Secure Storage Services and Erasure Code Implementation in Cloud Servers

G. Rahul Reddy[1], N. J. Subashini[2]

1 Research Scholar, M.Tech, Cloud Computing, Dept of Information Technology, SRM University, Chennai, India,

2 Assistant Professor, Dept of Information Technology, SRM University, Chennai, India

Abstract

Cloud storage allows users to store their data on cloud servers. User's can save their money and time as they do not maintain any local database and hardware to store the data. Security and privacy are main concerns for the data that has been stored in the cloud. A wide variety of encryption techniques and algorithms were used for protecting data that has been remotely stored. In order to address this security and privacy problems, we propose double encryption strategy and also providing back-up for the data that has been stored in cloud. We use HMAC scheme for encryption of the data. As with any MAC, it is used to simultaneously verify both the data integrity and the authentication of a message. Erasure coding is used for rebuilding lost encoded fragments from existing encoded fragments.

Keywords- *Data integrity, Keying Hash Functions for Message Authentication, Cloud Computing.*

1. Introduction

Cloud computing [1] promises to cut operational and capital costs and, more importantly, let IT departments focus on strategic projects instead of keeping the datacenter running. The datacenter is the collection of servers where the application to which you subscribe is housed. It could be a large room in the basement of your building or a room full of servers on the other side of the world that you access via the Internet. A growing trend in the IT world is virtualizing servers. That is, software can be installed allowing multiple instances of virtual servers to be used. In this way, you can have half a dozen virtual servers running on one physical server.

Software as a Service (SaaS) [2] is the model in which an application is hosted as a service to customers who access it via the Internet. When the software is hosted off-site, the customer doesn't have to maintain it or support it. On the other hand, it is out of the customer's hands when the hosting service decided to change it.. SaaS applications differ from earlier distributed computing solutions in that SaaS was developed specifically to use web tools, like the browser. This makes them web-native. It was also built with a multitenant back end in mind, which enables multiple customers to use an application. SaaS provides network-based access to commercially available software. Since the software is managed at a central location, customers can access their applications wherever they have web access.

Platform as a Service (PaaS) [3] is another application delivery model. PaaS supplies all the resources required to build applications and services completely from the Internet, without having to download or install software. PaaS services include application design, development, testing, deployment, and hosting. Other services include team collaboration, web service integration, database integration, security, scalability, storage, state management, and versioning.

Infrastructure as a Service (IaaS) [4] consists of hardware infrastructure that is located in the cloud. It includes cloud storage, cloud servers and cloud networks, and is also known as Hardware as a Service (HaaS).The infrastructure can be used to run software or simply to store data. The consumers can be end-users, developers or other cloud providers.

Objectives

- Providing Data Security.
- Providing Data Privacy.
- Preventing Server Colluding Attacks.
- Implementing Erasure Coding.

2. Modules

2.1. Data Owner

User is the person is going to see or download the data from the Cloud server. To access the data from the Cloud server, the users have to be registered with the cloud server. So that the user have to register their details like username, password and a set of random numbers. This is information will stored in the database for the future authentication.

Data owner is the Person who is going to upload the data in the Cloud Server. To upload the data into the Cloud server, the data owner have be registered in the cloud server. Once the data owner registered in cloud server, the space will be allotted to the data owner.

2.2. Main Cloud Server

Cloud Server is the area where the user going to request the data and also the data owner will upload their data. Once the user send the request regarding the data they want, the request will first send to the cloud server and the cloud server will forward your request to the data owner. The data owner will send the data the data the user via cloud server. The cloud server will also maintain the data owner and users information in their database for future purpose.

2.3. Data Splitting and Encryption

In this module, once the data was uploaded into the cloud server, the cloud server will split the data into many parts and store all the data in the separate data servers. In techniques wasn't used in proposed system so that there might be a chance of hacking the entire data. Avoid the hacking process, we splitting the data and store those data in corresponding data server. We're also encrypting the data segments before storing into the data server.

2.4. KEY SERVER

The encryption keys are stored in appropriate key servers. So that we can increase the security of the cloud network. If the user wants retrieve the data, they've to provide all the key that are stored in the appropriate key servers.

2.5. Party Bit Addition And Erasure Code

Once the data are stored in the corresponding data servers and the keys are stored in the key servers. Then we're adding the parity bits to the data, so that the data will be changed. Also we're applying the erasure code by using the XOR operation, while XORing the block data, the data will be converted in binary data.

2.6. Trusted Party Auditor

Once added the parity added bits, then the data will be given to the Trusted Parity auditor. The Trusted Parity Auditor will generate the signature using change and response method. The data will be audited in this module, if any changes occur it will provide the intimation regarding the changes.

2.7. REPLICA SERVER

We'll maintain the separate Replica Cloud server. If suppose the data in the data server was lost, then the Main Cloud server will contact the Replica Cloud server and get the data from the Replica Cloud Server. By using this concept, we can get the data if any data loss occurs.

3. SYSTEM ARCHITECTURE

Cloud computing consist of two components the front end and the back end. The front end of the cloud computing system comprises the client's device and some applications that are needed for accessing the cloud computing system. Back end refers to the cloud itself which encompass various computers, data storage systems and servers. The whole system is administrated via a central server that is also used for monitoring client's demand and traffic ensuring smooth function of the system. Cloud computing systems also will have a copy of all its client's data to restore the service which may arise due to device breakdown.

