Volume 2, Issue 12, December-2015, pp. 1192-1202 ISSN (O): 2349-7084

International Journal of Computer Engineering In Research Trends

Available online at: www.ijcert.org

Cooperative Provable Data Possession for Multi-Cloud Server Storage

[1]VIJAY KUMAR, [2] CH.HARIKA

[1](M.Tech) CSE, Dept. of Computer Science and Engineering
[2]Assistant Professor, Dept. of Computer Science and Engineering
Priyadarshini Institute of Technology & Science

Abstract— Provable data possession (PDP) is a strategy for guaranteeing the trustworthiness of information away outsourcing. In this paper, we address the construction of a proficient PDP scheme for disseminated distributed storage to bolster the versatility of administration and information migration, in which we consider the presence of different cloud administration providers to agreeably store and keep up the customers' information. We display an cooperative PDP (CPDP) scheme based on homomorphic verifiable response and hash file indexing order. We demonstrate the security of our scheme based on multi-proven zero-knowledge proof system, which can fulfill culmination, information soundness, and zero-information properties. In addition, we verbalize execution optimization instruments for our scheme, and specifically exhibit a proficient technique for selecting ideal parameter qualities to minimize the computation expenses of customers and capacity administration providers. Our examinations demonstrate that our solution presents lower computation and communication overheads in comparison with non-helpful methodologies

Index Terms— Storage Security, Provable Data Possession, Interactive Protocol, Zero-knowledge, Multiple Cloud, Cooperative

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 inter- faces, it has the capability to incorporate multiple internal and/or external cloud services together to pro- vide high interoperability. We call such a distributed cloud environment as a multi-Cloud (or hybrid cloud). Often, by using virtual infrastructure management (VIM) [1], a multi-cloud allows clients to easily access his/her resources remotely through interfaces such as web services provided by Amazon EC2.

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. There- fore, it is indispensable for cloud service providers (CSPs) to provide security techniques for managing their storage services.

Provable data possession (PDP) [2] (or proofs of irretrievability (POR) [3]) is such a probabilistic proof technique for a storage provider to prove the integrity and ownership of clients' data without downloading data. The proof-checking without downloading makes it especially important for large-size files and folders (typically including many clients' files) to check whether these data have been tampered with or deleted without downloading the latest version of data. Thus, it is able to replace traditional hash and signature functions in storage outsourcing. Various PDP schemes have been recently proposed, such as Scalable PDP [4] and Dynamic PDP [5]. However, these schemes mainly focus on PDP issues at un-trusted servers in a single cloud storage provider and are not suitable for a multi-cloud environment (see the comparison of POR/PDP schemes in Table 1).

.

2. LITERATURE REVIEW

Several researchers did research on the cloud service providers such as [1] Provable data possession at untrusted stores: authors G. Ateniese, R. Burns 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.[2] Efficient remote data possession checking in critical information infrastructures: Checking data possession in networked information systems such as those related to critical infrastructures (power facilities, airports, data vaults, defense systems, and so forth) is a matter of crucial importance.[3] Scalable and efficient provable data possession :Storage outsourcing is a rising trend which prompts a number of interesting security issues, many of which have been extensively investigated in the past. [4]Dynamic provable data possession: As storage-outsourcing services and resource-sharing networks have become popular, the problem of efficiently proving the integrity of data stored at untrusted servers has received increased attention. In the provable data possession (PDP) model, the client preprocesses the data and then sends it to an untrusted server for storage, while keeping a small amount of meta-data. The client later asks the server to prove that the stored data has not been tampered with or deleted (without downloading the actual data). [5]Enabling public verifiability and data dynamics for storage security in cloud computing: Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. It moves the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. This work studies the problem of ensuring the integrity of data storage in Cloud Computing.

STRUCTURE AND TECHNIQUES

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 homomorphic 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.

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 is 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.

We neither assume that CSP is trust to guarantee the security of the stored data, nor assume that data owner has the ability to collect the evidence of the CSP's fault after errors have been found. To achieve this goal, a TTP server is constructed as a core trust base on the cloud for the sake of security. We assume the TTP is reliable and independent through the following functions [12]: to setup and maintain the CPDP cryptosystem; to generate and store data owner's public key; and to store the public parameters used to execute the verification protocol in the CPDP scheme. Note that the TTP is not directly involved in the CPDP scheme in order to reduce the complexity of cryptosystem.

