International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Public Auditing System with Auto-Data Recovery System on Cloud Scheme

Dhane Sachin V.[1], Joshi P.[2]

[1]Master of Computer Engg, Savitribai Phule Pune University, G. H. Raisoni Collage of Engg and Techonology, Wagholi, Pune

[2]HOD of Information Technology, Savitribai Phule Pune University, G. H .Raisoni Collage of Engg and Techonology, Wagholi, Pune

Abstract: ***Now a day's storage and sharing with the cloud computing is becoming very popular. But always security questions raised when we store the data onto the cloud services, because in most of the cases we generally consume the cloud services offers by various provides. To protect our data in cloud against corruptions, cross verification of the data integrity becomes very critical. We have build the framework which will perform auditing as well as regenerating of the data in case of corruption. In this paper, we propose a public auditing scheme for the regenerating-code-based cloud storage. Currently we have system which will regenerate the data when use is online. User has to perform that operation manually and developing the framework which will able to perform the operation automatically, It works offline as well.***

Keywords: Cloud Computing, Public Auditing Framework, Data Reconstruction, Online/Offline mode.

1. Introduction

With data storage and sharing services, such as Google Drive, provided by the cloud, people can easily work together as a group by sharing data with each other. More specifically, once a user creates shared data in the cloud, every user in the group is able to not only access and modify shared data, but also share the latest version of the shared data with the rest of the group. Although cloud providers promise a more secure and reliable environment to the users, the integrity of data in the cloud may still be compromised, due to the existence of hardware/software failures and human error.

To protect the integrity of data in an un-trusted cloud, a number of mechanisms have been proposed. In these mechanisms, a signature is attached to each block in data, and the integrity of data relies on the correctness of these signatures. One of the most significant and common features of these mechanisms is their ability to allow not only the data owner, but also a public verifier, such as a third party auditor (TPA), to check data integrity in the cloud without downloading the entire data, referred to as public auditing.

We focus on the integrity verification problem in regenerating-code-based cloud storage, especially with the functional repair strategy. To repair our data in cloud we will develop distributed data base scheme also know as database cluster. We are also introducing the proxy mechanism which will have use keys for the regenerating the data in case of the corruption. It will also have the ability to new cluster schema.

2. Related Work

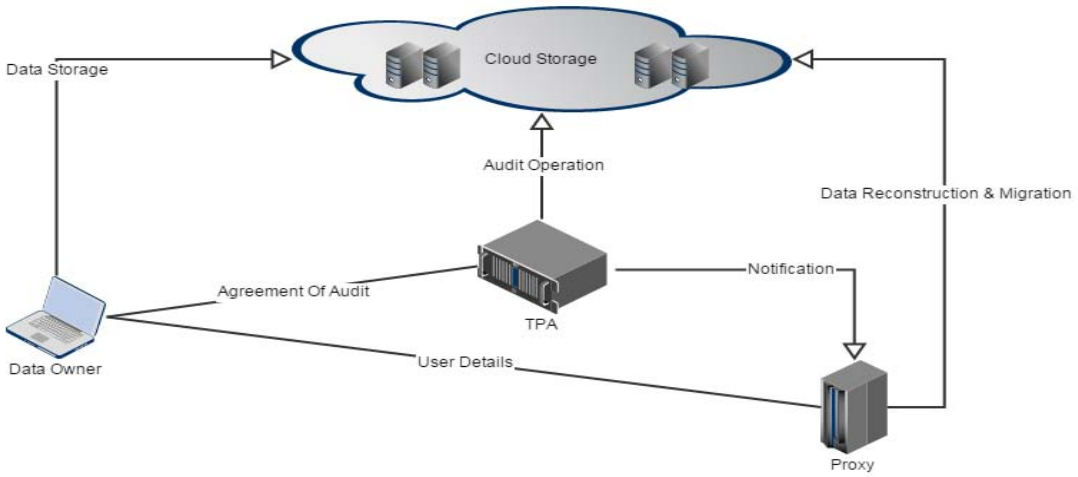
Various ways/approaches have been used by many researchers in field related data security on cloud. Approaches like protection of data privacy against auditor by combining cryptography method with bilinear property, also using mask techniques. Approaches are further extended to support data protection on mutlicloud system. Researchers kan yang ,Xiaohua Jia did research to provide data security with less communication cost ,less computation by checking all possibilities of moving data load from auditor to server.

