Multiple TPA for Extensive Reliability and Security in Cloud Storage

Swati khambete
D Y Patil COE, Akurdi,Savitribai Phule
Pune University,
Pune, India.
swatikhambete11@gmail.com

Mr.Abhijit Patankar
D Y Patil COE, Akurdi,Savitribai Phule
Pune University,
Pune, India.
abhijitpatankar@yahoo.com

Abstract— This paper focuses on the security and integrity of data hold on cloud data servers. The data integrity verification is finished by employing a third party auditor who is authorized to check integrity of data sporadically on behalf of client. The way application software and databases are stored has been modified. Currently they are stored in cloud data centers in which security is an apprehension from client point of view. The new development which is used to store and manage data without principal investment has brought many security challenges which are not thoroughly understood. Customer data from third party auditor notices of when data integrity is lost. The proposed system not only supports data integrity verification but also supports data mobility. Prior work has been done in this online data mobility and there is lack of true public auditability. Auditor task is to monitor data modifications like insert and remove. The proposed system is able to support both public auditability and data mobility. Problems with existing systems literature review has revealed that the motivation behind this work take up. Merkle hash tree block-level authentication is used to improve. Auditing functions to handle together Bilinear transform overall signature is used. The TPA for multiple clients concurrently enables audit. So here we evaluate the TPA based multi-user system. Experiments show that the proposed system is also very efficient and safe. For proposed work we are using multiple TPA's and multiple clients.

Keywords- *Data storage, public auditability, Data integrity, Data dynamics, Cloud computing, Cloud storage.*

*****

I. Introduction

Several trends are opening the era of cloud computing with an Internet based development and vast use of computer technology. Ever cheaper and more powerful Processor, together with "software-as-a-service” computing architecture, data centers are Pool of a huge scale computing service. Meanwhile, increase in network bandwidth and reliable yet flexible network connections make it possible for client to subscribe the high quality software services that only live on the remote data centers. Although a promising service for platform, the new Internet "cloud" data storage brings about many challenging design issues. A profound impact is on the security and performance of the overall system. Biggest concern with the data Storage in cloud is data integrity verification that untrusted server.

What is more serious is that to save money and storage space a service provider deliberately deletes simple customer-related data files which are accessed rarely. Considering the sheer size of the outsourced electronic data and the ability of a client resource, it puts constraint on how periodic integrity verifications to a local copy of the data file can be generalized by client.

Great efforts are made to design the solution to solve the problem of data integrity that plans to meet various requirements: high efficiency, Massive use of stateless verification, queries and retrievability of data etc. Consider the role of verifier in the model, all schemes presented before fall into two main categories: private and public auditability. Although private auditors can achieve higher efficiency but public auditors permits anyone to challenge the cloud server while not accessing any personal information. Clients are able to outsource to TPA (Third party auditor) the data integrity work without engaging their own resources as this would have been an overhead for the client’s resources. Supporting dynamic data operations is one of the major concerns in previous designs. In cloud computing, clients can not only store electronic data remotely but clients can also modify or update the data by block level modification or deletion. Unfortunately main focus in remote data storage is not on dynamic operations but only on static data stored.

II. RELATED WORK

A Usually the user a service supplier or a private company, they can log into the cloud. Cloud computing is based on a client-server jobs. Cloud server-based applications then provide information services as output on the client device.

Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou [7], In this paper Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. It moves the application databases and software to the centralized large data centers, where the management of the data and services may not be fully trustworthy. This unique paradigm contains many new security challenges, which have not been well understood. This paper discusses 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) which on behalf of the cloud client verifies the integrity of the dynamic data stored in the cloud

G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z.Peterson, and D. Song [3], They introduce a model for provable data possession (PDP) that allows a client that has stored data on server which are untrusted to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of control by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client keeps a constant amount of metadata to verify the proof. The response or challenge protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data, checks supports for large data sets in widely-distributed storage systems.

