INTERNATIONAL JOURNAL OF ADVANCE RESEARCH AND DEVELOPMENT

(Volume2, Issue6)

Available online at www.ijarnd.com

Public Auditing For Shared Data with Efficient User Revocation in the Cloud

[1]Dr. N. Revathy, [2]Mr. R. Ramesh

[1]Associate Professor, [2]MCA

[1,2]Department of Master of Computer Applications,

[1,2]Hindusthan College of Arts and Science Coimbatore, India

drnrevathy@gmail.com

Abstract: With data storage and sharing services in the cloud, users can easily modify and share data as a group. To ensure shared data integrity can be verified publicly, users in the group need to compute signatures on all the blocks in shared data. Different blocks in shared data are generally signed by different users due to data modifications performed by different users. For security reasons, once a user is revoked from the group, the blocks which were previously signed by this revoked user must be re-signed by an existing user. The straightforward method, which allows an existing user to download the corresponding part of shared data and re-sign it during user revocation, is inefficient due to the large size of shared data in the cloud. In this paper, we propose a novel public auditing mechanism for the integrity of shared data with efficient user revocation in mind. By utilizing the idea of proxy re-signatures, we allow the cloud to resign blocks on behalf of existing users during user revocation so that existing users do not need to download and re-sign blocks by themselves. In addition, a public verifier is always able to audit the integrity of shared data without retrieving the entire data from the cloud, even if some part of shared data has been re-signed by the cloud. Moreover, our mechanism is able to support batch auditing by verifying multiple auditing tasks simultaneously. Experimental results show that our mechanism can significantly improve the efficiency of user revocation.

Keywords: Cloud Computing, Scalability, TPA (Third Party Auditor).

I. INTRODUCTION

Domain Description

Cloud computing is an internet technology that utilizes both central remote servers and the internet to manage the data and applications. This technology allows many businesses and users to use the data and application without an installation. Users and businesses can access the information and files at any computer system having an internet connection. Cloud computing provides much more effective computing by centralized memory, processing, storage, and bandwidth. Cloud computing is a term to describe a technology that distributes computer services away from a local client. For companies, it represents a powerful distribution model, because it shifts the investment away from "just in case" network power, to "pay for usage" and thereby maximizing the ability of their users while avoiding idle network processing. Cloud computing is an internet technology that utilizes both central remote servers and the internet to manage the data and applications. This technology allows many businesses and users to use the data and application without an installation.

Users and businesses can access the information and files at any computer system having an internet connection. Cloud computing provides much more effective computing by centralized memory, processing, storage, and bandwidth. The process of cloud computing technology is broken down into three segments. Platform, Applications, and Infrastructure are three segments of this technology. Each part serves many functions and provides applications for both individuals and businesses all around the world.

Segments:

- Applications- On Demand
- Infrastructure
- Platform.

Applications- On Demand

Application segment is the only part of internet technology that has been proved as a useful business model. By accessing many business and individual applications all over the cloud from the central server, most of the businesses can cut some very serious costs. On the other side, on-demand applications come in dissimilar varieties of pricing schemes and how the internet applications are delivered to the customers.

Infrastructure

Infrastructure is the backbone of the whole concept of this technology. All infrastructure vendors allow users to make their own cloud applications. Amazon's S3 is considered as a segment of the infrastructure segment.

Platform

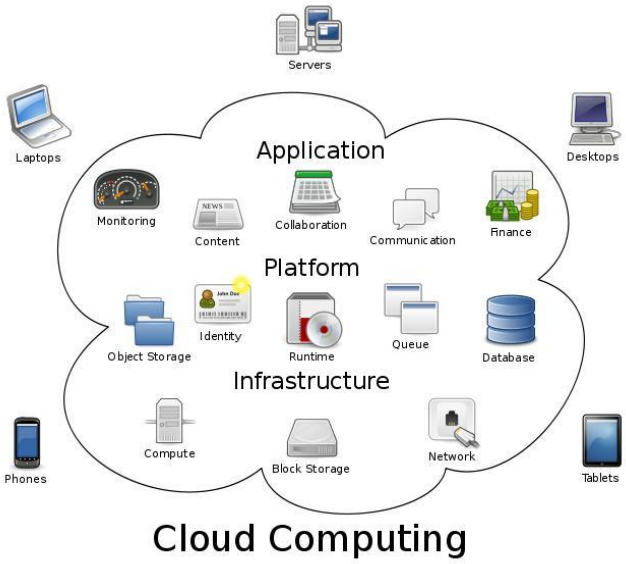
Most of the companies, who are providing several on demand services and applications, have been developed three platform services. These are Infrastructure as a Service, Platform as a Service and Software as a Service.

