Survey on data Integrity Approaches used in the Cloud Computing

T S Khatri
M.E Computer Engineering
Parul Institute of Engineering & Technology,
Limda,India

Prof G B Jethava
Information Technology Department
Parul Institute of Engineering & Technology,
Limda,India

Abstract

Cloud Computing has been envisioned as the next generation architecture of IT Enterprise .Clients release their burden of storing and maintaining the data locally by storing it over the cloud. As cloud provides many advantages, it also brings certain challenges. Though client cannot physically access the data from the cloud server directly, without client's knowledge, cloud provider can modify or delete data which are either not used by client from a long a time or occupies large space. Hence, there is a requirement of checking the data periodically for correction purpose, checking data for correction is called data integrity. In this paper we provide survey on the different techniques of data integrity. The basic schemes for data integrity in cloud are Provable Data Possession (PDP) and Proof of Retrievability (PoR). These two schemes are most active area of research in the cloud data integrity field. The objective of this survey is to offer new researchers a guideline, who are interested in this specific area and to understand the research work carried out in last few years.

Keyword: Cloud Computing, data Integrity, PDP, PoR

1. Introduction

Cloud computing is defined [16] as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud computing offers different types of services, including Software as a Service (SAAS), Platform as a Service (PAAS) and Infrastructure as a Service (IAAS). Cloud Storage is an important service of cloud computing, which allows data owners to move data from their local computing systems to the Cloud. More and more data owners start choosing to host their data in the Cloud. For small and medium sized businesses it is very cost effective, this is the main reason behind moving towards cloud. By hosting their data in the Cloud, data owners can avoid the initial investment of expensive infrastructure setup, large equipments, and daily maintenance cost. The data owners only need to pay the space they actually use. Another reason is that data owners can rely on the Cloud to provide more reliable services, so that they can access data from anywhere and at any time. Individuals or small-sized companies usually do not have the resource to keep their servers as reliable as the Cloud does have.

While Cloud Computing makes these advantages more appealing than ever, it also brings new challenging security threats towards user's outsourced data. Sometimes, the cloud service providers may be dishonest and they may discard the data which has not been accessed or rarely accessed to save the storage space or keep fewer replicas than promised [13]. Moreover, the cloud service providers may choose to hide data loss and claim that the data are still correctly stored in the Cloud. As a result, data owners need to be convinced that their data are correctly stored in the Cloud. So, one of the biggest concerns with cloud data storage is that of data integrity verification at untrusted servers. In order to solve the problem of data integrity checking, many schemes are proposed under different systems and security models.

In this paper we surveyed two core integrity

proving schemes in detail along with different methods used for data integrity in both the schemes. Moreover, we have described other methods proposed in the area. The rest of the paper is organized as follows: section 2 explains principles of data integrity verification that cover system model, threat model and objectives of data integrity. Section 3 reviews different data integrity proving schemes. We conclude our work in section 4.

2. Principles of Data Integrity

2.1 System Model

Cloud storage applications offer client (data owner) the opportunity to store, backup or archive their data in the cloud storage network. Such applications should ensure data integrity and availability on a long term basis. This objective requires developing appropriate remote data possession verification. Representative network architecture for cloud data storage is illustrated in Fig. 1. As shown in figure three different network entities can be identified as follows:

Client: Users, who have data to be stored in the cloud and rely on the cloud for data computation, consist of both individual consumers and organizations.

Cloud Storage server: is managed by the cloud service provider (CSP) to provide data storage service and has significant storage space and computation resources.

Third Party Auditor (TPA): an optional TPA, who has expertise and capabilities that users may not have, is trusted to assess and expose risk of cloud storage services on behalf of the users open request.

Verifier may be User (Data owner) or third party auditor. The role of the verifier fall into two categories:

Private Auditability: It allows only data owner for checking the integrity of the data file stored on cloud server.

Public Auditability: It allows anyone, not just the client (data owner), to challenge the cloud server for correctness of data.