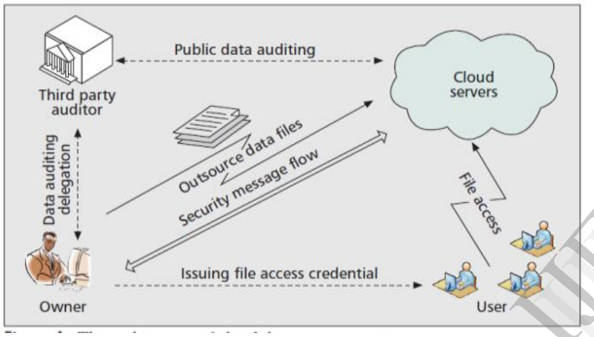

Fig.1.Cloud Storage Service Architecture.

Recently, the importance of ensuring the remote data integrity has been highlighted by the following research works under different system and security models [12], [13], [14], [15],[16]. 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 [23], but does not guarantee the data availability in case of server failures. 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. To ensure security we go for double encryption one at client side during file upload and secondly at time of file distribution. 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.

4. Ensuring Cloud Data Storage

In cloud data storage system, users store their data in the cloud and no longer possess the data locally. Thus, the correctness and availability of the data files being stored on the distributed cloud servers must be guaranteed. One of key issues is to effectively detect any unauthorized data modification and corruption, possibly due to server compromise and/or random Byzantine failures.

4.1. File Distribution Preparation

It is well known that erasure-correcting code [5],[6] may be used to tolerate multiple failures in distributed storage systems. In cloud data storage, we rely on this technique to disperse the data file F redundantly across a set of n= m + k distributed servers. An (m,k)Reed-Solomon erasure-correcting code is used to create k redundancy parity vectors from m data vectors in such a way that the original m data vectors can be reconstructed from any m out of the m þ k data and parity vectors. By placing each of the m þ k vectors on a different server, the original data file can survive the failure of any k of the m þ k servers without any data loss, with a space overhead of k=m. For support of efficient sequential I/O to the original file, our file layout is systematic, i.e., the unmodified m data file vectors together with k parity vectors is distributed across m þ k different servers.

4.2. Challenge Token Precomputation

For assurance of data storage correctness and data error localization simultaneously, our scheme entirely relies on the precomputed verification tokens. The token computation function we are considering belongs to a family of universal hash function [7].

Algorithm 1. Token Precomputation.

```
1: Choose parameters l; n and function f,$^{\phi}$;
2: Choose the number t of tokens;
3: Choose the number r of indices per verification;
4: Generate master key $K_{PRP}$ and challenge key $k_{chal}$;
5: for vector$G^{j}$ ; j ←1; n do
6: for round i← 1, t do
7: Derive  $\alpha_i = f_{kchal}(i)$ and $k_{prp}{}^{(i)}$ from  $K_{prp}$
```

```
Compute $v_i^{(j)}$
end for
end for
Store all the vi's locally
```

4.3. File Retrieval and Error Recovery

Since our layout of file matrix is systematic, the user can reconstruct the original file by downloading the data vectors from the first m servers, assuming that they return the correct response values.

Algorithm 2. Error Recovery.

```
procedure
% Assume the block corruptions have been detected among
% the specified r rows;
% Assume s≤k servers have been identified misbehaving
Download r rows of blocks from servers;
Treat s servers as erasures and recover the blocks.
Resend the recovered blocks to corresponding servers.
end procedure
```

4.4. Toward Third Party Auditing

As discussed in our architecture, in case the user does not have the time, feasibility, or resources to perform the storage correctness verification, he can optionally delegate this task to an independent third-party auditor, making the cloud storage publicly verifiable. However, as pointed out by the recent work [8],[9] to securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities toward user data privacy. Namely, TPA should not learn user's data content through the delegated data auditing. Now we show that with only slight modification, distributed protocol [10],[11], can support privacy-preserving third party auditing.

5. 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. To ensure security we adopt double encryption strategy one at client side during file upload and secondly at time of file distribution. 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.

6. REFERENCES

[1] R. Buyya, J. Broberg, and A. Goscinski (eds). Cloud Computing: Principles and Paradigms. ISBN13: 978-0470887998, Wiley Press, USA, February 2011

[2] Godse, M. ; Shailesh J Mehta Sch. of Manage., Indian Inst. of Technol. Bombay, Mumbai, India ; Mulik, S.

[3] J. Lakshmi and Sathish S. Vadhiyar Supercomputer Education and Research Centre Indian Institute of Science, Bangalore 560 012

[4] Jayasinghe, D. CERCS, Georgia Inst. of Technol., Atlanta, GA, USA Pu, C. ; Eilam, T. ; Steinder, M. ; Whally, I. ; Snible, E.

[5] 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.

[6] J. Hendricks, G. Ganger, and M. Reiter, "Verifying Distributed Erasure-Coded Data," Proc. 26th ACM Symp. Principles of Distributed Computing, pp. 139-146, 2007.

[7] L. Carter and M. Wegman, "Universal Hash Functions," J.Computer and System Sciences, vol. 18, no. 2, pp. 143-154, 1979.

[8] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. IEEE INFOCOM, Mar. 2010.

[9] C. Wang, K. Ren, W. Lou, and J. Li, "Towards Publicly Auditable Secure Cloud Data Storage Services," IEEE Network Magazine, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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. Networks (SecureComm '08), pp. 1-10,2008.

[16] 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.