There are many types of public cloud computing:

- Infrastructure as a service (IaaS)
- Platform as a service (PaaS)
- Software as a service (SaaS)
- Storage as a service (STaaS)
- Security as a service (SECaaS)
- Data as a service (DaaS)
- Test environment as a service (TEaaS)
- Desktop as a service (DaaS)
- API as a service (APIaaS)

Cloud computing provides much more effective computing by centralized memory, processing, storage, and bandwidth. Cloud computing is a term to describe a technology that distributes computer services away from a local client. Cloud computing is an internet technology that utilizes both central remote servers and the internet to manage the data and applications. This technology allows many businesses and users to use the data and application without an installation. Users and businesses can access the information and files at any computer system having an internet connection Cloud computing are an internet technology that utilizes both central remote servers and the internet to manage the data and applications.

II.AN OVERVIEW OF THE SYSTEM



With data storage and sharing services (such as Dropbox and 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 errors To protect the integrity of data in the 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 all the signatures. One of the most significant and common features of these mechanisms is to allow a public verifier to efficiently check data integrity in the cloud without downloading the entire data, referred to as public auditing (or denoted as Provable Data Possession. This public verifier could be a client who would like to utilize cloud data for particular purposes (e.g., search, computation, data mining, etc.) or a third-party auditor (TPA) who is able to provide verification services on data integrity to users. Most of the previous works focus on auditing the integrity of personal data. Different from these works, several recent works focus on how to preserve identity privacy from public verifiers when auditing the integrity of shared data. Unfortunately, none of the above mechanisms considers the efficiency of user revocation when auditing the correctness of shared data in the cloud. With shared data, once a user modifies a block, she also needs to compute a new signature for the modified block. Due to the modifications from different users, different blocks are signed by different users. For security reasons, when a user leaves the group or misbehaves, this user must be revoked from the group. As a result, this revoked user should no longer be able to access and modify shared data, and the signatures generated by this revoked user are no longer valid to the group. Therefore, although the content of shared data is not changed during user revocation, the blocks, which were previously signed by the revoked user, still need to be re-signed by an existing user in the group. As a result, the integrity of the entire data can still be verified with the public keys of existing users only.

III.LITERATURE REVIEW

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 untrusted 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 opposed to linear in the size of the data. Experiments using our implementation verify the practicality of PDP and reveal that the performance of
PDP is bounded by disk I/O and not by cryptographic computation.

INTRODUCTION

Verifying the authenticity of data has emerged as a critical issue in storing data on untrusted servers. It arises in peer- to-peer storage systems, network file systems, long-term archives, web-service object stores, and database systems. Such systems prevent storage servers from misrepresenting or modifying data by providing authenticity checks when accessing data. However, archival storage requires guarantees about the authenticity of data on storage, namely that storage servers possess data. It is insufficient to detect that data have been modified or deleted when accessing the data, because it may be too late to recover lost or damaged data. Archival storage servers retain tremendous amounts of data, little of which are accessed. They also hold data for long periods of time during which there may be exposure to data loss from administration errors as the physical implementation of storage evolves, e.g., backup and restore, data migration to new systems, and changing memberships in peer-to-peer systems.

Archival network storage presents unique performance demands. Given that file data are large and are stored at remote sites, accessing an entire file is expensive in I/O costs to the storage server and in transmitting the file across a network. Reading an entire archive, even periodically, greatly limits the scalability of network stores. (The growth in storage capacity has far outstripped the growth in storage access times and bandwidth). Furthermore, I/O incurred to establish data possession interferes with on-demand bandwidth to store and retrieve data. We conclude that clients need to be able to verify that a server has retained file data without retrieving the data from the server and without having the server access the entire file. Previous solutions do not meet these requirements for proving data possession. Some schemes provide a weaker guarantee by enforcing storage complexity: The server has to store an amount of data at least as large as the client’s data, but not necessarily the same exact data.

Moreover, all previous techniques require the server to access the entire file, which is not feasible when dealing with large amounts of data.

Compact Proofs of Retrievability

In a proof-of-retrievability system, a data storage center must prove to a verier 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 that is, it should be possible to extract the client's data from any prover that passes a verification check. In this paper, we give the _rst proof-of-retrievability schemes with full proofs of security against arbitrary adversaries in the strongest model, that of Juels and Kaliski.

