ISSN:2229-6093

Krunal N Chandewar et al , Int.J.Computer Technology & Applications,Vol 5 (1),119-122

Framework of Data Integrity Verification for Multi Clouds Using CPDP Scheme

Krunal N. Chandewar,Shailesh T. Khandare
Master of Engg. in Computer Science & Engg., BNCOE,Pusad,Sgbau Amravati University,India
krual_chandewar@rediffmail.com, khandare.shailesh@rediffmail.coml

Abstract

To provide the security first PDP scheme is proposed but due to I/O burden on the cloud. Secondly SPDP scheme is proposed but it is unsuitable for third party verification. Then DPDP scheme is proposed but it takes very large time in integrity verification. After then CPOR Scheme is proposed the Limitation of this work is a Lack of some security issues for large files Lastly CPDP scheme is proposed for integrity verification Cloud storage has become an attractive and cost effective alternative for enterprises to outsource their valuable business data. However, there are security concerns pertaining to the integrity of data as the cloud server is treated as "untrusted". To overcome this problem many security schemes came into existence. Recently Zhu et al. presented a Scheme known as Cooperative Provable Data Possession (CPDP) for data integrity in cloud.

1. Introduction

A Now a days, number of enterprises, organizations and even general users are creating a large amount of electronic data. In order to lessen the heavy burden of local data storage and maintenance, data is being outsourced into Therefore, how to backup data files in the user not the case, found an efficient and securely ways of good information to perform periodically verification, allowing users to know his information file is stored securely on the server, this data storage is cloud computing environment is an important security issue. Cloud computing has become popular in recent years. There are many cloud providers such as Amazon, IBM, Microsoft, and Oracle and so on. These cloud service provider help users to outsource their data to cloud. The cloud environment is built based on standards for interoperability. Thus multi-cloud environments in distributed architecture came into existence.

They are also known as hybrid clouds. Virtual Infrastructure Management (VIM) is one the features of these clouds. Amazon EC2 web services are an example for such distributed environment. There are many technologies or tools for multi-cloud. They include Ovirt, vSphere, VMware, and Platform VM Orchestrator. Distributed cloud storage platforms can be built using these tools. Thus is possible for the cloud to support data outsourcing facilities to users. However, there are security concerns regarding this concept as they cloud servers are treated by the users as untrusted. The valuable business data when stored in cloud server, without a local copy, there is security concern as the cloud storage providers do not take or data security generally or they have limitations in providing security. Therefore the data stored in cloud may be vulnerable to attacks thus causing irreversible losses to clients. This is the problem to be addressed. Many schemes came into existence to address this problem. Provable Data Possession (PDP) is one such scheme proposed in. This scheme ensures that the data integrity is not lost. However, this scheme needs the users to download data for verification which causes security problem again. Therefore it is essential to have a scheme where data downloading is not required for verification. Towards this end PDP scheme such as Scalable PDP and Dynamic PDP came into existence. These schemes focused on single cloud storage providers. There are schemes like SPDP, DPDP and Merkle Hash Tree (MHT) make use of authenticated skip list in order to verify the adjacent blocks for integrity. These schemes do not work in multi-cloud environments as they can't construct MHT for such environment. The other schemes such as CPOR and PDP make use of homomorphic verification tags where downloading data for verification is not required. To overcome the drawbacks of all existing methods, it is essential to build a cooperative PDP scheme that works in distributed multi-cloud environment. Data Leakage

ISSN:2229-6093

Attack and Tag Forgery Attack are the attacks to be prevented with new PDP scheme.

2. Motivation

To provide security first PDP scheme is proposed but due to I/O burden on the cloud. Secondly SPDP scheme is proposed it removes the limitation of I/O burden of first scheme but this sceme is unsuitable for third party verification. Then DPDP scheme is proposed but it takes very large time in integrity verification. After then CPOR Scheme is proposed the Limitation of this work is a Lack of some security issues for large files Lastly CPDP scheme is proposed which removes all the flaws of all the scheme for integrity verification One of the most important and most attention issues, that is in the cloud environment, servers within the data storage with security and integrity verification in this, we give an overview of our motivation in constructing CPDP. Our motivation is based on the following challenging questions that need to be addressed, which help us define our objectives in this paper.