A. Juels and B.S. Kaliski Jr [4], In this paper, they define and explore proofs of retrievability. A POR scheme enables an archive or back-up service to produce a concise proof that a user can retrieve a target file F, that is, that the archive retains and reliably transmits file data sufficient for the user to recover F in its entirety. A POR may be viewed as a kind of cryptographic proof of knowledge, but one specially designed to handle a large file F. They explore POR protocols here in which the communication costs, number of memory accesses for the provider storage requirements are small parameters which are essentially independent of the length of F. In addition to proposing new technique, for constructions, they explore implementation considerations and optimizations that bear on previously explored, related schemes.

H. Shacham and B. Waters [8], In a proof-of-retrievability system, a data storage center must have to prove to a verifier that he is actually storing all of a client's data. The central challenge is to build systems that are both efficient and provably secure. It should be possible to extract the client's data from any provider that passes a verification check. In this paper, it gives the rest proof-of-retrievability schemes with full proofs of scheme which allows public verifiability: anyone can act as a verifier, not just the file owner. Second scheme, which builds on pseudo random functions and is secure in the standard model, allows only private verification. security against capricious adversaries in the strongest model, that of Juels and Kaliski.

K.D. Bowers, A. Juels, and A. Oprea [9], In this paper, they propose a theoretical framework for the design of PORs. This system improves the previously proposed POR constructions of Juels-Kaliski and Shacham-Waters, and also sheds light on the conceptual limitations of previous theoretical models for PORs. It supports a fully Byzantine adversarial model, carrying only the restriction—fundamental to all PORs—that the adversary’ s error rate ⩽ be bounded when the client seeks to extract F. Our techniques support efficient protocols across the full possible range of ⩽, up to ⩽ no negligibly close to 1. They propose a new variation on the Juels-Kaliski protocol and describe a prototype implementation. They reveal practical encoding even for files F whose size exceeds that of client main memory.

III. PROPOSED APPROACH FRAMEWORK AND DESIGN

A. *Architecture*

The system architecture is as shown in Figure 1. The entities in the network are multiple clients, cloud storage server and multiple third party auditors. Clients are individual or organization who depends on cloud service provider for storing data files and maintaining them. Clients can perform dynamic operations on the data stored on the cloud server. The cloud storage server is having lot of storage space and estimated resources. It is conserved by cloud service provider. Third party auditors are trusted and have capabilities of auditing the client’s data on demand. If load on first TPA increases then that TPA uses load balancing to shift the extra load on another TPA. Multiple TPA makes the system more reliable and efficient.

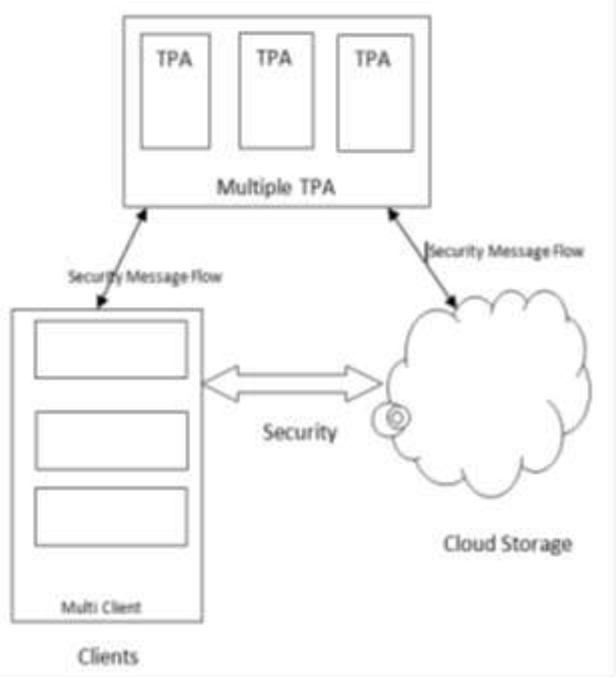


Figure. 1 Architecture Diagram

B. *Propose Work*

Cloud service provider stores cloud storage data in the server which is in control of cloud service provider. This model presumes two things. There are, a) cloud data provider who can delete files. b) Cloud data providers data center could be hiding prospective problems. In keeping with these assumptions, the mechanisms proposed are designed into the system.

