Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 3, March 2015, pg.208 – 211

RESEARCH ARTICLE

Verifying Integrity Using PDP Technique in Hybrid Cloud Storage

Chaitali Kharbade

Nuva College of Engineering & Technology, Nagpur

[1] chaitali2kharbade@gmail.com

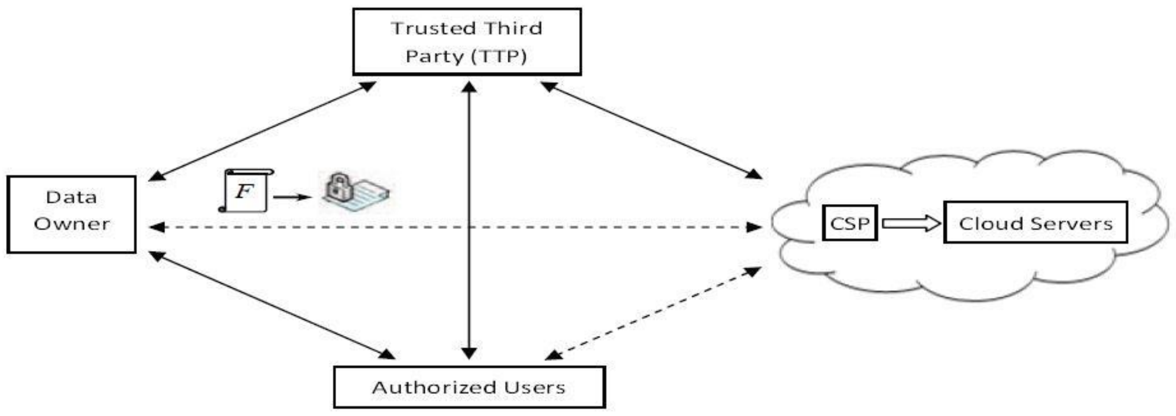
Abstract*— Cloud based data storage systems have many complexities regarding critical/confidential/sensitive data of client. The trust required on Cloud storage is so far had been limited by users. The role of the paper is to grow confidence in Users towards Cloud based data storage. We propose a cloud-based storage scheme that allows the data owner to benefit from the facilities offered by the CSP and enables indirect mutual trust between them. We prove the security of our scheme based on multi-prover zero-knowledge proof system, which can satisfy completeness, knowledge soundness, and zero knowledge properties. In addition, we articulate performance optimization mechanisms for our scheme, and in particular present an efficient method for selecting optimal parameter values to minimize the computation costs of clients and storage service providers. Our experiments show that our solution introduces lower computation and communication overheads in comparison with non-cooperative approaches

***Keywords*— Outsourcing data storage, dynamic environment, mutual trust, access control, CSP, TPA**

I. Introduction

In the current era of digital world, various organizations produce a large amount of sensitive data including personal information, electronic health records, and financial data. The local management of such huge amount of data is problematic and costly due to the requirements of high storage capacity and qualified personnel. Therefore, Storage-as-a-Service offered by cloud service providers (CSPs) emerged as a solution to mitigate the burden of large local data storage and reduce the maintenance cost by means of outsourcing data storage. to a remote CSP, there are some concerns regarding *confidentiality*, *integrity*, and *access control* of the data. The confidentiality feature can be guaranteed by the owner via encrypting the data before outsourcing to remote servers. For verifying data integrity over cloud servers, researchers have proposed provable data possession technique to validate the intactness of data stored on remote sites. A number of PDP protocols have been presented to efficiently validate the integrity of data. 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) [1], a multi-cloud allows clients to easily access his/her resources remotely through interfaces such as Web services provided by Amazon EC2. Even though

existing PDP schemes have addressed various security properties, such as public verifiability [2], dynamics [5], scalability [4], and privacy preservation [7].



