IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 7, September 2014.

www.ijiset.com

ISSN 2348 – 7968

Integrity Verification from single to Multi-Cloud Storage using CPDP with HMAC Function

Mr. Amarjeet Kurmi[1], Mr. Ajay Lala[2]

[1] Department of computer science & Engineering
Gyan Ganga Institute of Technology & science
Madhya Pradesh India

[2]HOD, Department of Information Technology
Gyan Ganga Institute of Technology & science
Madhya Pradesh India

Abstract

Provable data possession (PDP) is a technique for ensuring the integrity of data in storage outsourcing. In this paper, we address the construction of an efficient PDP scheme for distributed cloud storage to support the scalability of service and data migration, in which we consider the existence of multiple cloud service providers to cooperatively store and maintain the clients' data. We present a cooperative PDP (CPDP) scheme based on homomorphic verifiable response and hash index hierarchy. We prove the security of our scheme based on multi-prover zero-knowledge proof system, which can satisfy completeness, knowledge soundness, and zero-knowledge properties. In addition, we articulate performance optimization mechanisms for our scheme, and in particular present an efficient method for selecting optimal parameter values to minimize the computation costs of clients and storage service providers. Our experiments show that our solution introduces lower computation and communication overheads in comparison with non-cooperative approaches.

Keywords: *Scalability, Data Migration, Homomorphic, Multi-Prover.*

1. Introduction

1.1 Ensuring Data Storage Security In Cloud Computing

Cloud Computing has been envisioned as the next generation architecture of IT Enterprise. In contrast to traditional solutions, where the IT services are under proper physical, logical and personnel controls, Cloud Computing moves the application software and databases to the large data centers, where the management of the data and services may not be fully trustworthy. In this article, we focus on cloud data storage security, which has always been an important aspect of quality of service. To ensure the correctness of users' data in the cloud, we propose an effective and flexible distributed scheme with two salient features, opposing to its predecessors.

By Features, opposing to its predecessors. By utilizing the homomorphism token with distributed verification of erasure-coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server(s). Unlike most prior works, the new scheme further supports secure and efficient dynamic operations on data blocks, including: data update, delete and append. Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

Disadvantages:-

1. There is no feature of automatic blocking the cloud server attackers.
2. Less Security – No cryptographic technique is used on the cloud data

1.2 Privacy-Preserving Audit And Extraction Of Digital Contents

A growing number of online services, such as Google, Yahoo!, and Amazon, are starting to charge users for their storage. Customers often use these services to store valuable data such as email, family photos and videos, and disk backups. Today, a customer must entirely trust such external services to maintain the integrity of hosted data and return it intact. Unfortunately, no service is infallible. To make storage services accountable for data loss, we present protocols that allow a third-party auditor to periodically verify the data stored by a service and assist in returning the data intact to the customer. Most importantly, our protocols are privacy-preserving, in that they never reveal the data contents to the auditor. Our solution removes the burden of verification from the customer, alleviates both the customer's and storage service's fear of data .

ISSN 2348 – 7968

1.3 Provable Data Possession At Untrusted Stores

We introduce a model for provable data possession (PDP) that allows a client that has stored data at an un trusted 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 systems. 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 op- posed to linear in the size of the data. Experiments using our implementation verify the practicality of PDP and re- veal that the performance of PDP is bounded by disk I/O and not by cryptographic computation.

Disadvantages:-

1. There is no feature of automatic blocking the cloud server attackers.
2. Owner data will be stored in untrusted cloud servers.

2. Existing System

There exist various tools and technologies for multi cloud, such as Platform VM Orchestrator, VMware vSphere, and Overt. These tools help cloud providers construct a distributed cloud storage platform for managing clients' data. However, if such an important platform is vulnerable to security attacks, it would bring irretrievable losses to the clients. For example, the confidential data in an enterprise may be illegally accessed through a remote interface provided by a multi-cloud, or relevant data and archives may be lost or tampered with when they are stored into an· uncertain storage pool outside the enterprise. Therefore, it is indispensable for cloud service providers to provide security techniques for managing their storage services.

