ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Batch Verification Scheme for Economic Cloud Storage Services

K. Sudha[1], Mr. S. Sivaraj, ME[2]

II-M.E (CSE), Dept. of CSE, SSM College of Engineering, Komarapalayam, Tamilnadu, India[1]

Assistant Professor, Dept. of CSE, SSM College of Engineering, Komarapalayam, Tamilnadu, India[2]

Abstract: Cloud computing is used to share resources and data sources under computational applications. Hardware, software and information are provided in cloud environment. Cloud computing consolidates thousands of virtual machines. Private and public data are managed under the cloud environment. Remote data storages are used to share data and services in the cloud environment. Data provider uploads the shared data into the data centers. Public auditing methods are used to verify the data integrity in remote data storages. Third-party auditor (TPA) is used to check the integrity of outsourced data. Privacy preserving public auditing mechanism is used to verify the data integrity with privacy. TPA supports auditing for multiple users simultaneously. Batch auditing mechanism is used for multi user environment. Homomorphic linear authenticator and random masking techniques are used to protect the data from TPA. The privacy preserving public auditing scheme is enhanced to perform data verification for multi user environment. Batch verification scheme is adopted to multi user data sharing environment. Data dynamism is integrated with public data auditability scheme. Economic cloud operations are supported by the system with public auditing methods.

I. INTRODUCTION

Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the "software as a service" (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. Meanwhile, the increasing network bandwidth and reliable yet flexible network connections make it even possible that clients can now subscribe high-quality services from data and software that reside solely on remote data centers. Although envisioned as a promising service platform for the Internet, this new data storage paradigm in "Cloud" brings about many challenging design issues which have profound influence on the security and performance of the overall system. One of the biggest concerns with cloud data storage is that of data integrity verification at untrusted servers. For example, the storage service provider, which experiences Byzantine failures occasionally, may decide to hide the data errors from the clients for the benefit of their own. What is more serious is that for saving money and storage space the service provider might neglect to keep or deliberately delete rarely accessed data files which belong to an ordinary client. Consider the large size of the outsourced electronic data and the client's constrained resource capability, the core of the problem can be generalized as how can the client find an efficient way to perform periodical integrity verifications without the local copy of data files.

In order to solve the problem of data integrity checking, many schemes are proposed under different systems and security models. In all these works, great efforts are made to design solutions that meet various requirements: high scheme efficiency, stateless verification, unbounded use of queries and retrievability of data, etc. Considering the role of the verifier in the model, all the schemes presented before fall into two categories: private auditability and public auditability. Although schemes with private auditability can achieve higher scheme efficiency, public auditability allows anyone, not just the client (data owner), to challenge the cloud server for correctness of data storage while keeping no private information. Then, clients are able to delegate the evaluation of the service performance to an independent third party auditor (TPA), without devotion of their computation resources. In the cloud, the clients themselves are unreliable or may not be able to afford the overhead of performing frequent integrity checks. Thus, for practical use, it seems more rational to equip the verification protocol with public auditability, which is expected to play a more important role in achieving economies of scale for Cloud Computing. Moreover, for efficiency consideration, the outsourced data themselves should not be required by the verifier for the verification purpose.

Another major concern among previous designs is that of supporting dynamic data operation for cloud data storage applications. In Cloud Computing, the remotely stored electronic data might not only be accessed but also updated by the clients, e.g., through block modification, deletion, insertion, etc. Unfortunately, the state of the art in the context of remote data storage mainly focus on static data files and the importance of this dynamic data updates has received limited attention. Moreover, as will be shown later, the direct extension of the current provable data possession (PDP) or proof of retrievability (PoR) schemes to support data dynamics may lead to security loopholes. Although there are many difficulties faced by researchers, it is well believed that supporting dynamic data operation can be of vital importance to the practical application of storage outsourcing services. In view of the key role of public auditability and data dynamics for cloud data storage, we propose an efficient construction for the seamless integration of these two components in the protocol design.

Recently, much of growing interest has been pursued in the context of remotely stored data verification. Ateniese et al. are the first to consider public auditability in their defined “provable data possession” model for ensuring possession of files on untrusted storages. In their scheme, they utilize RSA-based homomorphic tags for auditing outsourced data, thus public auditability is achieved. However, Ateniese et al. do not consider the case of dynamic data storage, and the direct extension of their scheme from static data storage to dynamic case may suffer design and security problems. In their subsequent work, Ateniese et al. propose a dynamic version of the prior PDP scheme. However, the system imposes a priori bound on the number of queries and does not support fully dynamic data operations, i.e., it only allows very basic block operations with limited functionality, and block insertions cannot be supported. Wang et al. consider dynamic data storage in a distributed scenario, and the proposed challenge-response protocol can both determine the data correctness and locate possible errors. Similar to, they only consider partial support for dynamic data operation.