The entities in the propose network are multiple client,cloud storage server and multiple third party auditor. If load on TPA increases the system transfers the load to other TPA so that the load can be balanced and system performance can be increased. The cloud data provider may delete files of client knowingly or unknowingly. Cloud data provider may hide prospective problems in the data center. Cloud may manipulate the authentication process in the data dynamics operations. Single TPA has less reliability as compared to multiple TPA.

We propose a general formal PoR model with public verifiability for cloud data storage, in which block less verification is achieved; We equip the proposed PoR construction with the function of supporting for fully dynamic data operations, especially to support block insertion, which is missing in most existing schemes. We prove the security of our proposed construction and justify the performance of our scheme through concrete implementation and comparisons with the state-ofthe-art. We have improved the existing proof of storage models by manipulating the classic Merkle Hash Tree construction for block tag authentication to achieve efficient data dynamics. We further explore the technique of bi-linear aggregate signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive performance analysis and security shows that the proposed scheme is highly efficient and provably secure.

C. Data Flow Diagram

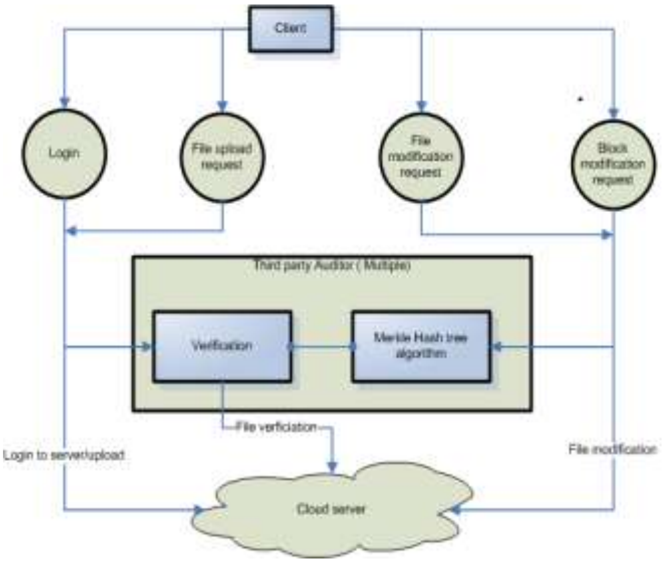


Figure. 2 Data Flow Diagram

D. Mathematical Model

1.Let S be the cloud security model/system
I, O, N, L, F where,
S={I,O,C,T,F }
I=Input
O=Output
C= Clients includes in system
T= Third party auditor in system
F= Security scheme of system
2. Let C will be number of client in system
C={c1,c2,c3.cn }
3. Let T will be number of Third party auditor in system
T={tpa1,tpa2,tpa3..tpan }
4. Let F will different security scheme use for cloud data
F={ f1,f1..fn }
5. Algorithm use for security

Key Generation algorithm :(Pk; Sk) <-- [KeyGen(1k)].This probabilistic algorithm is run by the client. It is taking input as security parameter 1k, and gives output as public key Pk and private key Sk.

Signature Generation algorithm: This algorithm is run by the client. It is taking input as private key Sk and a file F which is an ordered collection of blocks mi, and provides outputs as signature set[φ]which is an ordered collection of signatures [α] on mi.

E. Algorithms

1) Algorithm For Data Integrity Verification

1. Start the data integrity verification process.
2. TPA produces a random set.
3. CSS calculates root hash code founded on the filename/ blocks input.
4. CSS computes the value which is originally stored.
5. TPA decrypts the given content and compares with produced root hash.
6. After confirmation, the TPA can determine whether the integrity is breached.
7. Stop.

