Survey on Data Integrity Checking Protocols in Cloud Computing

P.Parvathi
Sudharsan Engineering College

T.Meyyappan, Ph.D
DCSE, Alagappa University

ABSTRACT

Cloud computing is a internet based computing model that supports convenient ,on-demand and pay-for-use model. In this computing, data owners host their data on cloud servers and clients can access the data from cloud servers. Due to the data outsourcing, efficient verification of the outsourced data becomes a formidable challenge for data security in Cloud Computing (CC). Therefore, an independent auditing service is required to make sure that the data is correctly hosted in the Cloud. Several protocols are introduced for performing integrity in cloud storage. This paper focuses on the different integrity checking protocols to address the above issue. This paper provides an overview of these protocols by presenting their characteristics, functionality, benefits and limitations.

Index Terms

Cloud computing, Data integrity, Data outsourcing

1. INTRODUCTION

Cloud computing has been envisioned as the next generation architecture of IT enterprise, due to its long list of unprecedented advantages in the IT history: On-demand self service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk. As a disruptive technology with profound implications, Cloud computing is transforming the very nature of how business use information technology. One fundamental aspect of this paradigm shifting is that data is being centralized or outsourced into the cloud in a flexible on demand manner brings appealing benefits: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc. While Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards users' outsourced data.

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 failure occasionally, may decide to hide the data errors from the clients for the benefit of their own. How to efficiently verify the correctness of outsourced cloud data without the local copy of data files becomes a big challenge for data storage security in Cloud Computing. Note that simply downloading the data for its integrity verification is not a practical solution due to the expensiveness in I/O cost and transmitting the file across the network. Considering the large size of the outsourced data and the users constrained resource capability, the ability to audit the correctness of the data in a cloud environment can be formidable and expensive for the cloud users. Besides, it is often insufficient to detect the data corruption when accessing the data, as it might be toolate for recover the data loss or damage. In order to slove theproblem of data integrity checking, many schemes are proposed under different systems and security models. In this paper , clearly investigate the requirements should be satisfied for a remote data possession checking protocol to be of practical use , existing auditing methods and its metrics and de metrics.

2. PRINCIPLES OF REMOTE DATA INTEGRITY CHECKING PROTOCOLS

Cloud storage applications offers 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 protocol.

2.1 Overview

A Representative network architecture for cloud data storage is illustrated in fig 1. Three different network entities can be identified as follows:

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

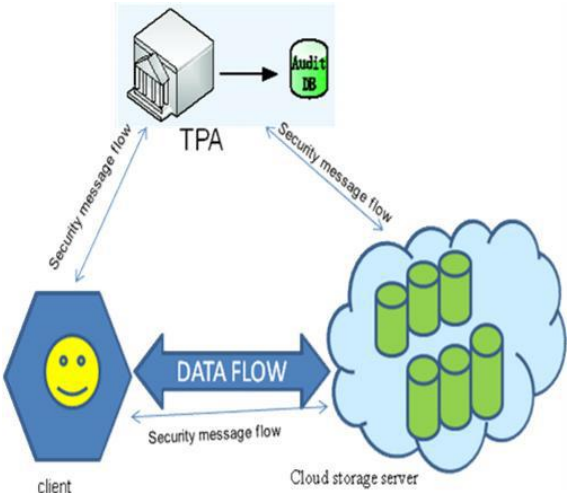

Fig. 1 Cloud data storage architecture

Third party auditor: 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.

3. DATA INTEGRITY PROTOCOLS

3.1 Entire data dependant tag-based protocols

The majority of verification protocols carry on with the idea to associate data with some metadata in the form of authentication tags. These tags are generally derived by the owner from the actual data with some secret, thus allowing both authenticating their origin and proving their integrity.The size of data tag is generally small, and may be therefore kept by the verifier.

Consider simplest proof of retrivability scheme can be made using a keyed hash function hk(F). In this scheme pre computes the cryptographic hash of F using secret key.To check if the integrity of the file F is lost the verifier releases the secret key to the cloud archive and asks to it to compute and return the value of hk(F). Pre-computed results of challenges to be stored at the verifier, where a challenge corresponds to the hashing of the data concatenated with a random number. The protocol requires low storage overhead at the verifier, yet it allows only a fixed number of challenges to be performed.

