

Data Integrity Checking And Access Confidentiality In Multicloud Storage

B.Pavithra¹ and C.S.Anita²

¹M.E(CSE),R.M.D Engineering College,
Kavaraipettai,chennai-601206 Tamil Nadu
paviasha93@gmail.com

²Associate professor of CSE department,
R.M.D Engineering College,
Kavaraipettai,chennai-601206 Tamil Nadu
csa.cse@rmd.ac.in

January 3, 2018

Abstract

In cloud computing to ensure data integrity on user uploaded data in multicloud maintaining security on cloud data and also provide dynamic reallocation of data at every access. The foundation of cloud computing lie in the outsourcing of computing task to the third-party. It entails the security risk in terms of confidentiality, integrity and availability of data and service. Cloud server allocates user files to verifier for data integrity checking. User privacy will be affected because of providing user file to verifier. User data always stored in static location of cloud server. In proposed model remote data integrity checking is required to secure users data in multicloud environment. Cloud server is managed by cloud service provider it has significant storage space and computation resource to maintain the clients data. Introducing the File allocation table(FAT) File System has proper indexing and metadatas(MD5) for the different chunks of the cloud storage and also gets data encoded using Base 64 Algorithm. Cloud provides random blocks to

verifier for integrity checking which is to protect user privacy from third party. File recovery is done by the verifier automatically if the data gets corrupted during checking. At every access of the file by the user, blocks of the data will be dynamically reallocated between the cloud servers. This achieves access confidentiality in cloud computing.

Key Words : Cloud computing, provable data possession, identity-based cryptography, distributed computing, bilinear pairings

1 Introduction

Cloud computing is a new computing paradigm connect to users elastically utilize a shared pool of cloud resources like processors, storages, applications, services are on-demand fashion applications. It takes the information to process service, such as storage, computing. It attracts more intention from the enterprises. The foundations of cloud computing lies in outsourcing tasks to the third party and so it entails the security risks in terms of confidentiality, integrity and availability with data and service. The issue to convince the cloud clients that their data are kept intact is especially vital in the clients do not store these data locally. Remote data integrity checking is a primitive to address this issue. When client stores his data on multicloud servers, distributed storage and integrity checking are indispensable.

The integrity checking protocol should be efficient in order to make it suitable for capacity-limited end devices and it is based on distributed computation. In PKI(Public Key Infrastructure), provable data possession protocol needs public key certificates, distributions and management. It meets with considerable overheads since the verifier checks certificates like the remote data integrity. In addition to the heavy certificate verification, the system suffers from the other complicated certificate management such as certificate generation, delivery, revocation, renewals. In cloud computing most verifiers have low computation capacity.

The identity-based public key cryptography can eliminate the complicated certificate management in order to increase efficiency the identity based provable data possession is more attractive. It

focuses on distributed provable data possession in multi cloud storage. These protocol can be made efficient by eliminating the certificate management. We introduce the new remote data integrity checking: ID-DPDP. In these model, the system and security are formally proposed. It is based on the bilinear pairings, the concrete ID-DPDP protocol is designed. These protocol is provably secure in the random oracle model. Based on the clients authorization, the proposed ID-DPDP protocol can realize private verification, delegated verification and public verification.

In this paper we propose an remote data integrity checking is required to secure users data in multicloud environment. The main contributions are as follows:

- The main aim of this paper is to ensure data integrity on user uploaded data in multicloud maintaining security on cloud data and also provide dynamic reallocation of data at every access.
- We introduce the five algorithms namely: Base 64 Algorithm, Message Digest(MD5), Dynamic block generation, Sting Matching, Verifiable Data Integrity checking.
- The enhancement of this paper is to get the file recovery, Access confidentiality in Dynamic reallocation of blocks at every access, Privacy preservation on public auditing, Dynamic Block Split.

The remainder of this paper is organized as follows. In Section 2 contains the literature review. In section 3, the system model is formalized. Finally we conclude this paper in section 4.