OUR SYSTEM AND ASSUMPTION

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. As the cloud technology growing on it is difficult for user to map the pros and cons of this technology, basically there are two main areas of cloud computing where lots of challenges are arriving with benefits: cloud storage and cloud backup/recovery. While both these services in cloud are used for to improve manageability, and can be integrated easily to backup most all aspects of a businesses‟ data requirements, from server to laptop. As the IT people are having higher-performance, more scalable and cheaper storage business benefits from cloud. But following are the challenges meet by the end user if the proper cloud services are not used by the client/CSPS (cloud service provider).[10]

II. STRUCTURE AND TECHNIQUE

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.

2.1 Verification Framework for Multi-Cloud

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.

2.3 Hash Index Hierarchy for CPDP

To support distributed cloud storage, we illustrate a representative architecture used in our cooperative PDP scheme as shown in Figure 2. Our architecture has a hierarchy structure which resembles a natural representation of file storage. This hierarchical structure $\mathcal{H}$ consists of three layers to represent relationships among all blocks for stored resources. They are described as follows:

1) **Express Layer**: offers an abstract representation of the stored resources;
2) **Service Layer**: offers and manages cloud storage services;

3) **Storage Layer**: realizes data storage on many physical devices.

Block Status Table

The block status table (BST) is a small *dynamic* data structure used to reconstruct and access file blocks outsourced to the CSP. The BST consists of three columns: serial number (*SN*), block number (*BN*), and key version (*KV*). *SN* is an indexing to the file blocks. It indicates the physical position of each block in the data file. *BN* is a counter used to make a logical numbering/indexing to the file blocks. Thus, the relation between *BN* and *SN* can be viewed as a mapping between the logical number *BN* and the physical position *SN*. The column *KV* indicates the version of the key that is used to encrypt each block in the data file. The BST is implemented as a linked list to simplify the insertion and deletion of table entries. During implementation, *SN* is not needed to be stored in the table; *SN* is considered to be the entry/table index. Thus, each table entry contains just two integers *BN* and *KV* (8 bytes), i.e., the total table size is 8*m* bytes, where *m* is the number of file blocks. When a data file is initially created, the owner initializes both *ctr* and *KV* of each block to 1. If block modification or insertion operations are to be performed following a revocation, *ctr* is incremented by 1 and *KV* of that modified/new block is set to be equal to *ctr*. shows some examples demonstrating the changes in the BST due to dynamic operations on a data file $F = \{bj\}1 \leq j \leq 8$. When the file blocks are initially created (Fig. 2a), *ctr* is initialized to 1, $SNj = BNj = j$, and $KVj = 1$: $1 \leq j \leq 8$. Fig. 2b shows no change for updating the block at position 5 since no revocation is performed. To insert a new block after position 3 in the file *F*, Fig. 2c shows that a new entry 4, 9, 1_ is inserted in the BST after *SN*3, where 4 is the physical position of the newly inserted block, 9 is the new logical block number computed by incrementing the maximum of all previous logical block numbers, and 1 is the version of the key used for encryption.

III. EXISTING ALGORITHM

In the existing cloud architecture listed below are the some of the methods available for the dynamic data storage on remote machine to access remote data.

Algorithms for Audit System

Firstly, we present the definition of two algorithms for the tag generation process as follows: ü Key Gen: takes a security parameter _ as input, and returns a public/secret key pair. TagGen (sk, F): takes as inputs the secret key sk and a file F, and returns the triple, where denotes the secret used to generate the verification tags, verification parameters u and index-hash and _ denotes the set of tags is a set of public.