To support the dynamic auditing, Ateniese et al. [1] developed a dynamic provable data possession protocol based on cryptographic hash function and symmetric key encryption. Their idea is to precompute a certain number of metadata during the setup period, so that the number of updates and challenges is limited and fixed beforehand. In their protocol, each update operation requires recreating all the remaining metadata, which is problematic for large files. Moreover, their protocol cannot perform block insertions anywhere (only append-type insertions are allowed). Erway et al. [2] also extended the PDP model to support dynamic updates on the stored data and proposed two dynamic provable data possession scheme by using a new version of authenticated dictionaries based on rank information. However, their schemes may cause heavy computation burden to the server because they relied on the PDP scheme proposed by Ateniese. In [3], the authors proposed a dynamic auditing protocol that can support the dynamic operations of the data on the cloud servers, but this method may leak the data content to the auditor because it requires the server to send the linear combinations of data blocks to the auditor. In [4], the authors extended their dynamic auditing scheme to be privacy preserving and support the batch auditing for multiple owners. However, due to the large number of data tags, their auditing protocols will incur a heavy storage overhead on the server. In [5], Zhu et al. proposed a cooperative provable data possession scheme that can support the batch auditing for multiple clouds and also extend it to support the dynamic auditing in [26]. However, it is impossible for their scheme to support the batch auditing for multiple owners. That is because parameters for generating the data tags used by each owner are different, and thus, they cannot combine the data tags from multiple owners to conduct the batch auditing. Another drawback is that their scheme requires an additional trusted organizer to send a commitment to the auditor during the

batch auditing for multiple clouds, because their scheme applies the mask technique to ensure the data privacy. However, such additional organizer is not practical in cloud storage systems. Furthermore, both Wang's schemes and Zhu's schemes incur heavy computation cost of the auditor, which makes the auditing system inefficient.

3. Proposed System

3.1 System Flow

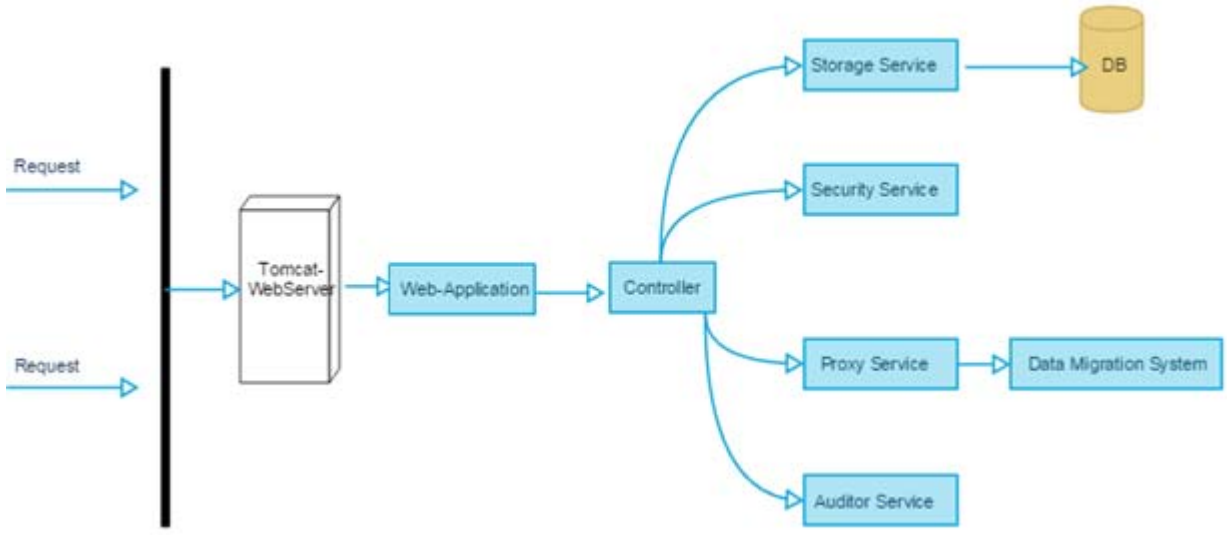


The cloud offers data storage and sharing services to users. The TPA is able to publicly audit the integrity of shared data in the cloud for users. In a group, there is one original user and a number of group users. The original user is the original owner of data. This original user creates and shares data with other users in the group through the cloud. Both the original user and group users are able to access, download and modify shared data. Shared data is further divided into a number of blocks. A user can modify a block in shared data by performing an insert, delete or update operation on the block.