2 Related works

2.1 Provable Data Possession at Untrusted Store

G.Ateniese proposed a model for Provable Data Possession(PDP)^[4] is to allows a client that has stored data at an untrusted server to verify that the server control the original data without retrieving it. This model produces the probabilistic proofs of control by sampling random sets of blocks from the server, and it can reduces the I/O

costs. The client maintains the constant amount of metadata to verify the proof. The PDP model for remote data checking supports a large data sets in widely-distributed storage systems. This model is unique in that it allows the server and it access the small portions of the file in generating the proof. All other techniques must access the entire file. In this model they give the first provably-secure scheme for the remote data checking and also it uses bandwidth.

The second Efficient PDP(E-PDP) scheme proves data possession using a single modular exponentiation at the server, even though it provides a weaker guarantee. This scheme is the fundamental construct underlying an archival introspection system they are developing for the long-term preservation of astronomy data. E-PDP schemes will ensure that the computational requirements of remote data checking do not burden the remote storage sites. Both schemes use homomorphic verifiable tags(HVT). The concept of the homomorphic verifiable tag that will be used as a building block for the PDP schemes. HVT following the properties are blockless verification and homomorphic tags. Because of the homomorphic property the tags can computed for a multiple file blocks and that can be combined into a single value, The client pre-computes etags for each block of a file and then stores the file and its tags with the server. At a later time, the client can verify that the server control of the file by generating a random challenge against a randomly selected set of file blocks. Using the queried blocks and their corresponding tags and server generates a proof of possession.

Construction of provable data possession at untrusted stores has the following features:

- The PDP protocol checks that an outsourced storage site retains a file, which consist of the collection of n blocks.
- The client pre-processing of the file that generating a piece of metadata that is stored locally and it transmits the file to the server and it may delete its local copy.
- The server can stores the file and respond to the challenges issued by the client. As a part of pre-processing, the client may alter the file to be stored at the server.
- The client may expand the file or additional metadata to be stored at the server. Before deleting its local copy of the

file and the client may execute a data possession challenge to make sure the server has successfully stored the file.

- The Clients may encrypt a file prior to outsourcing the storage. For our purposes encryption is an orthogonal issue so the file may consist of encrypted data and our metadata does not include encryption keys.
- The important performance parameters of a PDP scheme includes the computation complexity, block access complexity, communication complexity.

Challenges

- By verifying authenticity of data as a critical issue in storing data on untrusted servers. It arises in peer-to-peer storage systems, network file systems, long-term archive, web-service object store and database systems.
- Provable data possession, in which it desires to minimize the file block accesses, the computation on the server and the client-server communication. The solutions for PDP is to fit this model they incur a low overhead at the server and require a small, constant amount of communication per challenge.
- The model generates probabilistic proofs of possession by sampling the random sets of blocks from server, which drastically reduces I/O costs.
- Previous schemes that do not allow samplings are not practical when PDP are used to prove the possession of large amounts of data. These schemes also impose a significant I/O and computational burden on the sever.

2.2 Efficient Remote Data Possession Checking in Critical Information Infrastructures

F.Sebe have proposed the protection of critical infrastructures is the priority for governments and companies. The development of the information society has caused the vast majority of such infrastructures to critically depending upon the correct operation of an

underlying information systems that is why such information systems are often referred to the critical information infrastructures. The checking data control in networked information systems such as related to critical infrastructures like power facilities, airports, data vaults, defense systems is a matter of crucial importance. The Remote data possession checking protocols can permit to checking that a remote server can access an uncorrupted file and the verifier does not need to know beforehand the entire file is being verified or not. The data vaulting systems are increasingly used to store off-site copies or backups of critical data. If the backup data stored in the data vault become it can corrupted without knowledge of the data owner, no backup recovery is possible in case any loss of primary critical data. When using the remote data control checking protocol, a data vault customer might be able to periodically verifying the data vault provider is storing a current and complete backup of the critical files. Any corruption will be noticed by data owner who will be able to take immediate action. The remote data control checking is an important component of the intrusion detection systems(IDS) and it is used to detect the server corruption. The remote data possession checking protocol can follows as

- The server can backup the original files and it access them properly to run the protocol while using the corrupted versions to provide the service.
- In Dependable Intrusion Tolerance[DIT] architecture integrity checking is one of the building blocks used to detect corruption of remote data.