Fragment Structure and Secure Tags

To maximize the storage efficiency and audit performance, general fragment structure is introduced into our audit system for outsourced storage. An instance for this framework which is used in this scheme is showed in Figure 3: an outsourced file F is split into n blocks, and each block mi is split into s sectors. The fragment framework consists of n block-tag pair, where _i is a signature tag of block mi generated by some secrets. Finally, these block-tag pairs are stored in CSP and the encryption of the secrets _ (called as PVP) is in TTP.[7][10]

IV. IMPLEMENTATION AND PROPOSED ALGORITHM

We have implemented the proposed scheme on top of Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3) [26] cloud platforms. Our implementation of the proposed scheme consists of four modules: OModule (owner module), CModule (CSP module), UModule (user module), and TModule (TTP module). OModule, which runs on the owner side, is a library to be used by the owner to perform the owner role in the setup and file preparation phase. Moreover, this library is used by the owner during the dynamic operations on the outsourced data. CModule is a library that runs on Amazon EC2 and is used by the CSP to store, update, and retrieve data from Amazon S3. UModule is a library to be run at the authorized users' side, and include functionalities that allow users to interact with the TTP and the CSP to retrieve and access the outsourced data. TModule is a library used by the TTP to perform the TTP role in the setup and file preparation phase. Moreover, the TTP uses this library during the dynamic operations and to determine the cheating party in the system. **Implementation settings**. In our implementation we use a ”large” Amazon EC2 instance to run CModule. This instance type provides total memory of size 7.5GB and 4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each). One EC2 Compute Unit provides the equivalent CPU capacity of a 1.0 - 1.2GHz 2007 Opteron or 2007 Xeon processor. A separate server in the lab is used to run TModule. This server has Intel(R) Xeon(TM) 3.6GHz processor, 2.75GB RAM, and Windows XP operating system. The OModule is executed on a desktop computer with Intel(R) Xeon(R) 2GHz processor and 3GB RAM running Windows XP. A laptop with Intel(R) Core(TM) 2.2GHz processor and 4GB RAM running Windows 7 is used to execute the UModule. We outsource a data file of size 1GB to Amazon S3. Algorithms (hashing, broadcast encryption, digital signatures, etc.) are implemented using MIRACL library version 5.5.4. For a 128-bit security level, bENC uses an elliptic curve with a 256-bit group order. In the experiments, we utilize SHA-256, 256-bit BLS signature, and Barreto- Naehrig (BN) [27] curve defined over prime field $GF(p)$ with $|p| = 256$ bits and embedding degree = 12.

V. CONCLUSIONS

We have proposed a cloud-based storage scheme which supports outsourcing of dynamic data, where the owner is capable of not only archiving and accessing the data stored by the CSP, but also updating and scaling this data on the remote servers. The proposed scheme enables the authorized users to ensure that they are receiving the most recent version of the outsourced data. Moreover, in case of dispute regarding data integrity/ newness, a TTP is able to determine the dishonest party. The data owner enforces access control for the outsourced data by combining three cryptographic techniques: broadcast encryption, lazy revocation, and key rotation. We have studied the security features of the proposed scheme.

REFERENCES

[1] Ayad Barsoum and Anwar Hasan, "Enabling Dynamic Data and Indirect Mutual Trust for Cloud Computing Storage Systems", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 2013.

[2] F. Seb´e, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," *IEEE Trans. on Knowl. And Data Eng.*, vol. 20, no. 8, 2008.

[3] 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*, 2008, pp. 1–10.

[4] C. Erway, A. K¨upc¸ ¨ u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, 2009, pp. 213–222.

[5] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in *Proceedings of the 14th European Conference on Research in Computer Security*, 2009, pp. 355–370.

[6] A. F. Barsoum and M. A. Hasan, "Provable possession and replication of data over cloud servers," Centre For Applied Cryptographic Research, Report 2010-32.pdf.

[7] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: multiple-replica provable data possession," in *28th IEEE ICDCS*, 2008, pp. 411–420.

[8] A. F. Barsoum and M. A. Hasan, "On verifying dynamic multiple data copies over cloud servers," Cryptology ePrint Archive, Report 2011/447, 2011, 2011, http://eprint.iacr.org/.

[9] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: a high-availability and integrity layer for cloud storage," in *CCS '09: Proceedings of the 16th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2009, pp. 187–198.

[10] Y. Dodis, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in *Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography*, 2009.