

All Rights Reserved © 2016 IJARBEST

[4] K. Bowers, A. Juels, and A. Oprea, “HAIL: A High-Availability and Integrity Layer for Cloud Storage,” Proc. 16th ACM Conf. Computer and Comm. Security (CCS ’09), 2009.

[5] K. Bowers, A. Juels, and A. Oprea, “Proofs of Retrievability: Theory and Implementation,” Proc. ACM Workshop Cloud Computing Security (CCSW ’09), 2009.

[6] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, “Remote Data Checking for Network Coding-Based Distributed Storage Systems,” Proc. ACM Workshop Cloud Computing Security (CCSW ’10), 2010.

[7] H.C.H. Chen and P.P.C. Lee, “Enabling Data Integrity Protection in Regenerating-Coding-Based Cloud Storage,” Proc. IEEE 31stSymp. Reliable Distributed Systems (SRDS ’12), 2012.

[8] L. Chen, “NIST Special Publication 800-108,” Recommendation for Key Derivation Using Pseudorandom Functions (Revised), http:// csrc.nist.gov/publications/nistpubs/800-108/sp800-108.pdf, Oct. 2009.

[9] R. Curtmola, O. Khan, and R. Burns, “Robust Remote Data Checking,” Proc. ACM Fourth Int’l Workshop Storage Security and Survivability (StorageSS ’08), 2008.

[10] 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), 2008.

[11] A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, “Network Coding for Distributed Storage Systems,” IEEE Trans. Information Theory, vol. 56, no. 9, 4539-4551,Sept. 2010.

IJARBEST
Research at its Best !!!

All Rights Reserved © 2016 IJARBEST