Disadvantages

1. There is no feature of automatic blocking the cloud server attackers.
2. Less Security – No cryptographic technique is used on the cloud data
3. The data integrity is proving only based on the filename and not on the public key or any other key.
4. The attackers details are not dynamic instead its maintaining the log files to store the attacker details and viewing using data mining concepts which is time consuming job and less security.
5. There is no feature of automatic blocking the cloud server attackers.
6. Owner data will be stored in untrusted cloud servers.

3. Proposed System

To check the availability and integrity of outsourced data in cloud storages, researchers have proposed two basic approaches called Provable Data Possession and Proofs of Irretrievability. Ateniese et al. first proposed the PDP model for ensuring possession of files on untrusted storages and provided an RSA-based scheme for a static case that achieves the communication cost. They also proposed a publicly verifiable version, which allows anyone, not just the owner, to challenge the server for data possession..They proposed a lightweight PDP scheme based on cryptographic hash function and symmetric key encryption, but the servers can deceive the owners by using previous metadata or responses due to the lack of randomness in the challenges. The numbers of updates and challenges are limited and fixed in advance and users cannot perform block insertions anywhere.

4. Project Aim

To compute the cloud securely monitoring by Third Party Authority and achieving the data integrity, batch auditing

5. System Requirement Specifications

This Chapter describes about the requirements. It specifies the hardware and software requirements that are required in order to run the application properly. The Software Requirement Specification (SRS) is explained in detail, which includes overview of this dissertation as well as the functional and non-functional requirement of this dissertation.

SRS for Cooperative Provable

Functional	Control the file access at cs1, cs2, cs3, cs4, Data Integrity Proof at TPA. File Management
Non- Functional	Cloud servers never monitors and controls the TPA,Remote user never uploads the file
External interface	LAN , Routers
Performance	Maintaining the File Access between the remote user and the cs1, cs2, cs3, cs4, Finds the hackers in the cloud
Attributes	File Management, File req, res Services, Maintain CS1, CS2, CS3, CS4

Table: 5.1 Summaries of SRS

ISSN 2348 – 7968

6. Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:-

a. The Owner will divide the file into 'N' number of blocks and has to upload the specified cloud server (Cs1, Cs2, Cs3, and Cs4).
b. The Cloud server has to authorize the valid remote users. If the Remote user is hacker then he has to block in the cloud server. The data should be integrated by the cloud server.
c. The Third party auditor has to maintain the error localization and has to monitor the Cloud Server Activities.
d. The Remote user has to user correct Secret key and file name. If anyone is wrong then he is detected as attacker.
e. The Attributes are File Management, TPA, cloud server, owner, Remote user and blocked user.

7. System Configuration

7.1 H/W system configuration:-

Processor	-	Pentium IV
Speed	-	1.1 GHz
RAM	-	256 MB (min)
Hard Disk	-	20 GB
Floppy Drive	-	1.44 MB
Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	SVGA

7.2 S/W system configuration:-

Operating System	:	Windows95/98/2000/XP
Application Server	:	Tomcat5.0/6.X
Front End	:	HTML,Java (AWT,SWING,RMI,Networking)
Scripts	:	JavaScript.
Server side Script	:	Java
Database	:	MS Access / Mysql
Database Connectivity	:	JDBC

8. Module Description:

8.1 Multi cloud storage:

Distributed computing is used to refer to any large collaboration in which many individual personal computer owners allow some of their computer's processing time to be put at the service of a large problem. In our system the each cloud admin consist of data blocks. Cloud computing environment is constructed based on open architectures and interfaces, it has the capability to incorporate multiple internal and/or external cloud services.

8.2 Cooperative PDP:

Cooperative PDP (CPDP) schemes adopting zero-knowledge property and three-layered index hierarchy, respectively. In particular efficient method for selecting the optimal number of sectors in each block to minimize the computation costs of clients and storage service providers. Cooperative PDP (CPDP) scheme without compromising data privacy based on modern cryptographic techniques

8.3 Data integrity:

Data Integrity is very important in database operations in particular and Data warehousing and Business intelligence in general. Because Data Integrity ensured that data is of high quality, correct, consistent and accessible.