2.2 Threat Model

We consider the third party auditor is honest-but-curious. It performs honestly during the whole auditing procedure but it is curious about the received data. Thus, for the storage of secured data, there is also a privacy requirement for the third party auditing protocol. That is, no data will be leaked out to the third party auditor during the auditing procedure. But the server is dishonest and may conduct the following attacks:

- **Replace Attack:** Suppose the Server discarded a challenged data block m_i or its metadata t_i, in order to pass the auditing, it may choose another valid and uncorrupted pair of data block and metadata (m_k, t_k) to replace the original challenged pair of data block and metadata (m_k, t_k).
- **Replay Attack:** The Server generates the proof from the previous proof or other information, without querying the actual Owner's data.
- **Forge Attack:** The Server may forge the metadata of data block and deceive the auditor.

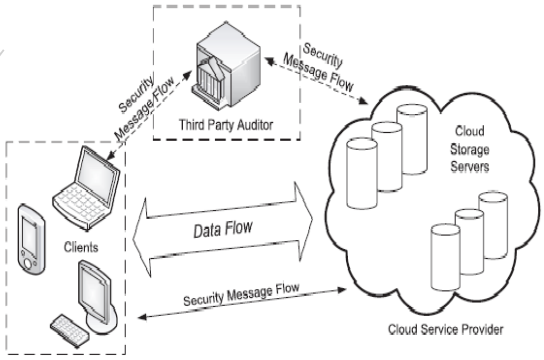

Figure 1. Cloud Data Storage Architecture [18]

2.3 Objectives of Data Integrity

- **Support Dynamic Operation:** Data dynamics means after clients store their data at the remote server, they can dynamically update their data at later times. The main operations supported by data dynamics are data insertion, data modification and data deletion. Moreover, when data is updated the updating overhead should be made as small as possible.
- **Public verifiability:** Public verifiability allows anyone (not just a client) to perform the integrity checking operation.

- **Privacy:** When the verification is performed by a third party verifier (not by a client), the protocol must ensure that no private information contained in the data is leaked. In Batch auditing, multiple delegated auditing tasks from different users can be performed simultaneously by the third party auditor. Cloud server may concurrently handle multiple verification sessions from different client.
- **Unforgeability:** to ensure that no dishonest cloud server that can pass the audit from the user/TPA without indeed keeping users' data intact.
- **Blockless verification**: no challenged file blocks should be retrieved by the verifier during verification process for both efficiency and security concerns.
- **Low Storage Overhead:** The additional storage used for auditing should be as small as possible on both the Auditor and the cloud server.
- **Low Computation:** for checking data integrity low computation require by the auditing scheme.
- **Low Communication:** the amount of communication required by the auditing scheme should be low.
- **Unbounded use of queries**: to ensure that the verifier draws unlimited number of queries in the challenge-response protocol for data verification.

3. Data Integrity Proving Schemes

3.1 Provable Data Possession (PDP)

Definition: A PDP scheme checks that a remote cloud server retains a file, which consists of a collection of n blocks. The data owner processes the data file to generate some metadata to store it locally. The file is then sent to the server, and the owner delete the local copy of the file. The owner verifies the possession of file in a challenge response protocol.

Characteristics of PDP Schemes are:

- In PDP Scheme the client verifies the data stored at a server and still possesses the data without retrieving it.
- In PDP scheme the server does not actually have to access the file blocks, supporting Blockless verification.
- PDP scheme not includes error-correcting codes to address concerns of corruption.
- PDP scheme lends itself more naturally to data that may undergo slight changes as dynamic expansion is supported.

Ateniese et al. [8] 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 Homomorphic Verifiable 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 In their subsequent work [9], Ateniese et al. proposed a dynamic version of the prior PDP scheme problems. 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. In [4], Wang et al. consider the proposed challenge-response protocol can both determine the data correctness and locate possible errors. Similar to [9], they only consider partial support for dynamic data operation. Erway et al. [5] were the first to explore constructions for dynamic provable data possession. They extend the PDP model in [8] to support provable updates to stored data files using rank-based authenticated skip lists. This scheme is essentially a fully dynamic version of the PDP solution. To support updates, especially for block insertion, they eliminate the index information in the "tag" computation in Ateniese's PDP model [8] and employ authenticated skip list data structure to authenticate the tag information of challenged or updated blocks first before the verification procedure. However, it's computational and communication complexity is both up to log (n). Feifei Liu[7] were proposed an improved dynamic model that reduce the computational and communication complexity to constant by using Skip-List, Block, Tag and Hash method. Table 1. Shows comparison different Methods used in the PDP Schemes.

