International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-6, September 2015
ISSN: 2395-3470
www.ijseas.com

Implement PDP Technique of Verifying Integrity in Cloud Computing

Chaitali Kharbade[1], Vishwajit Bajpai[2] , Shyam P. Dubey[3]
[1] Nuva College of Engineering & Technology, Nagpur, India
[2] MIET, Gondia, India
[3] Nuva College of Engineering & Technology, Nagpur, India

Abstract

Cloud computing is a subscription-based service where you can obtain networked storage space and computer resources. We propose a cloud-based storage scheme that allows the data owner to benefit from the facilities offered by the cloud service providers (CSPs) through Storage-as-a-Service (SaaS) and enables indirect mutual trust between owner and CSP.

Storage as a Service is a paid facility that enables organizations to outsource their data to be stored on remote servers and perform full block level dynamic operations and hence reduces the maintenance cost and mitigates the burden of large local data storage at the organization's end. The outsourced stored data can be accessed by a group of authorized users by the data owner. The owner has the privilege to grant or revoke access of the stored data in the cloud. The present system is providing a good security mechanism for stored data and proper sharing of keys among authorized users, and data owner for the cryptographic mechanism. It also ensures the newness property to the authorized users for receiving the most recent version of the stored data. We provide the security issues of the proposed scheme. Besides, we validate its performance through theoretical analysis and experimental evaluation of storage and computation overheads.

Keywords: *IT, CSPs, DSPC, Saas, trust, resource, overheads, cryptographic, etc.*

1. Introduction

IN recent years, cloud storage service has become a faster profit growth point by providing a comparably low-cost, scalable, position-independent platform for clients' data. Since cloud computing environment is constructed based on open architectures and interfaces, it has the capability to incorporate multiple internal and/or external cloud services together to provide high interoperability. We call such a distributed cloud environment as a *multi-Cloud* (or *hybrid cloud*). Often, by using virtual infrastructure management (VIM), a multi-cloud allows clients to easily access his/her resources remotely through interfaces such as Web services provided by Amazon EC2 [1].

There exist various tools and technologies for multi cloud, such as Platform VM Orchestrator, VMware vSphere, and Ovirt. These tools help cloud providers construct a distributed cloud storage platform (DCSP) for managing clients' data. However, if such an important platform is vulnerable to security attacks, it would bring irretrievable losses to the clients. For example, the confidential data in an enterprise may be illegally accessed through a remote interface provided by a multi-cloud, or relevant data and archives may be lost or tampered with when they are stored into an uncertain storage pool outside the enterprise. Therefore, it is indispensable for cloud service providers (CSPs) to provide security techniques for managing their storage services [2].

Cloud computing has received considerable attention from both academia and industry due to a number of important advantages including: cost effectiveness, low management overhead, immediate access to a wide range of applications, flexibility to scale up and down information technology (IT) capacity, and mobility where customers can access information wherever they are, rather than having to remain at their desks. Cloud computing is a distributed computational model over a large pool of shared-virtualized computing resources (e.g., storage, processing power, memory, applications, services, and network bandwidth). Cloud service providers (CSPs) offer different classes of services (Storage-as-a-Service (SaaS), Application-as-a-Service, and

Platform-as-a-Service) that allow organizations to concentrate on their core business and leave the IT operations to experts. In the current era of digital world, different organizations produce a large amount of sensitive data including personal information, electronic health records, and financial data. The amount of digital data increases at a staggering rate; doubling almost every year and a half [1, 2]. This data needs to be widely distributed and stored for a long time due to operational purposes and regulatory compliance. The local management of such huge amount of data is problematic and costly. While there is an observable drop in the cost of storage hardware, the management of storage has become more complex and represents approximately 75% of the total ownership cost [3]. SaaS offered by CSPs is an emerging solution to mitigate the burden of large local data storage and reduce the maintenance cost via the concept of outsourcing data storage. Through outsourcing data storage scenario, data owners delegate the storage and management of their data to a CSP in exchange for pre-specified fees metered in GB/month. Such outsourcing of data storage enables owners to store more data on remote servers than on private computer systems. Moreover, the CSP often provides better disaster recovery by replicating the data on multiple servers across multiple data centers achieving a higher level of availability. Thus, many authorized users are allowed to access the remotely stored data from different geographic locations making it more convenient for them [4, 5].