II. RELATED WORK

Juels et al. describe a “proof of retrievability” (PoR) model, where spot-checking and error-correcting codes are used to ensure both “possession” and “retrievability” of data files on remote archive service systems. However, the number of audit challenges a user can perform is fixed a priori, and public auditability is not supported in their main scheme. Although they describe a straightforward Merkle-tree construction for public PoRs, this approach only works with encrypted data. Later, Bowers et al. [4] propose an improved framework for POR protocols that generalizes Juels' work. Dodis et al. [5] also give a study on different variants of PoR with private auditability. Shacham and Waters design an improved PoR scheme built from BLS signatures with proofs of security in the security model defined. Similar to the construction, they use publicly verifiable homomorphic linear authenticators that are built from provably secure BLS signatures. Based on the elegant BLS construction, a compact and public verifiable scheme is obtained. Again, their approach is not privacy preserving due to the same reason.

Shah et al. [10] propose introducing a TPA to keep online storage honest by first encrypting the data then sending a number of precomputed symmetric-keyed hashes over the encrypted data to the auditor. The auditor verifies the integrity of the data file and the server's possession of a previously committed decryption key. This scheme only works for encrypted files, requires the auditor to maintain state, and suffers from bounded usage, which potentially brings in online burden to users when the keyed hashes are used up. Dynamic data have also attracted attentions in the recent literature on efficiently providing the integrity guarantee of remotely stored data. Ateniese et al. [1] is the first to propose a partially dynamic version of the prior PDP scheme, using only symmetric key cryptography but with a bounded number of audits. In [6], Wang et al. consider a similar support for partially dynamic data storage in a distributed scenario with additional feature of data error localization. In a subsequent work, Wang et al. [8] propose to combine BLS-based HLA with MHT to support fully data dynamics. Concurently, Erway et al. develop a skip list based scheme to also enable provable data possession with full dynamics support. However, the verification in both protocols requires the linear combination of sampled blocks as an input, and thus does not support privacy-preserving auditing.

In other related work, Sebe et al. thoroughly study a set of requirements which ought to be satisfied for a remote data possession checking protocol to be of practical use. Their proposed protocol supports unlimited times of file integrity verifications and allows preset tradeoff between the protocol running time and the local storage burden at the user. Schwarz and Miller propose the first study of checking the integrity of the remotely stored data across multiple distributed servers. Their approach is based on erasure-correcting code and efficient algebraic signatures, which also have the similar aggregation property as the homomorphic authenticator utilized in our approach. Curtmola et al. [2] aim to ensure data possession of multiple replicas across the distributed storage system. They extend the PDP scheme in [9] to cover multiple replicas without encoding each replica separately, providing guarantee that multiple copies of data are actually maintained. In [3], Bowers et al. utilize a two-layer erasure-correcting code structure on the remotely archived data and extend their POR model to distributed scenario with high-data availability assurance. While all the above schemes provide methods for efficient auditing and provable assurance on the correctness of remotely stored data, almost none of them necessarily meet all the requirements for privacy-preserving public auditing of storage. Moreover, none of these schemes consider batch auditing, while our scheme can greatly reduce the computation cost on the TPA when coping with a large number of audit delegations.

III. DATA VERIFICATION FOR CLOUDS

While cloud computing makes these advantages more appealing than ever, it also brings new and challenging security threats toward users' outsourced data. Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time [7]. Second, there do exist various motivations for CSP to behave unfaithfully toward the cloud users regarding their outsourced data status. For examples, CSP might reclaim storage for monetary reasons by discarding data that have not been or are rarely accessed, or even hide data loss incidents to maintain a reputation. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the success of cloud architecture. In particular, simply downloading all the data for its integrity verification is not a practical solution due to the expensiveness in I/O and transmission cost across the network. Besides, it is often insufficient to detect the data corruption only when accessing the data, as it does not give users correctness assurance for those unaccessed data and might be too late to recover the data loss or damage. Considering the large size of the outsourced data and the user's constrained resource capability, the tasks of auditing the data correctness in a cloud environment can be formidable and expensive for the cloud users. Moreover, the overhead of using cloud storage should be minimized as much as possible, such that a user does not need to perform too many operations to use the data (in additional to retrieving the data). In particular, users may not want to go through the complexity in verifying the data integrity. Besides, there may be more than one user accesses the same cloud storage, say in an enterprise setting. For easier management, it is desirable that cloud only entertains verification request from a single designated party.