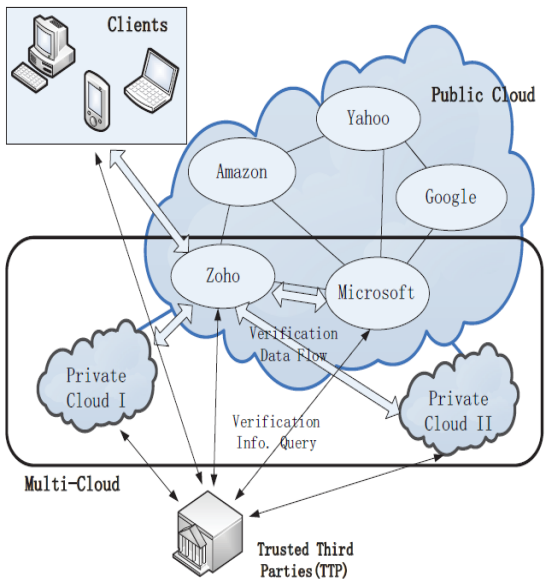

Fig1: Verification architecture for data integrity.

In this architecture, we consider the existence of multiple CSPs to cooperatively store and maintain the clients' data. Moreover, a cooperative PDP is used to verify

the integrity and availability of their stored data in all CSPs. The verification procedure is described as follows: Firstly, a client (data owner) uses the secret key to pre-process a file which consists of a collection of ◎ blocks, generates a set of public verification information that is stored in TTP, transmits the file and some verification tags to CSPs, and may delete its local copy; Then, by using a verification protocol, the clients can issue a challenge for one CSP to check the integrity and availability of outsourced data with respect to public information stored in TTP.

COOPERATIVE PDP SCHEME

In this section, we propose a CPDP scheme for multi cloud system based on the above-mentioned structure and techniques. This scheme is constructed on collision-resistant hash, bilinear map group, aggregation algorithm, and Homomorphic responses. In our scheme the manager first runs algorithm to obtain the public/private key pairs for CSPs and users. Then, the clients generate the tags of outsourced data by using. Anytime, the protocol is performed by a 5-move interactive proof protocol between a verifier and more than one CSP, in which CSPs need not to interact with each other during the verification process, but an organizer is used to organize and manage all CSPs. This protocol can be described as follows: 1) the organizer initiates the protocol and sends a commitment to the verifier; 2) the verifier returns a challenge set of random index-coefficient pairs to the organizer; 3) the organizer relays them into each in according to the exact position of each data block; 4) each returns its response of challenge to the organizer; and 5) the organizer synthesizes a final response from received responses and sends it to the verifier. The above process would guarantee that the verifier accesses files without knowing on which CSPs or in what geographical locations their files reside. In contrast to a single CSP environment, our scheme differs from the common PDP scheme in two aspects:

1)Tag aggregation algorithm: In stage of commitment, the organizer generates a random and returns its commitment ◎′ 1 to the verifier. This assures that the verifier and CSPs do not obtain the value of. Therefore, our approach guarantees only the organizer can compute the final by using and received from CSPs.

2) Homomorphic responses: Because of the homomorphic property, the responses computed from CSPs. in a multi-cloud can be combined into a single final response.

3. SYSTEM ANALYSIS

We give a brief security analysis of our CPDP construction. This construction is directly derived from multi-proven zero-knowledge proof system (MPZKPS), which satisfies following properties for a given assertion,

1) Completeness: whenever there exists a strategy for the proves that convinces the verifier that this is the case;

2) Soundness: whenever, whatever strategy the proves employ, they will not convince the verifier that;

3) Zero-knowledge: no cheating verifier can learn anything other than the veracity of the statement.

According to existing IPS research [15], these properties can protect our construction from various attacks, such as data leakage attack (privacy leakage), tag forgery attack (ownership cheating), etc. In details, the security of our scheme can be analyzed as follows:

In our CPDP scheme, the collision resistant of index hash hierarchy is the basis and prerequisite for the security of whole scheme, which is described as being secure in the random oracle model. Although the hash function is collision resistant, a successful hash collision can still be used to produce a forged tag when the same hash value is reused multiple times, e.g., a legitimate client modifies the data or repeats to insert and delete data blocks of outsourced data. To avoid the hash collision, the hash value, which is used to generate the tag in CPDP scheme, is computed from the set of values .As long as there exists one bit difference in these data, we can avoid the hash collision. As a consequence, we have the following theorem (see Appendix B): Theorem 1 (Collision Resistant): The index-hash hierarchy in CPDP scheme is collision resistant, even if the client generates files with the same file name and cloud name, √ and the client repeats times to modify, insert and delete data blocks, where the collision probability is at least

4. PERFORMANCE EVALUATION

In this section, to detect abnormality in a low overhead and timely manner, we analyze and optimize the performance of CPDP scheme based on the above scheme from two aspects: evaluation of probabilistic queries and optimization of length of blocks. To validate the effects of scheme, we introduce a prototype of CPDP-based audit system and present the experimental results.