The following requirements to be satisfied for the remote data possession checking protocol to be a practical use:

- The verifier should not be required to keep the entire copy of the files to be checked. It would be impractical for the verifier is to be replicate the content of all the provers to be verified.
- Even if the prover is malicious then the protocol has to be stay secure.
- The amount of communication is required by the protocol should be low.

- The protocol should be an efficient in terms of the computation.
- It ought to be possible to run the the verification an unlimited number of times.

Challenges

- The new remote data possession checking protocol is to allows an unlimited number of file integrity verifications and the maximum running time can be chosen at set-up time and traded off against storage at the verifier.
- The proposal work follows the ideas of the Diffie-Hellman-based approach in that it seeks to permit an arbitrary number of verifications.
- The clear advantage of proposal is that it satisfies the protocol should be efficient in tems of computation, because the computing time of verification can be reduced at the set-up stage.
- The generalization of this protocol is to checking the integrity of set of files in a single verification and to do this, an ordering or structure between the set of files can be regarded as a superfile.
- Once upon the superfile is defined, its integrity can be checked using this protocol without any modification.

2.3 Scalable and Efficient Provable Data Possession

G.Ateniese and R.Dipietro have presented the PDP scheme as relies only on efficient symmetric key operations in both setup and verification phases. The effective PDP technique must be equally applicable to malicious and unreliable servers. The problem is complicated by the fact that the client might be a small device with limited CPU, battery power, communication facilities and its need to minimize bandwidth, local computation overhead for the client in performing each verification. PDP is also sometimes referred to

as proof of data retrievability(POR)^[5]. POR has popped up in the research literature. PDP scheme is more efficient than the POR as it requires no bulk encryption of outsourced data and no data expansion due to additional sentinel blocks.

The PDP technique is to allow an outsourcing of dynamic data and it efficiently supports operations, such as block modification, deletion and append. The features of outsourcing include reduced costs from savings in storage, maintenance and an increased availability and transparent keep-up of data. The PDP is a public key based technique it allows any verifier to query the server and it obtains the interactive proof of data possession. The interaction can be repeated at any number of times and each time resulting in a fresh proof. During the verification phase, the clients ask for randomly picked sentinels and it checks whether they are intact. Whether the server modifies parts of the data, then the sentinels would also be affected with a certain probability.

Challenges

- The issue is how to efficiently, frequently and securely verify that the server is faithfully storing its clients outsourced data. The storage server can be assumed to be an untrusted in terms of the security and reliability.
- The problem can be exacerbated by the client being a small computing device with the limited resources. The Prior work has addressed to this problem using public key cryptography or requiring the client to outsource data in the encrypted form.
- The new scheme supports secure and efficient dynamic operations on outsourced data blocks including modification, deletion and append.
- The features of these scheme is very low cost and support for dynamic outsourced data make it attractive for realistic applications.

2.4 Dynamic provable data possession

C.C. Erway have presented storage outsourcing services and resource sharing networks have become popular, the problem is efficiently proving the data stored at untrusted servers^[1] has received increased attention. Data is preprocessed by the client and meta-data and it is used for verification purposes can produced in the PDP model. The file is sent to an untrusted server for storage, and the local copy of the file can be deleted by the client. The client keeps some information to check servers response later. The original PDP scheme applies only to static files. Outsourcing the storage of very large files to remote servers presents an additional constraints the client should not download all stored data in order to validate and this may be prohibitive in terms bandwidth and time especially if the client performs this check frequently. The static model fits some application scenarios such as libraries and scientific datasets and it is crucial to consider the dynamic case, modifying stored blocks while maintaining data possession guarantees.

The Dynamic PDP scheme is an essential in practical cloud computing systems for file storage, database services and peer-to-peer storage. The PDP schemes based on the authenticated skip list(DPDP I) and RSA trees(DPDP II). The performance of this schemes denote with the number of blocks. The server computation, i.e., the time taken by the server to process an update or compute a proof for a block is for(DPDP I) and (DPDP II). The time taken by the client to verify a proof returned by the server is for both schemes and the communication complexity and the size of the meta-data stored locally by the client is for both schemes and the probability of detecting server misbehavior is for both schemes are fixed logarithmic communication complexity, where the file is the ratio of corrupted blocks and constant that is independent of n .