Public auditability allows an external party, in addition to the user himself, to verify the correctness of remotely stored data. However, most of these schemes do not consider the privacy protection of users' data against external auditors. Indeed, they may potentially reveal user's data to auditors. This severe drawback greatly affects the security of these protocols in cloud computing. From the perspective of protecting data privacy, the users, who own the data and rely on TPA just for the storage security of their data, do not want this auditing process introducing new vulnerabilities of unauthorized information leakage toward their data security. Moreover, there are legal regulations, such as the US Health

Insurance Portability and Accountability Act (HIPAA), further demanding the outsourced data not to be leaked to external parties [10]. Simply exploiting data encryption before outsourcing could be one way to mitigate this privacy concern of data auditing, but it could also be an overkill when employed in the case of unencrypted/public cloud data, due to the unnecessary processing burden for cloud users. Besides, encryption does not completely solve the problem of protecting data privacy against third-party auditing but just reduces it to the complex key management domain. Unauthorized data leakage still remains possible due to the potential exposure of decryption keys.

Therefore, how to enable a privacy-preserving thirdparty auditing protocol, independent to data encryption, is the problem we are going to tackle in this paper. Our work is among the first few ones to support privacy-preserving public auditing in cloud computing, with a focus on data storage. Besides, with the prevalence of cloud computing, a foreseeable increase of auditing tasks from different users may be delegated to TPA. As the individual auditing of these growing tasks can be tedious and cumbersome, a natural demand is then how to enable the TPA to efficiently perform multiple auditing tasks in a batch manner, i.e., simultaneously.

IV. PROBLEM STATEMENT

To address these problems, our work utilizes the technique of public key-based homomorphic linear authenticator [9], [8], which enables TPA to perform the auditing without demanding the local copy of data and thus drastically reduces the communication and computation overhead as compared to the straightforward data auditing approaches. By integrating the HLA with random masking, our protocol guarantees that the TPA could not learn any knowledge about the data content stored in the cloud server (CS) during the efficient auditing process. The aggregation and algebraic properties of the authenticator further benefit our design for the batch auditing.

Remote data storages are used to share data and services in the cloud environment. Data provider uploads the shared data into the data centers. Public auditing methods are used to verify the data integrity in remote data storages. Third-party auditor (TPA) is used to check the integrity of outsourced data. Privacy preserving public auditing mechanism is used to verify the data integrity with privacy. TPA supports auditing for multiple users simultaneously. Batch auditing mechanism is used for multi user environment. Homomorphic linear authenticator and random masking techniques are used to protect the data from TPA. The following drawbacks are identified in the existing system. Data dynamism is not tuned for batch auditing scheme. Commercial cloud operations are not supported by the system. Data dynamism is not adapted for privacy preserved auditing mechanism. Privacy is provided for single user verification process.

V. BATCH VERIFICATION SCHEME FOR ECONOMIC CLOUDS

The privacy preserving public auditing scheme is enhanced to perform data verification for multi user environment. Batch verification scheme is adopted to multi user data sharing environment. Data dynamism is integrated with public data auditability scheme. The system is improved to support public auditing based data sharing under commercial cloud environment. The cloud data sharing scheme is designed to manage data sharing based on economic model. Batch auditing mechanism is adapted for the data verification process. Dynamic data updates are managed with auditing process. The system is divided into five major modules. They are data center, Third Party Auditor, client, data dynamism handler and batch auditing. The cloud data center manages the shared data values. Auditing operations are initiated by the Third Party Auditor. Client application is designed to manage data upload and download operations. Data update operations are managed under data dynamism module. Batch auditing is designed for multi user data verification process.

5.1. Data Center

The data center application is designed to allocate storage space for the data providers. Data center maintains data files for multiple providers. Different sized storage area is allocated for the data providers. Data files are delivered to the clients. To enable privacy-preserving public auditing for cloud data storage under the aforementioned model, our protocol design should achieve the following security and performance guarantees. Public auditability: to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to the cloud users. Storage correctness: to ensure that there exists no cheating cloud server that can pass the TPA's audit without indeed storing users' data intact. Privacy preserving: to ensure that the TPA cannot derive users' data content from the information collected during the auditing process. Batch auditing: to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously. Lightweight: to allow TPA to perform auditing with minimum communication and computation overhead.

5.2. Third Party Auditor

The Third Party Auditor (TPA) maintains the signature for shared data files. TPA performs the public data verification for data providers. Data integrity verification is performed is using Secure Hashing Algorithm (SHA). Homomorphic linear authenticator and random masking techniques are used for privacy preservation process. We consider a cloud data storage service involving three different entities. The cloud user, who has large amount of data files to be stored in the cloud; the cloud server, which is managed by the cloud service provider to provide data storage service and has significant storage space and computation resources; the third-party auditor, who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. As users no longer possess their data locally, it is of critical importance for users to ensure that their data are being correctly stored and maintained. To save the computation resource as well as the online burden potentially brought by the periodic storage correctness verification, cloud users may resort to TPA for ensuring the storage integrity of their outsourced data, while hoping to keep their data private from TPA.