Our first scheme, built from BLS signatures and secure in the random oracle model, features a proof-of-retrievability protocol in which the client's query and server's response are both extremely short. This scheme allows public verifiability: anyone can act as a verifier, not just the file owner. Our second scheme, which builds on pseudorandom functions (PRFs) and is secure in the standard model, allows only private verification. It features a proof-of-irretrievability protocol with an even shorter server's response than our first scheme, but the client's query is long. Both schemes rely on homomorphic properties to aggregate a proof into one small authenticator value.

Introduction: Proofs of storage. Recent visions of \cloud computing" and \software as a service" call for data, both personal and commercial, to be stored by third parties, but deployment has lagged. Users of outsourced storage are at the mercy of their storage providers for the continued availability of their data. Even Amazon's S3, the best-known storage service, has experienced significant downtime. The solution, as Shah et al. argue, is storage auditing: cryptographic systems that would allow users of outsourced storage services (or their agents) to verify that their data is still available and ready for retrieval if needed. Such a capability can be important to storage providers as well. Users may be reluctant to entrust their data to an unknown startup; an auditing mechanism can reassure them that their data is indeed still available.

Early proof-of-storage systems were proposed by Deswarte, Quisquater, and Safi was done Gazzoni Filho and Barreto, and Schwarz and Miller.

Evaluation: formal security models. Such proof-of-storage systems should be evaluated by both \systems" and \crypto" criteria. Systems criteria include: (1) the system should be as efficient as possible in terms of both computational complexity and communication complexity of the proof of storage protocol, and the storage overhead on the server should be as small as possible; (2) the system should allow unbounded use rather than imposing a priori bound on the number of audit protocol interactions; (3) verifiers should be stateless, and not need to maintain and update state between audits, since such state is difficult to maintain if the verifier's machine crashes or if the verifier's role is delegated to third parties or distributed among multiple machines.1 Statelessness and unbounded use are required for proof-of-storage systems with public verifiability, in which anyone can undertake the role of the verifier in the proof-of-storage protocol, not just the user who originally stored the _le. Public verifiability for proof-of-storage schemes was first proposed by Ateniese et al.

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. This unique attribute, however, poses many new security challenges which have not been well understood. 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 utilizing the homomorphic 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.

Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing

Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. It moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. This unique paradigm brings about many new security challenges, which have not been well understood. This work studies 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), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA eliminates the involvement of client through the auditing of whether his data stored in the cloud is indeed intact, which can be important in achieving economies of scale for Cloud Computing. The support for data dynamics via the most general forms of data operation, such as block modification, insertion, and deletion, is also a significant step toward practicality since services in Cloud Computing are not limited to archive or backup data only. While prior works on ensuring remote data integrity often lack the support of either public verifiability or dynamic data operations, this paper achieves both. We first identify the difficulties and potential security problems of direct extensions with fully dynamic data updates from prior works and then show how to construct an elegant verification scheme for seamless integration of these two salient features in our protocol design. In particular, to achieve efficient data dynamics, we improve the Proof of Retrievability model by manipulating the classic Merkle Hash Tree (MHT) construction for block tag authentication. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure.

Privacy-Preserving Public Auditing for Secure Cloud Storage

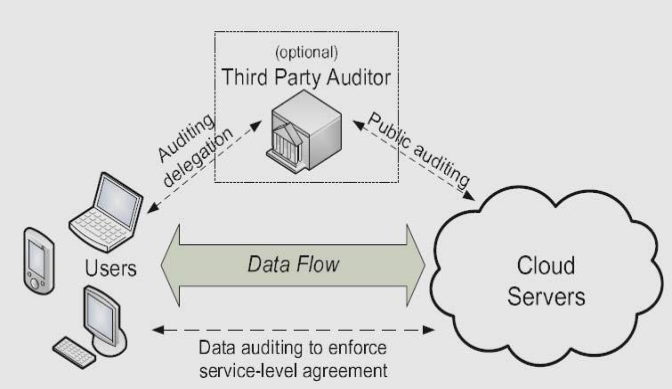
Using Cloud Storage, users can remotely store their data and enjoy the on-demand high-quality applications and services

from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud Computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public audibility for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to the user. In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient.

Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds

In this paper, we propose a dynamic audit service for verifying the integrity of untrusted and outsourced storage. Our audit service, constructed based on the techniques, fragment structure, random sampling and index hash table, can support probable updates to outsourced data, and timely abnormal detection. In addition, we propose an efficient approach based on the probabilistic query and periodic verification for improving the performance of audit services. Our experimental results not only validate the effectiveness of our approaches but also show our audit system has a lower computation overhead, as well as a shorter extra storage for audit metadata.