2. Techniques & Structures

In this section, we present our verification framework for multi-cloud storage and a formal definition of CPDP. We introduce two fundamental techniques for constructing our CPDP scheme: hash index hierarchy (HIH) on which the responses of the clients' challenges computed from multiple CSPs can be combined into a single response as the final result; and homomorphism verifiable response (HVR) which supports distributed cloud storage in a multi-cloud storage and implements an efficient construction of collision resistant hash function, which can be viewed as a random oracle model in the verification protocol.

2.1 Framework of Multi cloud Verification

Although existing PDP schemes offer a publicly accessible remote interface for checking and managing the tremendous amount of data, the majority of existing PDP schemes are incapable to satisfy the inherent requirements from multiple clouds in terms of communication and computation costs. To address this problem, we consider a multi-cloud storage service as illustrated in Figure 1. In this architecture, a data storage service involves three different entities: Clients who have a large amount of data to be stored in multiple clouds and have the permissions to access and manipulate stored data; Cloud Service Providers (CSPs) who work together to provide data storage services and have enough storages and computation resources; and Trusted Third Party (TTP) who is trusted to store verification parameters and offer public query services for these parameters [6, 7].

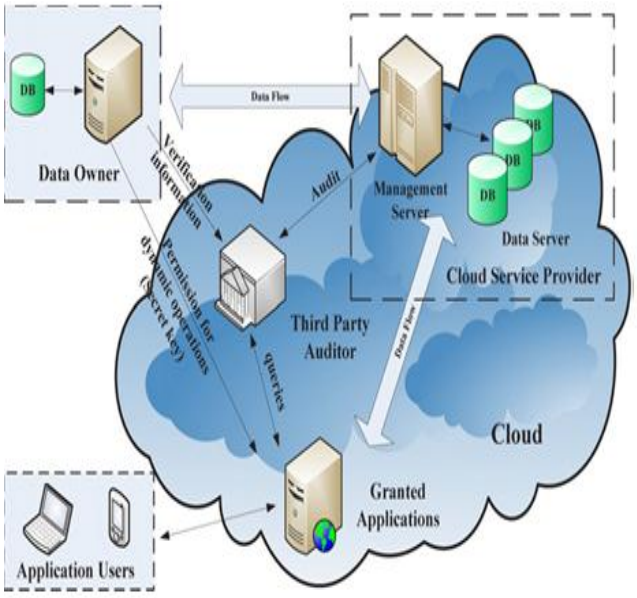

Fig: Data Integrity verification Architecture

2.2 Co operative PDP Definition

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-prove zero-knowledge proof system (MPZKPS), as follows:

Definition 1 (Cooperative-PDP): A cooperative provable data possession $\mathcal{S} = (KeyGen, TagGen, Proof)$ is a collection of two algorithms $(KeyGen, TagGen)$ and an interactive proof system $Proof$, as follows:

$KeyGen(1^{\kappa})$**:** takes a security parameter κ as input, and returns a secret key sk or a public-secret key pair (pk, sk);

$TagGen(sk, F, \mathcal{P})$**:** takes as inputs a secret key sk, a file F, and a set of cloud storage providers $\mathcal{P} = \{Pk\}$, and returns the triples (ζ, ψ, σ), where ζ is the secret in tags, $\psi = (u, \mathcal{H})$ is a set of verification parameters u and an index hierarchy $\mathcal{H}$ for F, $\sigma = \{\sigma(k)\}Pk \in \mathcal{P}$ denotes a set of all tags, $\sigma(k)$ is the tag of the fraction $F(k)$ of F in Pk;