3.2 Proof of Retrievability (PoR)

Definition: In PoR Scheme a cloud server proves to a data owner that a target file is intact, in the sense that the client can retrieve the entire file from the server with high probability. Hence, PoR guarantees not only correct data possession but it also assures retrievability upon some data corruptions.
Characteristics of PoR Scheme are:

- This scheme purposed only works with static data sets.
- It is more suited for storage of large, unchanging data.
- This scheme also includes error-correcting codes to address concerns of corruption.
- It supports only a limited number of queries as a challenge since it deals with a finite number of check blocks (sentinels).

Juels and Kaliski [1] describe a "proof of retrievability" model, where spot-checking and error-correcting codes are used to ensure both "possession" and "retrievability" of data files on archive service systems. Specifically, some special blocks called "sentinels" are randomly embedded into the data file F for detection purpose, and F is further encrypted to protect the positions of these special blocks. The number of queries a client can perform is also a fixed priori, and the introduction of precomputed "sentinels" prevents the development of realizing dynamic data updates. In addition, public auditability is not supported in their scheme. Shacham and Waters [11] design an improved PoR scheme with full proofs of security in the security model defined in [1]. They use publicly verifiable homomorphic authenticators built from BLS signatures [6], based on which the proofs can be aggregated into a small authenticator value, and public retrievability is achieved. Still, the authors only consider static Data files. Table 2. shows comparison of different Methods used in the PoR Scheme.

3.3 Other Auditing Methods

3.3.1 Message Authentication Code (MAC) Method

Assume the outsourced data file F consists of a finite ordered set of blocks m_1; m_2; . . .; m_n. One straightforward way to ensure the data integrity is to precompute MACs for the entire data file. Specifically, before data outsourcing, the data owner precomputes MACs of F with a set of secret keys and stores them locally. During the auditing process, the data owner each time reveals a secret key to the cloud server and asks for a fresh keyed MAC for verification. This approach provides deterministic data integrity assurance straightforwardly as the verification covers all the data blocks. However, the number of verifications allowed to be performed in this solution is limited by the number of secret keys. Once the keys are exhausted, the data owner has to retrieve the entire file of F from the server in order to compute new MACs, which is usually impractical due to the huge communication overhead. Moreover, public auditability is not supported as the private keys are required for verification.

3.3.2 Signature Method

The data owner precomputes the signature of each block and sends both F and the signatures to the cloud server for storage. To verify the correctness of F, the data owner can adopt a spot-checking approach, i.e., requesting a number of randomly selected blocks and their corresponding signatures to be returned.

Notice that the above methods can only support the static data and also a large communication overhead that greatly affects system efficiency. Table 3. shows comparison of different methods used for data integrity.

4. Conclusion

In this survey we observed that data integrity is emerging area in cloud computing for security purpose. Based on the PDP and PoR schemes researcher proposed efficient new schemes.PDP scheme easily support dynamic operation where PoR scheme is not, also PDP Scheme is not include error-correcting code where PoR include it, so significant amount of overhead in the PoR scheme comes from the error-correcting codes which are not present in the PDP scheme. Beside PDP and PoR Schemes other data integrity Schemes are also proposed but many of them are not supported public auditability and unbounded queries which are basic requirements of data integrity verification. Therefore we can say that designing efficient, secure and fully dynamic remote data integrity is still open area of research.

Table 1.

Method Used	Public Auditibility	Static/ Dynamic operation	Unbounded queries
Homomorphic Verifiable Tags[8]	Yes	Static	--
Symmetric Key Cryptography [9]	No	Dynamic*	--
Rank-based Authenticated Skip lists[5]	No†	Dynamic	--
Skip-List, Block, Tag, Hash[7]	--	Dynamic	--
Hybrid Cryptography [15]	Yes	Dynamic	--
Bi-Linear Map and Merkle Hash Tree[10]	Yes	Dynamic	Yes
RSA based storage security[14]	Yes	Dynamic	Yes
Reed-Solomon codes based on Cauchy matrices[2]	Yes	Dynamic	--

* This scheme only supports bounded number of integrity challenges and partially data updates, i.e., data insertion is not supported

† No explicit implementation of public auditability is given for this scheme.

Table 2.