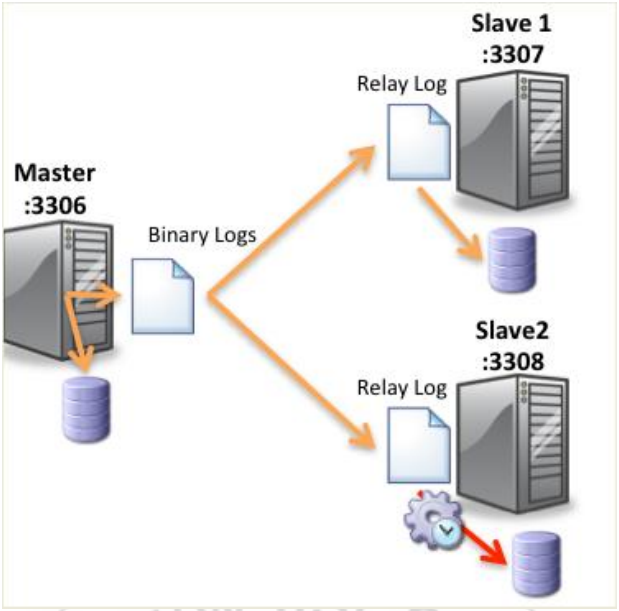
TPA is trusted and its audit result is unbiased for both data owners and cloud servers; and a proxy agent, who is semi-trusted and acts on behalf of the data owner to regenerate authenticators and data blocks on the failed servers during the repair procedure. Notice that the data owner is restricted in computational and storage resources compared to other entities and may becomes off-line even after the data upload procedure. The proxy, who would always be online, is supposed to be much more powerful than the data owner but less than the cloud servers in terms of computation and memory capacity. To save resources as well as the online burden potentially brought by the periodic auditing and accidental repairing, the data owners resort to the TPA for integrity verification and delegate the reparation to the proxy.