3. Contributions

The main contributions of this work are summarized as follows:

1. We introduce a formal framework for provable data possession (PDP)
2. Then we provide the first efficient fully Scalable PDP solution
3. After we present a DPDP scheme with computation and communication.
4. We give an detailed construction of CPOR Scheme
5. Lastly we discuss all about our proposed Scheme which is nothing but the CPDP

4. Related Work

G Ateniese et.al.[1] proposed a Provable Data Possession Scheme (PDP) a PDP scheme is a collection of four algorithms (KeyGen,TagBlock,GenProof, CheckProof) We introduce a 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. Archival network storage presents unique performance demands. Given that file data are large and are stored at remote sites, accessing an entire file is expensive in I/O costs to the storage server and in transmitting the file across a network. Reading an entire archive, even periodically, greatly limits the scalability of network stores. Furthermore, I/O incurred to establish data possession interferes with on-demand bandwidth to store and retrieve data. We conclude that clients need to be able to verify that a server has retained file data without retrieving the data from the server and without having the server access the entire file.

We define a model for provable data possession (PDP) that provides probabilistic proof that a third party stores a file. The model is unique in that it allows the server to access small portions of the file in generating the proof; all other techniques must access the entire file. Within this model, we give the first provably-secure scheme for remote data checking. The client stores a small O(1) amount of metadata to verify the server's proof. Also, the scheme uses O(1) bandwidth1. The challenge and the response are each slightly more than 1 Kilobit. We also present a more efficient version of this scheme that proves data possession using a single modular exponentiation at the server, even though it provides a weaker guarantee. Such schemes also impose a significant I/O and computational burden on the server. Which is to be solved out in our next shame named SPDP

R D Pietro et.al.[2] says Scalable PDP first give our provable data possession scheme that supports sampling. In the Setup phase, the client computes a homomorphic verifiable tag (Ti,mi , Wi) for each block mi of the file. In order to maintain constant storage, the client generates the random values Wi by concatenating the index i to a secret value v; thus, TagBlock has an extra parameter, i. Each value Ti,mi includes information about the index i of the block mi, in the form of a value h(Wi) This binds the tag on a block to that specific block, and prevents using the tag to obtain a proof for a different block. These tags are stored on the server together with the file F. The extra storage at the server is the price to pay for allowing thin clients that only store a small, constant amount of data, regardless of the file size.

In the Challenge phase, the client asks the server for proof of possession of c file blocks whose indices are randomly chosen using a pseudo-random permutation keyed with a fresh randomly- chosen key for each challenge. This prevents the server from anticipating which blocks will be queried in each challenge. C also generates a fresh (random) challenge gs = gs to ensure that S does not reuse any values from a previous Challenge phase. The server returns a proof

ISSN:2229-6093

of possession that consists of two values: T and ρ. T is obtained by combining into a single value the individual tags Ti,mi corresponding to each of the requested blocks. ρ is obtained by raising the challenge gs to a function of the requested blocks. The value T contains information about the indices of the blocks requested by the client (in the form of the h(Wi) values). C can remove all the h(Wi) values from T because it has both the key for the pseudo-random permutation (used to determine the indices of the requested blocks) and the secret value v (used to generate the values Wi). C can then verify the validity of the server's proof by checking if a certain relation holds between T and ρ. Secondly we propose the SPDP scheme but it is unsuitable for third party verification. So we next proposed DPDP which solves this problem.

C. C. Erway et.al.[3] says Our DPDP solution is based on a new variant of authenticated dictionaries, where we use rank information to organize dictionary entries. Thus we are able to support efficient authenticated operations on files at the block level, such as authenticated insert and delete. We prove the security of our constructions using standard assumptions. We also show how to extend our construction to support data possession guarantees of a hierarchical file system as well as file data itself. To the best of our knowledge, this is the first construction of a provable storage system that enables efficient proofs of a whole file system, enabling verification at different levels for different users (e.g., every user can verify her own home directory) and at the same time not having to download the whole data. Our scheme yields a provable outsourced versioning system, which is evaluated in our discussion by using traces of CVS repositories of three well-known projects.