2) Algorithm For Updating and Deleting Data Present in CSS

1. Start the update/delete process.
2. Client produces new Hash for tree then sends it to CSS.
3. CSS updates F(file) and computes new root (R).
4. Client computes Root (R).
5. By Client signature is confirmed .If it is fails then output is FALSE .
6. Compute new R and confirm the update.
7. Stop

IV. PRACTICAL RESULT AND ENVIRONMENT

- Following Fig.3 showing the Third Party screen. This window shoes the third party and it will start the server or requester.
- The third party accepts the input data from requester and makes the changes like update.
- Following Fig.4 shows the client after starting the service.

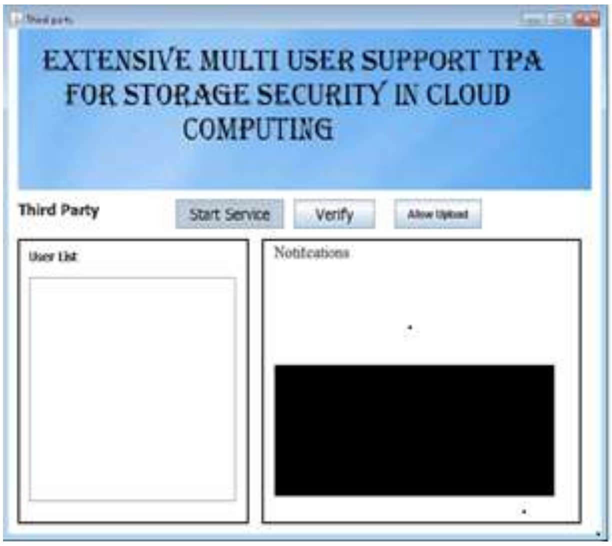
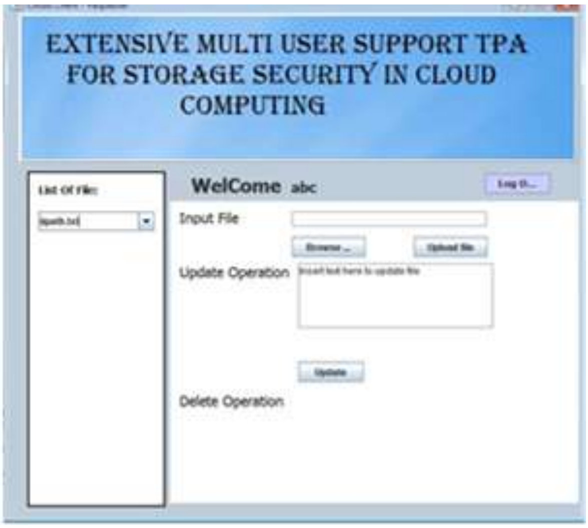

EXTENSIVE MULTI USER SUPPORT TPA FOR STORAGE SECURITY IN CLOUD COMPUTING

Figure. 3 Third party window



EXTENSIVE MULTI USER SUPPORT TPA FOR STORAGE SECURITY IN CLOUD COMPUTING

Figure. 4 Requester Window

V. CONCLUSION

Cloud data storage need to ensure the safety of data. It is an objective and independent approach to assessing the quality of service. Clients do not want to devote their time for continuously performing data validation task and thus delegate their work to TPA to check for data integrity as TPA provides reliable computing performance verifications which don't require resources commitment from client. This paper focus on the problem of employing simultaneous public auditability and data dynamics for remote data integrity check in Cloud Computing is explored. The model is designed for meeting these two main goals but efficiency is set as the main goal. For achieving data dynamics that are effective, the existing proof of storage models is more improved through use of the construction of classic Merkle Hash Tree for authentication of block tag. For better support of multiple numbers of auditing tasks, the method of signature is further explored for extending the main result into a multi-user setting, where TPA is able to perform multiple auditing tasks in a simultaneous manner. Heavy security as well as performance analysis proves that the proposed plan is efficient and safe to a greater extent.

