Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-6, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

Ensuring Data Integrity for Resource Constrained Devices in Storage Cloud

Shalini J[1] & Dr. K. Raghuveer[2]
[1]M.tech Student Department of Information Science and Engineering, The National Institute Of engineering, Mysore, India
[2]Professor & Head Department of Information Science and Engineering, The National Institute Of engineering, Mysore, India

Abstract— *with increasing volume of data usage by mobile users with their smart phones, the users are using storage clouds to store their data on cloud. When they are doing so the data stored on the cloud must be secure and integral. Security can be implemented by using encryption and integrity can be implemented by using auditing. But these features cannot be implemented on resource constrained smart phones with control still at mobile users. In this work, we propose a trusted third party auditing solution which offloads the auditing functionality to third party auditor with control of it in mobile user hands.*

I. Introduction

With raise of smart phone revolution, lot of data is at users hands. The smart phones cannot store huge volumes of data and user uses cloud for storage of his data. Some data may be sensitive and cannot be stored as plain text. Encryption is the solution and the data must be encrypted to store the data. But still data stored in cloud can suffer from data integrity problem. Attacker can modify the data in the cloud. Once it is done, there must a way for the user to know that his data at cloud is modified. This can be done by user frequently polling cloud and checking the integrity of data. But this is a huge overhead at mobile user end as this operation will consume major share of resources in the smart phone and the phone will become slow. There must a way to audit without affecting the performance of users smartphone.

The auditing must be supported batch mode and individual mode and the control of auditing must be at the user end. Also the auditor must never be able to find the content of file and must do auditing without compromising the file security.

The system must also be secure against various attacks like replay attack and must work for any kind of files. Most of existing solutions don't consider the replay attack and file types. Also most schemes don't consider file types and involve lot of network overhead in communication and user has to bear the cost due to network communication. In our solution we consider this and propose a new auditing solution for resource constrained devices.

II. Related Work

In this section we survey the protocols for ensuring integrity of cloud storage.

G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores,"[1]
In this paper author introduceda 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.

Author focused on the problem of verifying if an untrusted server stores a client's data. Our solutions for PDP fit this model: They incur a low (or even constant) overhead at the server and require a small, constant amount of communication per challenge. Key components of our schemes are the homomorphic verifiable tags. They allow verifying data possession

without having access to the actual data file.
Juels and B. S. K. Jr., "Pors: proofs of retrievability for large files,"[2]

In this paper, author define and explore proofs of retrievability(PORs). A POR scheme enables an archive or back-up service (prover) to produce a concise proof that a user (verifier) 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 (POK), but one specially designed to handle a large file (or bitstring) F. We explore POR protocols here in which the communication costs, number of memory accesses for the prover, and storage requirements of the user (verifier) are small parameters essentially independent of the length of F. In addition to proposing new, practical POR constructions, we explore implementation considerations and optimizations that bear on previously explored, related schemes. In a POR, unlike a POK, neither the prover nor the verifier need actually have knowledge of F. PORs give rise to a new and unusual security definition whose formulation is another contribution of our work. We view PORs as an important tool for semi-trusted online archives. Existing cryptographic techniques help users ensure the privacy and integrity of files they retrieve. It is also natural, however, for users to want to verify that archives do not delete or modify files prior to retrieval. The goal of a POR is to accomplish these checks without users having to download the files themselves. A POR can also provide quality-of-service guarantees, i.e., show that a file is retrievable within a certain time bound.

Qian Wang1, Cong Wang1, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing"[25]
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 lacks 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 [1] 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.
Cong Wang, Student Member "Privacy-Preserving Public Auditing for Secure Cloud Storage"[12]
In this paper, author proposed 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. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design.

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 auditability 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 user.

,
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," [13]
In this paper, author proposed 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 provable updates to outsourced data, and timely abnormal detection. In addition, we propose an efficient approach based on 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.

In this paper, author introduces a dynamic audit service for integrity verification of untrusted and outsourced storages. Our audit system, based on a novel audit system architecture, can support dynamic data operations and timely abnormal detection with the help of several effective techniques, such as fragment structure, random sampling, and index-hash table. Furthermore, we propose an efficient approach based on probabilistic query and periodic verification for improving the performance of audit services. A proof of concept prototype is also implemented to evaluate the feasibility and viability of our proposed approaches. Our experimental results not only validate the effectiveness of our approaches, but also show our system has a lower computation cost, as well as a shorter extra storage for integrity verification.

III. PROBLEM DEFINITION

Mobile constrained devices upload their encrypted files to cloud and these files has to be audited by third party semi trusted auditor and notify the user whenever the file is modified by attacker in cloud.
Also the network overhead for detection and notification of attack must be as low as possible to keep the network communication cost to a low value.