Method Used	Public Auditibility	Static/ Dynamic operation	Unbounded queries
Symmetric-key Cryptography , Error-Coding[1]	No	Static	No
BLS signatures, Pseudorandom Functions (PRFs)[11]	Yes	Static	--
Generate, Encrypt and Append Metadata[21]	No	Static	No
Fragment structure, Random Sampling and index-hash table[24]	Yes	Dynamic	--
Bi-Linear Map,Merkle Hash Tree[19]	Yes	Dynamic	Yes

Table 3.

Method Used	Public Auditibilty	Static/ Dynamic Operation	Unbounded queries
RSA assumption [25]	Yes	Static	--
Bloom Filter[22]	--	Dynamic	--
HLAs and RSA signature [23]	Yes	Dynamic	--
Encryption Algorithm [17]	Yes	--	--

5. References

[1]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.

[2]Bo Chen and Reza Curtmola. "Robust Dynamic Provable Data Possession," 1545-0678/12 $26.00 © 2012 IEEE.

[3]B. Priyadharshini and P. Parvathi, "Data Integrity in Cloud Storage", ISBN: 978-81-909042-2-3 ©2012 IEEE

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

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

[6] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '01), pp. 514-532, 2001.

[7]Feifei Liu, Dawu Gu, Haining Lu," An Improved Dynamic Provable Data Possession Model," 978-1-61284-204-2/11/$26.00 ©2011 IEEE

[8]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, 2007.

[9] 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.

[10]He-Ming.Ruan, Yu-Shian Chen, Chin-Laung Lei, "Data Integrity on Remote Storage for On-line Co-working," 978-0-7695-4776-3/12 $26.00 © 2012 IEEE

[11]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.

[12] Jeff Whitworth, "Proving Retrievability and Possession within the Cloud Storage Model"

[13]Kan Yang · Xiaohua Jia, "Data storage auditing service in cloud computing: challenges, methods and opportunities", DOI 10.1007/s11280-011-0138-0.

[14]M.Venkatesh, M.R.Sumalatha, Mr.C.SelvaKumar, "Improving Public Auditability, Data Possession in Data Storage Security for Cloud Computing," ISBN: 978-1-4673-1601-9/12/$31.00 ©2012 IEEE

[15] Narn - Yih Lee, Yun - Kuan Chang, "Hybrid Provable Data Possession at Untrusted Stores in Cloud Computing," 1521-9097/11 $26.00 © 2011 IEEE

[16] P. Mell and T. Grance, "The NIST definition of cloud computing (draft)," *NIST special publication*, vol. 800, p. 145, 2011.

[17]P.Varalakshmi#1, Hamsavardhini Deventhiran#2, "Integrity Checking for Cloud Environment Using Encryption Algorithm," ISBN: 978-1-4673-1601-9/12/$31.00 ©2012 IEEE.

[18]Qian Wang, Cong Wang, Kui Ren, Wenjing Lou and Jin Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," 1045-9219/11/$26.00 _ 2011 IEEE.

[19]Shu Ni-Na, Zhang Hai-Yan, "On providing integrity for dynamic data based on the third-party verifier in cloud computing," 978-0-7695-4519-6/11 $26.00 © 2011 IEEE.

[20]Solomn Guadie worku, Zhong Ting, Qin Zhi-Guang, "Survey on Cloud Data Integrity Proof Techniques", 978-0-7695-4776-3/12 $26.00 © 2012, IEEE.

[21]Sravan Kumar R, Ashutosh Saxena, "Data Integrity Proofs in Cloud Storage," 978-1-4244-8953-4/11/$26.00 c 2011 IEEE

[22]T. Aditya, P.K. Baruah, R. Mukkamala, "Space-efficient Bloom Filters for Enforcing Integrity of Outsourced Data in Cloud Environments" 978-0-7695-4460-1/11 $26.00 © 2011 IEEE.

[23]Wenjun Luo, Guojing Bai, "ENSURING THE DATA INTEGRITY IN CLOUD DATA STORAGE," 978-1-61284-204-2/11/$26.00 ©2011 IEEE.

[24] Y. Zhu, G.Joon Ahn, Hongxin Hu, Stephen S. Yau, Ho G. An, S.Chen, "Dynamic Audit Services for Outsourced Storages in Clouds," 1939-1374/11/$26.00 © 2011 IEEE.

[25]Zhang lianhong, Chen Hua, "Secuirty Storage in the Cloud Computing: A RSA-based Assumption Data Integrity Check without Original Data," 978-1-4244-8035-7110/$26.00 © 2010 IEEE.