9. Third Party Auditor:

Trusted Third Party (TTP) who is trusted to store verification parameters and offer public query services for these parameters. In our system the Trusted Third Party, view the user data blocks and uploaded to the distributed cloud. In distributed cloud environment each cloud has user data blocks. If any modification tried by cloud owner a alert is send to the Trusted Third Party.

10. Cloud User:

The Cloud User who has a large amount of data to be stored in multiple clouds and have the permissions to access and manipulate stored data. The User's Data is converted into data blocks. The data blocks are uploaded to the cloud. The TPA views the data blocks and Uploaded in multi cloud. The user can update the uploaded data. If the user wants to download their files, the data's in multi cloud is integrated and downloaded.

11. Summary

We presented the construction of an efficient PDP scheme for distributed cloud storage. Based on homomorphic verifiable response and hash index hierarchy, we have proposed a cooperative PDP scheme to support dynamic scalability on multiple storage servers. We also showed that our scheme provided all security properties required by zero knowledge interactive proof system, so that it can resist various attacks even if it is deployed as a public audit service in clouds.

Furthermore, we optimized the probabilistic query and periodic verification to improve the audit performance. Our experiments clearly demonstrated that our approaches only introduce a small amount of computation and communication overheads. Therefore, our solution can be treated as a new candidate for data integrity verification in outsourcing data storage systems. As part of future work, we would extend our work to explore more effective CPDP constructions. Finally, it is still a challenging problem for the generation of tags with the length irrelevant to the size of data blocks. We would explore such a issue to provide the support of variable-length block verification.

12. Conclusion

In this paper, we presented the construction of an efficient PDP scheme for distributed cloud storage. Based on homomorphic verifiable response and hash index hierarchy, we have proposed a cooperative PDP scheme to support dynamic scalability on multiple storage servers. We also showed that our scheme provided all security properties required by zero knowledge
Interactive proof system, so that it can resist various attacks even if it is deployed as a public audit service in clouds. Furthermore, we optimized the probabilistic query and periodic verification to improve the audit performance. Our experiments clearly demonstrated that our approaches only introduce a small amount of computation and communication overheads. Therefore, our solution can be treated as a new candidate for data integrity verification in outsourcing data storage systems.

a. Future Enhancement 1
Generating Mobile Alerts on File Attackers on the Cloud
b. Future Enhancement 2
Generating Meta data for TPA, Public Auditing, blocking and UN blocking and Applying ECC Algorithm for data encryption and decryption.

13. . Reference

[1]YanZhu,Hongxin Hu,Gail-Joon Ahn,Senior Member,IEEE,Mengyang Yu "Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage" IEEE Transactions on parallel and distributed systems.10.1109/TPDS.2012.66,PP 1-13.

[2]Y. Zhu, H. Hu, G.-J. Ahn, Y. Han, and S. Chen, "Collaborative integrity verification in hybrid clouds," in IEEE Conference on the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing, CollaborateCom, Orlando, Florida, USA, October 15-18, 2011, pp. 197–206

[3]Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Dynamic audit services for integrity verification of outsourced storages in clouds," in SAC, W. C. Chu, W. E. Wong, M. J. Palakal, and C.-C. Hung, Eds. ACM, 2011, pp. 1550–1557.

[4]C. C. Erway, A. K¨upc¸ ¨u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in ACM Conference on Computer and Communications Security, E. Al-Shaer, S. Jha, and A. D. Keromytis, Eds. ACM, 2009, pp. 213–222.

Mr. Amarjeet Kurmi- I completed Bachelor of Engineering with Information Technology branch from GRKIST Jabalpur M.P. (India) and perusing Master of Technology with Computer Science & Engineering From Gyan Ganga Institute of Science & Technology Jabalpur M.P. India.

Mr. Ajay Lala- He has completed B.E. & M.Tech previously from reputed institute & has around 15 years of teaching experience in various academic institutes. He is currently working as a Head of The Department in GGITS & Guided many Research papers.