IV. PROPOSED SOLUTION

Architecture of the system is given below

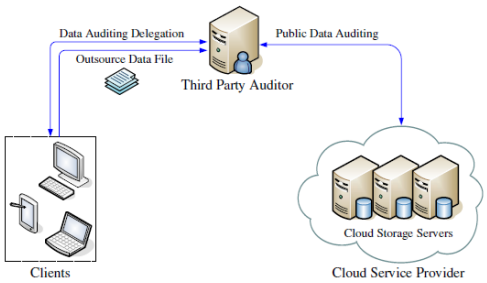
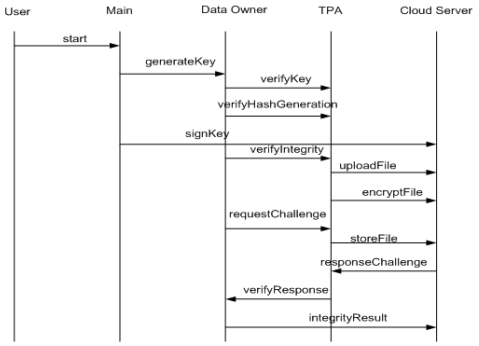


Fig. 1: Cloud data storage architecture

Client module: an entity that has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations.
Cloud Storage Server (CSS) module: an entity, which is managed by Cloud Service Provider (CSP), has significant storage space and computation resource to maintain client's data. The CSS is required to provide integrity proof to the clients or cloud audit server during the integrity checking phase.
Cloud Audit Server (CAS) module: a TPA, which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request. In this system, the cloud audit server also generates all the tags of the files for the users before uploading to the cloud storage server.
The basic goal of PoR model is to achieve proof of retrievability. Informally, this property ensures that if an adversary can generate valid integrity proofs of any file F for a non-negligible fraction of challenges, we can construct a PPT machine to extract F with overwhelming probability.
It is formally defined by the following game between a challenger C and an adversary A, where C plays the role of the audit server (the client) and A plays the role of the storage server:

The sequence of steps for auditing is as follows

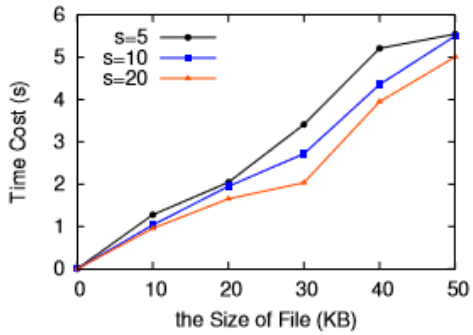


The hash is computed using the merkel algorithm for files by the user and uploaded to auditor with a seed key. The auditor sends encrypted challenge to cloud server and verifies the challenge response from the cloud server, if the signature matching fails user will be notified by email.

The hash computed is very less size, after merkel, so that the network overhead incurred for communication between auditor and cloud service provider is reduced.

V. RESULTS

We implemented the proposed system in java and measured the auditing time for files of different size and given below



VI. CONCLUSION

In this work, we have explained the EasyCrash solution. We have ranked the function based on 5 metric we have proposed and reduced the developer effort in terms of analyzing each crash to fixing the functions.

REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in CCS '07: Proceedings of the 14th ACM conference on Computer and communications security. New York, NY, USA: ACM, 2007, pp. 598–609.

[2] A. Juels and B. S. K. Jr., "Pors: proofs of retrievability for large files," in CCS '07: Proceedings of the 14th ACM conference on Computer and communications security. New York, NY, USA: ACM, 2007, pp. 584–597.

[3] H. Shacham and B. Waters, "Compact proofs of retrievability," in ASIACRYPT '08: Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 90–107.

[4] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: theory and implementation," in Proceedings of CCSW 2009. ACM, 2009, pp. 43–54.

[5] M. Naor and G. N. Rothblum, "The complexity of online memory checking," J. ACM, vol. 56, no. 1, pp. 2:1–2:46, Feb. 2009. [Online]. Available: http://doi.acm.org/10.1145/1462153. 1462155

[6] E.-C. Chang and J. Xu, "Remote integrity check with dishonest storage server," in Proceedings of ESORICS 2008, volume 5283 of LNCS. Springer-Verlag, 2008, pp. 223–237.

[7] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008, http://eprint.iacr.org/.

[8] A. Oprea, M. K. Reiter, and K. Yang, "Space-efficient block storage integrity," in In Proc. of NDSS 2005, 2005.

[9] T. S. J. Schwarz and E. L. Miller, "Store, forget, and check: Using algebraic signatures to check remotely administered storage," in ICDCS '06: Proceedings of the 26th IEEE International Conference on Distributed Computing Systems. Washington, DC, USA: IEEE Computer Society, 2006.

[10] Q. Wang, K. Ren, S. Yu, and W. Lou, "Dependable and secure sensor data storage with dynamic integrity assurance," ACM Transactions on Sensor Networks, vol. 8, no. 1, pp. 9:1–9:24, Aug. 2011. [Online]. Available: http: //doi.acm.org/10.1145/1993042.1993051

[11] L. V. M. Giuseppe Ateniese, Roberto Di Pietro and G. Tsudik, "Scalable and efficient provable data possession," in International Conference on Security and Privacy in Communication Networks (SecureComm 2008), 2008.

[12] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in INFOCOM, 2010, pp. 525–533.

[13] 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," in SAC, 2011, pp. 1550–1557.

[14] Q. Zheng and S. Xu, "Fair and dynamic proofs of retrievability," in CODASPY, 2011, pp. 237–248.

[15] J. Li, X. Chen, J. Li, C. Jia, J. Ma, and W. Lou, "Fine-grained access control system based on attribute-based encryption," ESORICS, 2013.