Inspired from RSA, the verifier keeps the tag T=gd. In each challenge, a nonce N=gr is generated by the verifier and sent to the prover (a nonce is a unique and randomly chosen value).The prover combines the nonce with the data using F to prove the freshness of the answer and obtains R=Nd. The prover's response will be compared by the verifier with a value computed over T. Since the verifier can perform the following
operation:

$$T^r = (g^d)^r = (g^r)^d$$

3.2 Data block dependant tag-based protocols

To reduce the computing time of verification, Sebé et al. in [5] propose to trade off the computing time required at the prover against the storage required at the verifier. The data is split in a number n of blocks {di}1≤i≤n, the verifier holds {Ti=gdi}1≤i≤n and asks the prover to compute a sum function of the data blocks {di}1≤i≤n a challenge N=gr and n random coefficients {ci}1≤i≤n generated from a new seed handed out by the verifier at each challenge. The response is computed as:

$$R=\sum_{i=1}^{n} d_i c_i \bmod N$$

When the verifier receives holder's response, it compares R with:

$$r \qquad \prod_{i=1}^{n} (T_i)^{c_i * r}$$

The index n is the ratio of trade off between the storage kept by the verifier and the computation performed by the prover. It allows an unlimited number of verifications and the maximum running time can be chosen at setup time and traded off against storage at verifier.

The PDP (Provable Data Possession) scheme in [2] improves the probabilistic model by presenting a new form of tags:

$$T_i = (h(v,i)g^{d_i})^{ks} \bmod N$$

where h(.) is a hash function, v a secret random number known only by the owner and the verifier, N an RSA modulus with ks being the owner's signature key, and g a generator of the cyclic group of ZN*. With such homomorphic verifiable tags, any number of tags chosen randomly can be compressed into just one value by far smaller in size than the entire set, which means that communication complexity is independent of the number of indices requested per verification.

Recent studies [6-9] are all data block depended tag based protocols which suggest to guarantee the dynamic operations in cloud storage data without having to transmit the whole data to the client. Most of the works [1][6][7] suggest the public verifaility. Privacy of the data against untrutesd server and third party auditor is achieved by [6].

4 DATA INDEPENDENT TAG-BASED PROTOCOLS

Some approaches consider tags which are not generated from data. The POR protocol (Proof of Retrievability) in [4] explicitly expresses the independency between the data and the metadata used for verification (tags). The protocol is based on verification of sentinels which are random values independent
of the owner's original data. These sentinels are disguised among data blocks. The verification is probabilistic with the number of verification operations allowed being limited to the number of sentinels.

These protocols introduce an extra storage overhead. This overhead can be, however, limited by reusing tags for the storage of different data.

5 DATA REPLICATION-BASED PROTOCOLS

Generally, data replication techniques are of great interest for verification protocols, since they improve the probability of data recovery in case the approach does not detect the destruction of some parts of the stored data. The HAIL (High-Availability and Integrity Layer) protocol in [3] proposed a data replication-based verification protocol. in which the key insight is to embedded MACs in the parity blocks of the dispersal code. As both MACs and parity blocks can be based on universal hash functions. The verifier checks the correctness of a random subset of rows in the encoded matrix. Each server returns a linear combination of
the blocks. To combine server responses, an aggregation code implemented with a Reed-Solomon code is used. The combined response is validated by first decoding and then checking that at least one of the responses of the secondary servers is valid. The main downside of these approaches is that if the parity blocks does not match, it is difficult (depends on the number of the used parity blocks) and computationally expensive to recognize the faulty or dishonest holder.

6. ANALYSIS AND COMPARISION

In this paper concentrate on these above categories to go through their qualitative evaluation and comparison with respect to security and efficiency considerations.