Finally, our work is closely related to memory checking, for which lower bounds are presented. Specifically, it is proved that all non-adaptive and deterministic checkers have read and write query complexity summing up to (log n/ log log n) (necessary for sub linear client storage), justifying the O(log n) cost in our scheme. Note that for schemes based on cryptographic hashing, an (log n) lower bound on the proof size has been shown. But the limitation of this scheme is it takes very large time in integrity verification which is to be solved in next proposed sheme.

A Juels et.al.[4] says due to the expensiveness of communication, especially, for large-size files. Recently, several PDP schemes are proposed to address this issue. In fact, PDP is essentially an interactive proof between a CSP and a client because the client makes a false/true decision for data possession without downloading data. Existing PDP schemes mainly focus on integrity verification issues at untrusted stores in public clouds, but these schemes are not suitable for a hybrid cloud environment since they were originally constructed based on a two-party interactive proof system. For a hybrid cloud, these schemes can only be used in a trivial way: clients must invoke them repeatedly to check the integrity of data stored in each single cloud. The traditional cryptographic technologies for data integrity and availability, based on Hash functions and signature schemes cannot support the outsourced data without a local copy of data. It is evidently impractical for a cloud storage service to download the whole data for data validation after this Scheme is proposed the Limitation of this work is a Lack of some security issues for large files which is to be solved in our CPDP scheme.

Y Zhu et.al.[5] says, we address the problem of provable data possession in distributed cloud environments from the following aspects high security, transparent verification, and high performance. To achieve these goals, we first propose a verification framework for multi-cloud storage along with two fundamental techniques hash index hierarchy (HIH) and homomorphic verifiable response (HVR).

We then demonstrate that the possibility of constructing a cooperative PDP (CPDP) scheme without compromising data privacy based on modern cryptographic techniques, such as interactive proof system (IPS). We further introduce an effective construction of CPDP scheme using above-mentioned structure. Moreover, we give a security analysis of our CPDP scheme from the IPS model. We prove that this construction is a multi-prover zero-knowledge proof system (MP-ZKPS), which has completeness, knowledge soundness, and zero-knowledge properties. These properties ensure that CPDP scheme can implement the security against data leakage attack and tag forgery attack. In order to prove the integrity of data stored in a multi-cloud environment, we define a framework for CPDP based on interactive proof system (IPS) and multi-prover zero-knowledge proof system (MPZKPS), as follows: Definition (Cooperative-PDP): A cooperative provable data possession $\mathcal{S}$ = ($KeyGen$, $T\ agGen$, $Proof$) is a collection of two algorithms ($KeyGen$, $TagGen$) and an interactive proof system

5. Conclusions

The limitation of all derived scheme are deprecated by our proposed cooperative PDP scheme in this scheme, we presented the construction of an efficient PDP scheme for distributed cloud storage.
Based on homomorphic verifiable response and hash index hierarchy, we have proposed a cooperative PDP

ISSN:2229-6093

scheme to support dynamic scalability on multiple storage servers. We also showed that our scheme provided all security properties required by zero knowledge interactive proof system, so that it can resist various attacks even if it is deployed as a public audit service in clouds by using this construction we are Deprecating all the limitation which is to be found in previously derived scheme.

6. References

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner "Provable Data Possession at Untrusted Stores", 14th ACM Conference on Computer and Communications Security,2007

[2] R.D. Pietro,L.V. Mancin, G. Ateniese,",Scalable and Efficient Provable Data Possession",2008

[3] C.C. Erway, A. Kupcu, C. Papamanthou, R. Tamassia,"Dynamic Provable Data Possession",November 29, 2009

[4] A. Juels, B. S. Jr.,"Pors: proofs of retrievability for large files", ACM Conference on Computer and Communications Security, 2007, pp. 584–597.

[5] Y. Zhu, H. Hu, G. Ahn,"Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage" ,IEEE Transactions On Parallel And Distributed Systems,Digital Object Indentifier 10.1109/TPDS 2012.66 April 2012.

.