In this paper they provide a definitional framework and efficient constructions for dynamic provable data possession(DPDP) as follows:

- In the DPDP, which extends the PDP model to support provable updates on the stored data. Given a fileconsisting of n blocks, they define an updates as either insertion of a new block or modification of any block.

- Update operation describes the most general form of modifications on a client and it may wish to perform on a file.
- DPDP solution is based upon a new variant of authenticated dictionaries, where we use rank information to organize dictionary entries.
- Thus, it can be able to support efficient authenticated operations on files at the block level such as authenticated insert and delete. They prove the security of constructions using standard assumptions.

Challenges

- They present a rank-base authenticated dictionary built over a skip list. This construction yields a DPDP scheme with logarithmic computation, communication and the same detection probability as the original PDP scheme.
- They give an alternative construction of a rank-based authenticated dictionary using an RSA tree. This construction results in the DPDP scheme with improved detection probability but higher server computation.

2.5 Cooperative Schedule Data Possession for Integrity Verification in Multi-Cloud Storage

Y. Zhu have presented a cloud computing environment is constructed based upon the open architectures and interfaces. The multiple internal and external cloud services can provide a high interoperability. The muticloud that allows a clients to easily access his/her resources and web services provided by amazon EC2. In multicloud there exist various tools and technologies such as platform VM Orchestrator, VMware vSphere and ovirt. The distributed cloud storage platform is constructed by help of these tools. The important platform is vulnerable to security attacks so it would bring the irretrievable losses to the clients.

Provable data possession is a technique to ensuring the data integrity in storage outsourcing. The scalability of service and the

data migration is supported by the distributed cloud storage for the efficient PDP scheme. The multiple cloud service providers is to co-operatively store and maintain the clients data. They proposed a lightweight PDP scheme based on cryptographic hash function and symmetric key encryption^[2]. The servers can deceive the owners by using previous metadata. Introduce a cooperative PDP schemes adopting zero-knowledge property and three-layered index hierarchy, respectively. The efficient method selected for the optimal number of sectors in each block to minimize the computation costs of clients and storage service providers. CPDP scheme without compromising data privacy based on modern cryptographic techniques.

- PDP or POR^[3] is such a probabilistic proof technique for storage provider to prove the integrity of clients data without downloading data.
- The proof-checking is an especially important for large-size files folders to check whether these data have been deleted without downloading the latest version of data.
- This scheme is able to replace traditional hash functions and signature in storage outsourcing.

Challenges

- Present a cooperative scheme based on ^[1]homomorphic verifiable response and hash index hierarchy. They prove the security of their scheme based on multi-prover zero-knowledge proof system, which can satisfy completeness, soundness and zero-knowledge properties.
- In addition, they can articulate performance optimization mechanisms are based on these scheme and it can present an efficient method for selecting optimal parameter values to minimize the computation costs of client and storage service providers.
- The experiments show that our solution introduces lower computation and communication.

- The number of updates and challenges are limited and it is fixed in advance and the users cannot perform block insertions anywhere.