$Proof(\mathcal{P}, V)$**:** is a protocol of proof of data possession between CSPs ($\mathcal{P} = \{Pk\}$) and a verifier (V), that is,

$\langle \Sigma\, Pk \in \mathcal{P}\ Pk(F(k), \sigma(k)) \longleftrightarrow V \rangle (pk, \psi)$

$= \{$ 1 $F = \{F(k)\}$ is intact

0 $F = \{F(k)\}$ is changed ,

Where, each Pk takes as input a file $F(k)$ and a set of tags $\sigma(k)$, and a public key pk and a set of public parameters ψ are the common input between P and V. At the end of the protocol run, V returns a bit $\{0|1\}$ denoting false and true.

Where,

$\Sigma\, Pk \in \mathcal{P}$ denotes cooperative computing in $Pk \in \mathcal{P}$.

A trivial way to realize the CPDP is to check the data stored in each cloud one by one, i.e.,$Pk \in \mathcal{P}$ $\langle Pk(F(k), \sigma(k)) \longleftrightarrow V \rangle (pk, \psi)$,

Where,

$\wedge$ denotes the logical AND operations among the Boolean outputs of all protocols $\langle Pk, V \rangle$ for all $P_k \in \mathcal{P}$.

However, it would cause significant communication and computation overheads for the verifier, as well as a loss of location-transparent. Such a primitive approach obviously diminishes the advantages of cloud storage: scaling arbitrarily up and down on demand.

2.3 System Relation & Components

The cloud computing storage model considered in this work consists of four main components as illustrated in Fig. 1: (i) a data owner that can be an organization generating sensitive data to be stored in the cloud and made available for controlled external use; (ii) a CSP who manages cloud servers and provides paid storage space on its infrastructure to store the owner's files and make them available for authorized users; (iii) authorized users – a set of owner's clients who have the right to access the remote data; and (iv) a trusted third party (TTP), an entity who is trusted by all other system components, and has expertise and capabilities to detect and specify dishonest parties. In Fig. 1, the relations between different system components are represented by double-sided arrows, where solid and dashed arrows represent trust and distrust relations, respectively.

For example, the data owner, the authorized users, and the CSP trust the TTP. On the other hand, the data owner and the authorized users have mutual distrust relations with the CSP. Thus, the TTP is used to enable indirect mutual trust between these three components. There is a direct trust relation between the data owner and the authorized users.

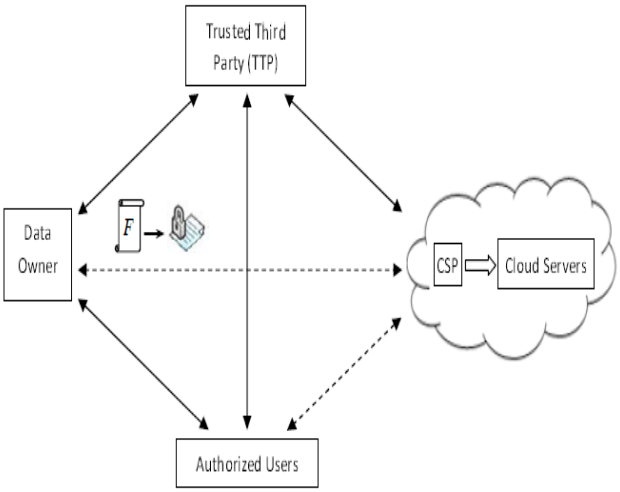

Fig: 2 Cloud Computing Data storage System Mode

3. System Preliminaries

3.1 Lazy Revocation

The proposed scheme in this work allows the data owner to revoke the right of some users for accessing the outsourced data. In lazy revocation, it is acceptable for revoked users to read (decrypt) unmodified data blocks. However, updated or new blocks must not be accessed by such revoked users. The idea is that allowing revoked users to read unchanged data blocks is not a significant loss in security. This is equivalent to accessing the blocks from cashed copies. Updated or new blocks following a revocation are encrypted under new keys. Lazy revocation trades re-encryption and data access cost for a degree of security. However, it causes fragmentation of encryption keys, i.e., data blocks could have more than one key. Lazy revocation has been incorporated into many cryptographic systems [8]. To insert “Tables” or “Figures”, please paste the data as stated below. being described, using 8pt font and please make use of the specified style “caption” from the drop-down menu of style categories.