4. Proposed System Diagram

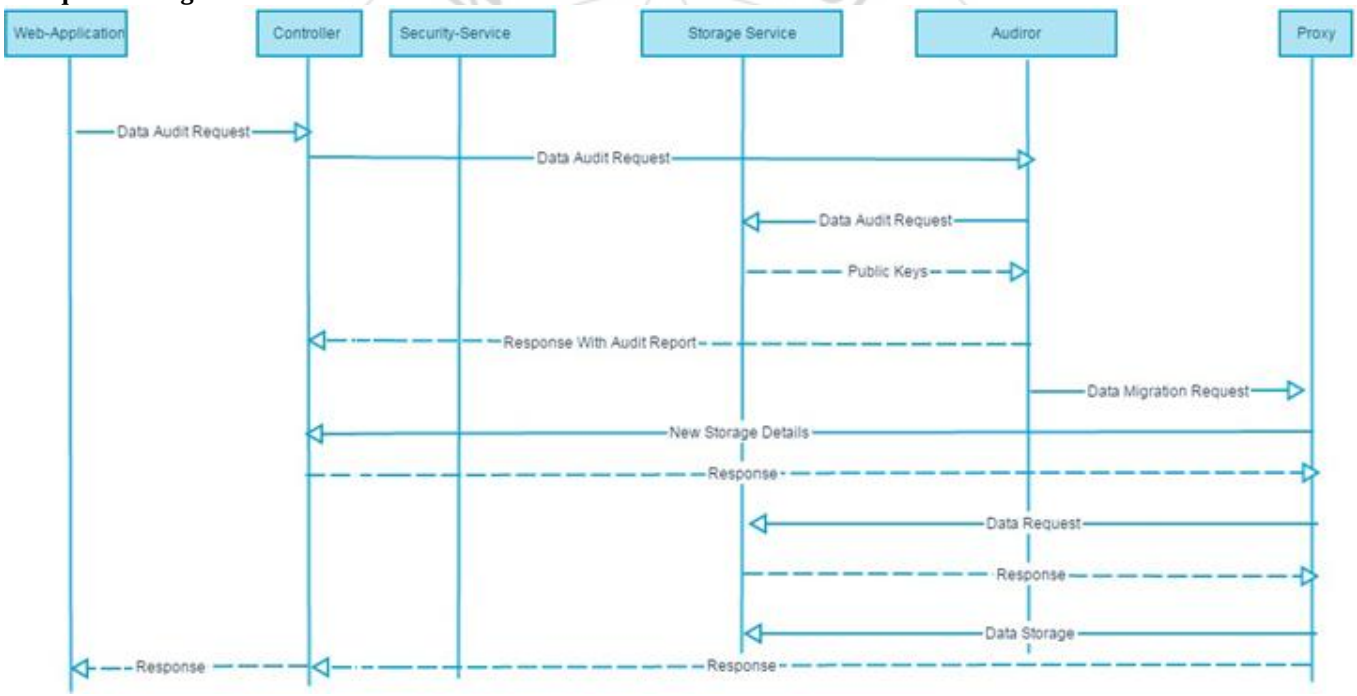
1. System Diagram



2. Database Cluster



3. Sequence Diagram



This system includes four models / entity.

- **Database-Cluster**

Database-Cluster will store user data in distributed environment also keep MSD for the same.Here MSSQL cluster is used . MySQL Cluster is a technology providing shared-nothing clustering and auto-sharding for the MySQL database management system. It is designed to provide high availability and high throughput with low latency, while allowing for near linear scalability.

MySQL Cluster provides you with the following benefits:-

In-Memory Database Delivering 200 Million QPS

Using memory-optimized tables, MySQL Cluster provides real-time response time and throughput meet the needs of the most demanding web, telecommunications and enterprise applications - delivering 200 Million Queries Per Second.

Auto-sharding for Write-scalability

MySQL Cluster automatically shards (partitions) tables across nodes, enabling databases to scale horizontally on low cost, commodity hardware while maintaining complete application transparency.

99.999% Availability

With its distributed, shared-nothing architecture, MySQL Cluster has been designed to deliver 99.999% availability ensuring resilience to failures and the ability to perform scheduled maintenance without downtime.

SQL & NoSQL APIs
MySQL Cluster enables users to blend the best of both relational and NoSQL technologies into solutions that reduce cost, risk and complexity.

Multi-site Clusters with Active Active Geographical Replication
Update-anywhere geographic replication enables multiple clusters to be distributed geographically for disaster recovery and the scalability of global web services.

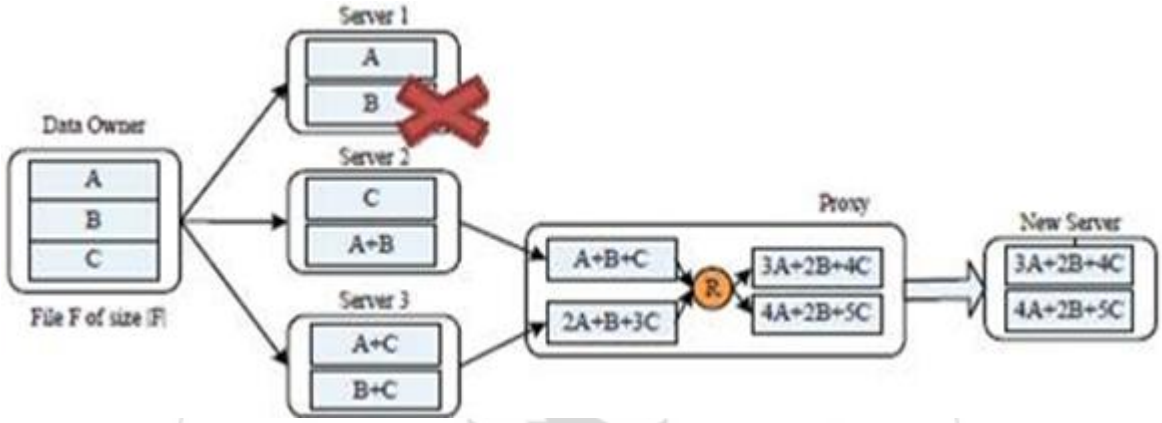
Online Scaling & Schema Upgrades
To support continuous operation, MySQL Cluster allows on-line addition of nodes and updates to live database schema to support rapidly evolving and highly dynamic workloads.