5.3. Client

The client application is designed to access the hard data values. The cloud user initiates the download process. Data access information is updated to the data center. Data center transfers the data as blocks. A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, VerifyProof). KeyGen is a key generation algorithm that is run by the user to setup the scheme. SigGen is used by the user to generate verification metadata, which may consist of digital signatures. GenProof is run by the cloud server to generate a proof of data storage correctness, while VerifyProof is run by the TPA. The user initializes the public and secret parameters of the system by executing KeyGen, and preprocesses the data file F by using SigGen to generate the verification metadata. The user then stores the data file F and the verification metadata at the cloud server, and deletes its local copy. As part of preprocessing, the user may alter the data file F by expanding it or including additional metadata to be stored at server. The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit. The cloud server will derive a response message by executing GenProof using F and its verification metadata as inputs. The TPA then verifies the response via VerifyProof.

5.4. Data Dynamism Handler

Shared data values are managed with blocks. Block update and delete operations are handled with signature update process. Block insertion operations are also supported in data dynamism process. Block signatures are also updated in data dynamism process. To achieve privacy-preserving public auditing, we propose to uniquely integrate the homomorphic linear authenticator with random masking technique. In our protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated by the server. With random masking, the TPA no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the user's data content, no matter how many linear combinations of the same set of file blocks can be collected. On the other hand, the correctness validation of the block-authenticator pairs can still be carried out in a new way which will be shown shortly, even with the presence of the randomness. Our design makes use of a public key-based HLA, to equip the auditing protocol with public auditability. Specifically, we use the HLA proposed, which is based on the short signature scheme proposed by Boneh, Lynn, and Shacham.

5.5. Batch Auditing

Data integrity verification is carried out under auditing process. Batch auditing is applied to perform simultaneous data verification process. Batch auditing is tuned for multi user environment. Data dynamism is integrated with batch auditing process. This section presents our public auditing scheme which provides a complete outsourcing solution of data—not only the data itself, but also its integrity checking. After introducing notations and brief preliminaries, we start from an overview of our public auditing system and discuss two straightforward schemes and their demerits. Then, we present our main scheme and show how to extent our main scheme to support batch auditing for the TPA upon delegations from multiple users. Finally, we discuss how to generalize our privacy-preserving public auditing scheme and its support of data dynamics.

VI. CONCLUSION

Cloud computing environment supports data sharing on third party storages. Cloud data providers maintain the shared data in remote storages. TPA mechanism is used to verify data integrity in cloud storages. Batch auditing mechanism is provided for multi user environment. Commercial cloud services are supported with privacy preserved data verification schemes. The system support multi user data verification process. Client resource consumption is reduced by the system. Data dynamism is supported for multi user environment. Simultaneous data verification is performed in batch verification mechanism. Data verification is carried out simultaneously for all data providers.

REFERENCES

[1] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, “Scalable and Efficient Provable Data Possession,” Proc. Int’l Conf. Security and Privacy in Comm. Networks, pp. 1-10, 2008.
[2] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, “MR-PDP: Multiple-Replica Provable Data Possession,” Proc. IEEE Int’l Conf. Distributed Computing Systems, pp. 411-420, 2008.
[3] 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, 2009.
[4] K.D. Bowers, A. Juels, and A. Oprea, “Proofs of Retrievability: Theory and Implementation,” Proc. ACM Workshop Cloud Computing Security, 2009.
[5] Y. Dodis, and D. Wichs, “Proofs of Retrievability via Hardness Amplification,” Proc. Theory of Cryptography Conf. Theory of Cryptography, 2009.
[6] C. Wang, and W. Lou, “Towards Secure and Dependable Storage Services in Cloud Computing,” IEEE Trans. Service Computing, Apr.-June 2012.
[7] Amazon.com, “Amazon s3 Availability Event: July 20, 2008,” July 2008.
[8] Q. Wang, W. Lou, and J. Li, “Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing,” IEEE Trans. Parallel and Distributed Systems, May 2011.
[9] G. Ateniese, R. Burns, and D. Song, “Provable Data Possession at Untrusted Stores,” Proc. 14th ACM Conf. Computer and Comm. Security, 2007.
[10] M.A. Shah, R. Swaminathan, and M. Baker, “Privacy-Preserving Audit and Extraction of Digital Contents,” Cryptology ePrint Archive, Report 2008/186, 2008.