3.2 Key Rotatio

Key rotation is a technique in which a sequence of keys can be generated from an initial key and a master secret key. The sequence of keys has two main properties: (i) only the owner of the master secret key is able to generate the next key in the sequence from the current key, and (ii) any authorized user knowing a key in the sequence is able to generate all previous versions of that key. In other words, given the i-th key Ki in the sequence, it is computationally infeasible to compute keys fKlg for l > i without having the master secret key, but it is easy to compute keys fKjg for j < i. The first property enables the data owner to revoke access to the data by producing new keys in the sequence, which are used to encrypt updated/new blocks following a revocation (lazy revocation). It is intended to prevent a user revoked during the i-th time from getting access to data blocks encrypted during the l-th time for l > i.

3.3 Broadcast Encryption

Broadcast encryption (bENC), allows a broadcaster to encrypt a message for an arbitrary subset of a group of users. The users in the subset are only allowed to decrypt the message. However, even if all users outside the subset collude they cannot access the encrypted message. Such systems have the collusion resistance property, and are used in many practical applications including TV subscription services and DVD content protection. The proposed scheme in this work uses bENC to enforce access control in outsourced data. The bENC is composed of three algorithms: SETUP, ENCRYPT, and DECRYPT.

4. Conclusions

In this paper, we presented the construction of an efficient PDP scheme for distributed cloud storage. Based on homomorphism verifiable response and hash Index hierarchy, we have proposed a cooperative PDP 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. Furthermore, we optimized the probabilistic query and periodic verification to improve the audit performance. Our experiments clearly demonstrated that our approaches only introduce a small amount of computation and communication overheads. Therefore, our solution can be treated as a new candidate for data integrity verification in outsourcing data storage systems..

Acknowledgments

The work of Y. Zhu and M. Yu was supported by the National Natural Science Foundation of China. This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS 14 Ahn and Hongxin Hu was partially supported by the grants from US National Science Foundation.

References

[1] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. T. Foster, "Virtual infrastructure management in private and hybrid clouds," *IEEE Internet Computing*, vol. 13, no. 5, pp. 14–22, 2009.

[2] A. Singh and L. Liu, "Sharoes: A data sharing platform for outsourced enterprise storage environments," in Proceedings of the 24th International Conference on Data Engineering, ICDE. IEEE, 2008, pp. 993–1002.

[3] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008.

[4] A. Juels and B. S. K. Jr., "Pors: proofs of retrievability for large files," in *ACMConference on Computer and Communications Security*, P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds. ACM, 2007, pp. 584–597.

[5] C. Erway, A. K¨upc¸ ¨ u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in CCS '09: Proceedings of the 16th ACM Conference on Computer and Communications Security, New York, NY, USA, 2009, pp. 213–222.

[6] Q. Wang, C.Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in *ESORICS*, ser. Lecture Notes in Computer Science, M. Backes and P. Ning, Eds., vol. 5789. Springer, 2009, pp. 355–370.

[7] Y. Dodis, S. P. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in *TCC*, ser. Lecture Notes in Computer Science, O. Reingold, Ed., vol. 5444. Springer, 2009, pp. 109–127.

[8] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology (CRYPTO'2001)*, vol. 2139 of LNCS, 2001, pp. 213–229.

[9] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in SecureComm '08: Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, New York, NY, USA, 2008, pp.1–10.

[10] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," Cryptology ePrint Archive, Report 2009/081, 2009, http://eprint.iacr.org/.

[11] P. S. L. M. Barreto, S. D. Galbraith, C. O'Eigeartaigh, and M. Scott, "Efficient pairing computation on supersingular abelian varieties," *Des. Codes Cryptography*, vol. 42, no. 3, pp. 239–271, 2007.