Secure and Dependable Storage Services in Cloud Computing



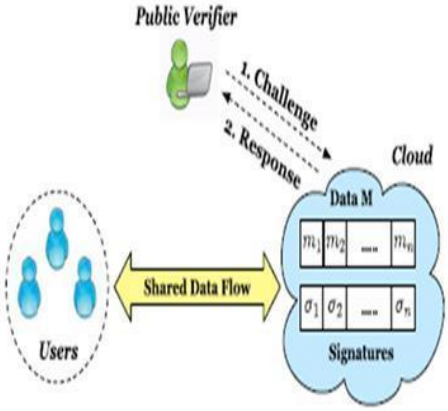
Cloud storage enables users to remotely store their data and enjoy the on-demand high-quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing users ’physical possession of their outsourced data, which inevitably poses new security risks towards the correctness of the data in the cloud. In order to address this new problem and further achieve a secure and dependable cloud storage service, I propose in this paper a flexible distributed storage integrity auditing mechanism, utilizing the homomorphism taken and distributed erasure-coded data. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result not only ensures strong cloud storage correctness guarantee but also simultaneously achieves fast data error localization, i.e., the Identification of misbehaving server. Considering the cloud data are dynamic in nature, the proposed design further supports secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append. The analysis shows the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

IV.SYSTEM IMPLEMENTATION

System Implementation is the stage in this paper where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and in giving confidence on the new system for the user that it will work efficiently and effectively.

The existing system was long time process. The proposed system was developed using Java Swing. The existing system caused long time transmission process but the system developed now has a very good user-friendly tool, which has a menu-based interface, graphical interface for the end user.

After coding and testing, the paper is to be installed on the necessary system. The executable file is to be created and loaded into the system. Again the code is tested in the installed system. Installing the developed code in the system in the form of executable file is implementation.



V.CONCLUSION

In this paper, we proposed a new public auditing mechanism for shared data with efficient user revocation in the cloud. When a user in the group is revoked, we allow the semi-trusted cloud to re-sign blocks that were signed by the revoked user with proxy re-signatures. Experimental results show that the cloud can improve the efficiency of user revocation, and existing users in the group can save a significant amount of computation and communication resources during user revocation.

VI.REFERENCES

[1] B. Wang, B. Li, and H. Li, "Public Auditing for Shared Data with Efficient User Revocation in the Cloud," Proc. IEEE INFOCOM, pp. 2904-2912, 2013.

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

[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 (CCS'07), pp. 598-610, 2007.

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

C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th ACM/IEEE Int'l Workshop Quality of Service (IWQoS'09),
pp. 1-9, 2009.

[6] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing," Proc. 14th European Conf. Research in Computer Security (ESORICS'09), pp. 355-370, 2009.

[7] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc. IEEE INFOCOM, pp. 525-533, 2010.

[8] 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," Proc. ACM Symp. Applied Computing (SAC'11), pp. 1550-1557, and 2011.

[9] C. Wang, Q. Wang, K. Ren, and W. Lou, "Towards Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Jan. 2012.

[10] Y. Zhu, G.-J. Ahn, H. Hu, S.S. Yau, H.G. An, and C.-J. Hu, "Dynamic Audit Services for Outsourced Storages in Clouds," IEEE Trans. Services Computing, vol. 6, no. 2, pp. 227-238, Apr.- June 2013.

[11] N. Cao, S. Yu, Z. Yang, W. Lou, and Y.T. Hou, "LT Codes-Based Secure and Reliable Cloud Storage Service," Proc. IEEE INFOCOM, pp. 693-701, 2012.

About Authors

Dr. Revathy Nanjappan had completed B.Sc., Computer Science in the year 2000 and Master of Computer Applications (MCA) in the year 2003 under Bharathiar University. Completed M.Phil in Computer Science from Alagappa University in the year 2005. Completed Ph.D. in Computer Science from Mother Teresa Women's University, Kodaikanal in the year 2013 and the area of research is Neural Networks. Other areas of interest are Mobile Computing, Data Mining, and Artificial Intelligence. At present working as an Associate professor in the Department of Master of Computer Applications at Hindusthan College of Arts and Science at Coimbatore-641 028 and published 11 papers in International Journals, presented 8 papers in International Conferences and 56 papers in National Conferences.

Ramesh Rathinasamy had completed B.Sc Chemistry in the year 2014. Currently pursuing Master of Computer Applications. Area of interest is software testing and animation. Attended an International Conference in the year 2016 at Hindusthan College of Arts and Science, Coimbatore, Tamilnadu, India