MySQL Cluster Auto-Installer
Get MySQL Cluster up and running in minutes! Graphically configure and provision a production-grade cluster, automatically tuned for your workload and environment.

MySQL Cluster Manager
MySQL Cluster Manager simplifies the creation and management of the MySQL Cluster Carrier Grade Edition database by automating common management tasks.

- **Proxy**

This will act on behalf of the data owner, this module is very critical while doing data migration



User wants to upload onto the cloud storage system. As you can see in the system there are 3 storage servers.
Server 1 :- A , B
Server 2 :- C , A + B
Server 3 :- A + C , B + C
* A , B , C , stands for the file , A + B stands for File A and File B. Now what happen in case your auditor will detected problems with the server 1 , which holds the data A & B files .Proxy program will come into the action , it will grab the data from server-2 and server-3 , proxy will linearly combines copies from the server 2 and server 3. After regeneration of the data, it will send to the healthy server.

- **Data Owner**

Owner of the data. Data Owner can upload file to Cloud Storage Server. Cloud Storage server is distributed one , which has MDS , metadata storage server which keeps track of all the copies and location where data is present. Data Owner share security key with Third Party Auditor for the auditing of the record. Data Owner share security key / signature with the Proxy server. Public Auditor has capability to integrate the Proxy , so it can directly send the audit report to the proxy server. Proxy server perform the analysis of the auditor report ,if the audit report fails then it will perform linear transformation operation onto data which is not corrupted.

- **TPA (Trusted third party auditor)**

Public Auditor is nothing a program or services which run continuously on background which verifies data integrity . i.e It will cross verify whatever data that has been uploaded by the client is same as data copy present on cloud server. In older system Data Owner need to send request to auditor to verify data, In our case we will write it as services or background process. It will also have result and notification feature (*Email notification*)

5. Result

1. Auditing Graph
2. Time required to perform the auditing.
3. History of all the auditing system.

6. Conclusion And Future Scope

In this paper, we propose a public auditing scheme for there regenerating-code-based cloud storage system, where the data owners are privileged to delegate TPA for their data validity checking. To protect the original data privacy against theTPA, We randomize the coefficients in the beginning rather than applying the blind technique during the auditing process. Considering that the data owner cannot always stay online in practice, in order to keep the storage available and verifiable after a malicious corruption, we introduce a semi-trusted proxy into the system model and provide a privilege for the proxy to handle the reparation of the coded blocks and authenticators. To better appropriate for the regenerating-code-scenario, we design our authenticator based on the BLS signature. This authenticator can be efficiently generated by the data owner simultaneously with the encoding procedure. Extensive analysis shows that our scheme is provable secure, and the performance evaluation shows that our scheme is highly

efficient and can be feasibly integrated into a regenerating-code-based cloudstorage system.

In future it is also possible
To build this system on the Hybrid Cloud Platform, MDS will be on Amazon Cloud / Google Cloud Computing Platform and rest of the system on another cloud server. Need to add dynamic feature to create a new Healthy database, automatically. It is also possible add Third Party Security Service to secure our data from the data-owner.

References

[1] Scalable and Efficient Provable Data Possession G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik.
[2] Dynamic Provable Data Possession C.C. Erway, A. Ku¨ pc¸u¨ , C. Papamanthou, and R. Tamassia,
[3] Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li
[4] Public Auditing for Data Storage Security in Cloud Computing C. Wang, Q. Wang, K. Ren, and W. Lou
[5] Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage Y. Zhu, H. Hu, G. Ahn, and M. Yu
[6] Dynamic Audit Services for Integrity Verification of Outsourced Storages in Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S.
[7] Demonstrating Data Possession and Untreatable Data Transfer, D.L.G. Filho and P.S.L.M. Barreto
[8] Fast Integrity for Large Data G. Yamamoto, S. Oda, and K. Aoki,
[9] Auditing to Keep Online Storage Services Honest,M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan
[10] Toward Publicly Auditable Secure Cloud Data Storage Services C. Wang, K. Ren, W. Lou, and J. Li

Author Profile

Sachin Vasant Dhane is a graduated from Pune Institute of Computer Technology in the year 2004 . Worked as Sofware Engineer well known in IT industries . Now persuing M.E. Computer Engineering from G. H .Raisoni Collage of Engg anTechonology, Wagholi, Pune.