6.1 Security

Remote data is vulnerable to two classes of threats:accidental faults (e.g., caused by a bit error in the storage medium), and voluntary data damage due to holder selfishness. In both cases, destruction or corruption of data stored at a holder should be detected as soon as possible. The main security problem is then the detection of such damage. We distinguish two main categories of verification schemes:
probabilistic and deterministic protocols. The first type of protocols achieves only probabilistic detection of data damage that increases with the iteration of the protocol; whereas data damage detection is complete in the deterministic case.
Protocols with entire data dependant tags (e.g., [6,7,8]) provide generally deterministic guarantees for data damage detection.With probabilistic verification, the assurance on data integrity preservation is increased with the iteration of the verification protocol.
Data dissemination into the system may expose the verification protocol to new attacks. Collusion attacks where by multiple holders collude so that only one of them keeps a data copy that will be used to correctly answer all verifier's challenges directed toward these holders. Preventing this attack may consist on personalizing each copy to its holder, as explained in , or using an erasure coding mechanism for data replication [9][10]. Even without a collusion attack, the verification protocols based on the replication technique are still vulnerable to replay attacks where one of the holders may derive or retransmit a response captured from another holder's response message. This attack can be hampered by using common authentication and encryption mechanisms during the verification process.

6.2 Efficiency

The costs of verifying the proper storage of some data should be considered for the two parties that take part in the verification process, namely the verifier and the holder.

Communication overhead is significantly low for the majorityof the presented protocols, notably the more cryptographically advanced ones (using for instance bilinear maps [7]), or replication based schemes (e.g., [3]) since challenges and responses are aggregated between all replica holders.

Storage overhead . The verifier must store a meta information that makes it possible to generate a time-variant challenge based on the proof of knowledge protocol mentioned above for the verification of the stored data. The size of this meta information must be reduced as much as possible even though the data being verified is very large. The effectiveness of storage at the holder must also be optimized. The holder
should store the minimum extra information along with the data. Computation complexity at the holder side is the main drain of protocols using entire data-dependant tags. These protocols are generally based on expensive functions that are applied to the whole data. To overcome this problem, improved versions (e.g., [5]) have been proposed whereby the data is split into multiple blocks from which tags are produced. With this technique, computation complexity at the holder is reduced; on the other hand, the holder or the verifier should store more tags.

7. CONCLUSION

In this survey, we analyzed a large list of protocols for remote data integrity verification and compared them. The majority of the existing verification protocols are built with data integrity verification as primary objective. Other security primitives like guarantying data confidentiality or owner privacy[6] addressed . Moreover most of the protocols support dynamic operations in cloud storage. Designing remote dataintegrity verification is still a hot topic in the research community and further performance improvement and practical extensions to suit a large spectrum of storage applications are underway.

8. REFERENCES

[1] Ateniese G, Di Pietro R, Mancini LV, Tsudik G (2008) Scalable and efficient provable data possession. In Proceedings of the 4th international conference on security and privacy in communication networks (SecureComm'08), 1–10, 2008.

[2] Ateniese G, Burns R, Curtmola R, Herring J, Kissner L, Peterson Z, Song D (2007) Provable data possession at untrusted stores. In Proceedings of the 14th ACM conference on computer and communications security, ACM, 2007, 598–609.

[3] Bowers KD, Juels A, Oprea A (2009) HAIL: a high-availability and integrity layer for cloud storage. 16th ACM Conference on Computer and Communications Security CCS, November 9–13, 2009.

[4] Juels A, Kaliski BS (2007) PORs: proofs of retrievability for large files. Cryptology ePrint archive, June 2007. Report 2007/243.

[5] Sebé F, Domingo-Ferrer J, Martínez-Ballesté A, Deswarte Y, Quisquater J-J (2007) Efficient remote data possession checking in critical information infrastructures. IEEE Trans Knowl Data Eng 20:1034–1038, Aug 2008. ISSN:1041-4347.

[6] Wang C, Wang Q, Ren K, Lou W (2010) Privacy-preserving public auditing for data storage security in cloud computing. In Proceedings of the 29th conference on information communications, San Diego, California, USA,525–533, March 14–19, 2010.

[7] Wang Q, Wang C, Li J, Ren K, Lou W (2009) Enabling public verifiability and data dynamics for storage security in cloud computing. 14th European Symposium on Research in Computer Security (ESORICS 2009), Saint Malo,France, pp. 355–70, September 21–25, 2009.

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

[9] 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.

[10] C. Wang, Q. Wang, K. Ren, and W. Lou,"Towards Secure and Dependable Storage Service in Cloud Computing" 2010.