ACKNOWLEDGMENT

I am Swati Khambete and would like to thank the publishers,researchers for making their resources available and Mr.Abhijit Patankar for their guidance. We also thank the college authorities Mrs. M. A. Potey: Head of Computer Department and Ms. S. S. Pawar: PG coordinator for providing the required infrastructure and support. Finally, we would like to extend a heartfelt gratitude to friends and family members.

REFERENCES

[1] Cong Wang, Qian Wang, Kui Ren, Ning Cao, Wenjing Lou," Toward Secure and Dependable Storage Services in Cloud Computing" ,IEEE Transaction on service computing vol. 5 no. 2,2012.

[2] Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, and Jin Li",Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing" IEEE Transaction on PARALLEL AND DISTRIBUTED SYSTEMS VOL. 22, NO. 5, MAY 2011.

[3] 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 (CCS07) pp. 598-609, 2007.

[4] A. Juels and 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] Tripathi, A. Mishra, A. IT Div.," Cloud Computing Security Considerations,Signal Processing, Communications and Computing (ICSPCC)" IEEE International Conference 2011.

[6] Haskar P., Admela J, Dimitrios K, Yves G.," Architectural Requirements for Cloud Computing Systems", An Enterprise Cloud Approach. J. Grid Computing 9(1), 3- 26 (2011).

[7] Q. Wang, K. Ren, W. Lou, and Y. Zhang," Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance", Proc. IEEE INFOCOM pp. 954-962, Apr. 2009.

[8] H. Shacham and B. Waters,"Compact Proofs of Retrievability", Proc.14th Intl Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT 08) pp. 90-107, 2008.

[9] K.D. Bowers, A. Juels, and A. Oprea",Proofs of Retrievability: Theory and Implementation", Report 2008/175, Cryptology ePrint Archive 2008.

[10] C. Wang, Q. Wang, K. Ren, and W. Lou",Ensuring Data Storage Security in Cloud Computing", Proc. 17th IntlWorkshop Quality of Service (IWQoS 09) 2009.

[11] C. Wang, B.Zhang, K Ren, J. M. Roved, Privacy-Assured Outsourcing of Image Reconstruction Service in Cloud , IEEE TRANSACTIONS ON March 2013.

[12] Kan Yang, Xiaohua Jia, An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 2013.

[13] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia," Dynamic Provable Data Possession", Proc. 16th ACM Conf. Computer and Comm.Security (CCS 09) 2009.

[14] K.D. 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) pp. 187-198, 2009.

[15] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession",Proc. Fourth Intl Conf. Security and Privacy in Comm. Networks (SecureComm 08) pp. 1-10, 2008.

[16] E.C. Chang and J. Xu," Remote Integrity Check with Dishonest Storage Server",Proc. 13th European Symp. Research in Computer Security (ESORICS 08) pp. 223-237, 2008.

[17] M.A. Shah, R. Swaminathan, and M. Baker, "Privacy-Preserving Audit and Extraction of Digital Contents", Report 2008/186, Cryptology ePrint Archive 2008.

[18] T. Schwarz and E.L. Miller, Store, "Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage", Proc. 26th IEEE Intl Conf. Distributed Computing Systems (ICDCS06) p. 12, 2006.

[19] M. Naor and G.N. Rothblum, "The Complexity of Online Memory Checking", Proc. 46th Ann. IEEE Symp. Foundations of Computer Science (FOCS 05) pp. 573-584, 2005.

[20] A. Oprea, M.K. Reiter, and K. Yang, "Space-Efficient Block Storage Integrity", Proc. 12th Ann. Network and Distributed System Security Symp. (NDSS 05) 2005. S. Lin an.