We present the computation cost of our CPDP scheme in Table 3. We use to denote the computation cost of an exponent operation in namely where is a positive integer in and We neglect the computation cost of algebraic operations and simple modular arithmetic operations because they run fast enough [16]. The most complex operation is the computation of a bilinear map between two elliptic points .We recall the probabilistic verification of

common PDP scheme (which only involves one CSP), in which the verification process achieves the detection of CSP server misbehavior in a random sampling mode in order to reduce the workload on the server. The detection probability of disrupted blocks is an important parameter to guarantee that these blocks can be detected in time. Assume the CSP modifies blocks out of the block file, that is, the probability of disrupted blocks is. Let be the number of queried blocks for a challenge in the verification protocol. We have detection probability

Parameter Optimization: In the fragment structure, the number of sectors per block ◎ is an important parameter to affect the performance of storage services and audit services.

Hence, we propose an optimization algorithm for the value of s in this section. Our results show that the optimal value can not only minimize the computation and communication overheads, but also reduce the size of extra storage, which is required to store the verification tags in CSPs.

CPDP for Integrity Audit Services: Based on our CPDP scheme, we introduce audit system architecture for outsourced data in multiple clouds by replacing the TTP with a third party auditor (TPA) in Figure 1. In this architecture, this architecture can be constructed into a visualization infrastructure of cloud-based storage service [1]. In Figure 5, we show an example of applying our CPDP scheme in Hadoop distributed file system (HDFS) 4, which a distributed, scalable, and portable file system [19]. HDFS' architecture is composed of Name Node and Data Node, where Name Node maps a file name to a set of indexes of blocks and Data Node indeed stores data blocks. To support our CPDP scheme, the index-hash hierarchy and the metadata of Name Node should be integrated together to provide an enquiry service for the hash value ◎(3) ◎,◎ or index-hash record ◎◎.

Based on the hash value, the clients can implement the verification protocol via CPDP services. Hence, it is easy to replace the checksum methods with the CPDP scheme for anomaly detection in current HDFS. To validate the effectiveness and efficiency of our proposed approach for audit services, we have implemented a prototype of an audit system. We simulated the audit service and the storage service by using two local IBM servers with two Intel Core 2 processors at 2.16 GHz and 500M RAM running Windows Server 2003. These servers were connected via 250 MB/sec of network bandwidth. Using GMP and PBC libraries, we have implemented a cryptographic library upon which our scheme can be constructed. This C library contains approximately 5,200 lines of codes and has been tested on both Windows and Linux platforms.

The elliptic curve utilized in the experiment is a MNT curve, with base field size of 160 bits and the embedding degree 6. The security level is chosen to be 80 bits, which means

5. CONCLUSION

In this paper, 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 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.

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] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner, Z. N. J. Peterson, and D. X. Song, "Provable data possession at untrusted stores," in ACM Conference on Computer and Communications Security, P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds. ACM, 2007, pp. 598–609.

[3] 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.

[4] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proceedings of the 4th international conference on Security and privacy in communication netowrks, SecureComm, 2008, pp. 1–10.

[5] C. C. Erway, A. K¨upc¸ ¨u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in ACM Conference on Computer and Communications Security, E.

Al-Shaer, S. Jha, and A. D. Keromytis, Eds. ACM, 2009, pp. 213–222.

[6] H. Shacham and B. Waters, "Compact proofs of retrievability," in ASIACRYPT, ser. Lecture Notes in Computer Science, J. Pieprzyk, Ed., vol. 5350. Springer, 2008, pp. 90–107.

[7] 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.

[8] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Dynamic audit services for integrity verification of outsourced storages in clouds," in SAC, W. C. Chu, W. E. Wong, M. J Palakal, and C.-C. Hung, Eds. ACM, 2011, pp. 1550–1557.

[9] K. D. Bowers, A. Juels, and A. Oprea, "Hail: a high-availability and integrity layer for cloud storage," in ACM Conference on Computer and Communications Security, E. Al-Shaer, S. Jha, and A. D. Keromytis, Eds. ACM, 2009, pp. 187–198.

[10] 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.

[11] L. Fortnow, J. Rompel, and M. Sipser, "On the power of multiprover interactive protocols," in Theoretical Computer Science, 1988, pp. 156–161.

[12] Y. Zhu, H. Hu, G.-J. Ahn, Y. Han, and S. Chen, "Collaborative integrity verification in hybrid clouds," in

Applications and Worksharing, CollaborateCom, Orlando,

Florida, USA, October 15-18, 2011, pp. 197–206.

[13] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech. Rep., Feb 2009.

[14] 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.