3 System model

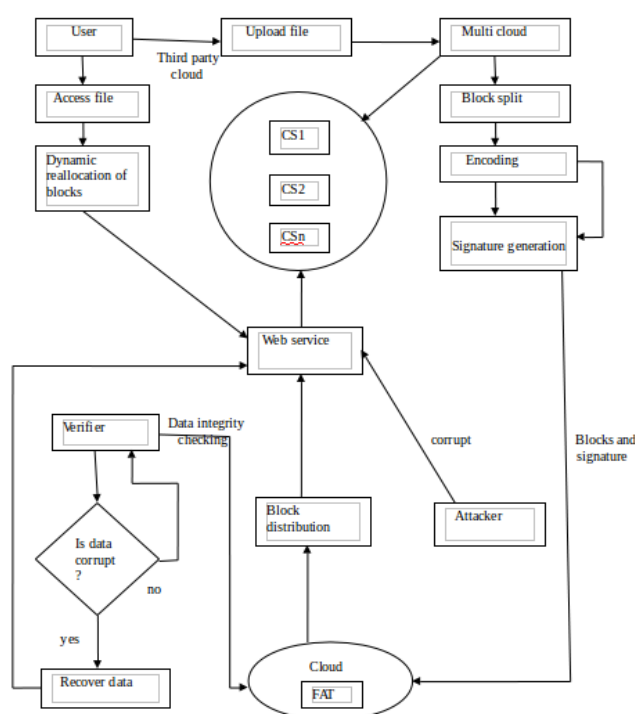


Figure 3.1 : System model of proposed architecture diagram

User: It has an initial level registration process at the web end. The users provide their own personal information for this process. The server in turn stores the information in its database. After registration, user can upload files to the server. Upload files will be stored in a server. Then split the whole file F into n blocks. The client prepares to store the block in the multiple cloud servers.

Cloud server: It is used for data storage has a separate storage area that will be formatted with FATFS (File Allocation Table

File System). When the user uploads the data to different cloud by the time it is splitted into different chunks and each chunk will be appended with signatures before storing the data in FATFS. Data is divided into many blocks, where each block is independently signed by the user. Also the data gets encoded using for Base64 algorithm. FATFS has proper indexing and metadatas for the different chunks of the data that is being uploaded by user.

Attacker: If attacker corrupts data in multicloud, the continuous auditing process helps the verifier to perform block level and file level checking.

Verifier: It performs integrity checking on cloud data. Cloud allocates random combination of all the blocks to the Verifier, instead of the whole file is retrieved during integrity checking. This is to protect user privacy from a third-party (verifier). File recovery is done by the verifier.

4 Conclusion

In this paper, overview of future work of Data integrity checking and access confidentiality in multicloud storage is presented. System design of remote data integrity checking is done by the verifier effectively for maintaining security and privacy in multicloud is formalized. Survey of various techniques are used to secure the clients data in cloud server is taken and their challenges are analyzed. Future direction is to implement the proposed work and analyze the performance of data integrity checking and access confidentiality in multicloud storage.

References

- [1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, *Provable Data Possession at Untrusted Stores*, in Proc. CCS, 2007, pp. 598-609.
- [2] F. Sebe, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J. Quisquater, *Efficient Remote Data Integrity Checking in Critical Information Infrastructures*, IEEE Trans. Knowl. Data Eng., vol. 20, no.8, pp. 1034-1038, Aug. 2008.

- [3] G. Ateniese, R. Dipietro, L.V. Mancini, and G. Tsudil, *Scalable and Efficient Provable Data Possession*, in Proc. SecureComm, 2008, pp. 1-10.
- [4] C.C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, *Dynamic Provable Data Possession*, in Proc. CCS, 2009, pp. 213-222.
- [5] Y. Zhu, H. Hu, G.J. Ahn, and M. Yu, *Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage*, IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231-2244, Dec. 2012.
- [6] H.Q.Wang.(2013,Oct / Dec). *Proxy Provable Data Possession In Public Clouds*. IEEE Trans. Serv. Comput. [online]. 6(4), pp.551-559. Available: <http://doi.ieeecomputersociety.org/10.1109/TSC.2012.35>.
- [7] Y.Zhu, H.Wang , Z.Hu, G.J. Ahn, H.Hu and S.S. Yau, *Efficient Provable Data Possession For Hybrid Clouds*, in proc. CCS, 2010, pp.756-758.
- [8] R.Curtmola, O.Khan, R.Burns and G.Ateniese, *MR-PDP: Multiple-Replica Provable Data Possession*, in proc. ICDCS, 2008, pp. 411-420.
- [9] A.F. Barsoum and M.A.Hasan, *Provable Possession And Replication Of Data Over Cloud Servers*, Centre Appl. Cryptogr. Res., Univ. Waterloo, Waterloo, ON, Canada, Rep. 2010/32. [online]. Available: <http://www.cacr.math.uwaterloo.ca/techreports/2010/cacr2010-32.pdf>.
- [10] Z.Hao and N.Yu, *A Multiple-Replica Remote Data Possession Checking Protocol with Public Verifiability*, in proc. 2nd Int.Symp. Data, Privacy, E-Comm., 2010, pp. 84-89.

