ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 12, March 2016

VERIFIABLE REPLICA BASED DYNAMIC DATA POSSESSION IN CLOUD COMPUTING SYSTEMS

Sangeetha.P [1], Mohanapriya.M [2]

1. P.G. Student, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India
2. HoD, Dept. of MCA, VSB Engineering College, Karur, Tamilnadu, India

***Abstract*—Increasingly more and more organizations are opting for outsourcing data to remote cloud service providers (CSPs). Customers can rent the CSPs storage infrastructure to store and retrieve almost unlimited amount of data by paying fees metered in gigabyte/month. For an increased level of scalability, availability, and durability, some customers may want their data to be replicated on multiple servers across multiple data centers. The more copies the CSP is asked to store, the more fees the customers are charged. Therefore, customers need to have a strong guarantee that the CSP is storing all data copies that are agreed upon in the service contract, and all these copies are consistent with the most recent modifications issued by the customers. In this paper, we propose a map-based provable multicopy dynamic data possession (MB-PMDDP) scheme that has the following features: 1) it provides an evidence to the customers that the CSP is not cheating by storing fewer copies; 2) it supports outsourcing of dynamic data, i.e., it supports block-level operations, such as block modification, insertion, deletion, and append; and 3) it allows authorized users to seamlessly access the file copies stored by the CSP. We give a comparative analysis of the proposed MB-PMDDP scheme with a reference model obtained by extending existing provable possession of dynamic single-copy schemes. The theoretical analysis is validated through experimental results on a commercial cloud latform. In addition, we show the security against colluding servers, and discuss how to identify corrupted copies by slightly modifying the proposed scheme.**

***Index Terms*—Cloud computing, data replication, outsourcing data storage, dynamic environment.**

1. INTRODUCTION

Outsourcing data to a remote cloud service provider (CSP) permits society to store additional data on the CSP than on private computer systems. Such Out sourcing of data storage allows society to focus on improvement and relieves the load of constant server updates and other computing matter. On one occasion the data has been outsourced to a remote CSP which may not be dependable, the data owners drop the direct control over their confidential data. This need of control raises new difficult and demanding tasks connected to data confidentiality and integrity protection in cloud computing. The confidentiality issue can be feeling by encrypting confidential data before outsourcing to remote servers. As such, it is a vital demand of customers to have strong proofs that the cloud servers still have their data and it is not being corrupt with or partially deleted over time. As a result, many researchers have payed attention on the problem of provable data possession (PDP) and proposed different systems to review the data stored on remote servers. PDP is a method for authenticating data integrity over remote servers. In a typical PDP model, the data owners produce some metadata for a data file to be used later for verification purposes through a challenge-response protocol with the remote/cloud server. The owner sends the file to be stored on a remote server which may be untrusted, and erases the local copy of the file. One of the core design ethics of outsourcing data is to provide dynamic behavior of data for a variety of applications. This means that the slightly stored data can be not only accessed by the authorized users, but also efficient and scaled Examples of PDP constructions that deal with dynamic data [10]-[14]. The final are how-ever for a single copy of the data file. PDP method has been obtainable for multiple copies of static data [15]–[17]. PDP system directly deals with multiple copies of dynamic data. When proving multiple data copies, generally system integrity check fails if there is one or more corrupted copies were present. To deal with this issue and recognize which copies have been corrupted, a slight modification has been applied to the proposed scheme.

2.LITERATURE REVIEW

2.1. Title: Efficient Provable Data Possession for Hybrid Clouds (2010)

Provable data possession is a technique for ensuring the integrity of data in outsourcing storage service. In this paper, we propose a cooperative provable data possession scheme in hybrid clouds to support scalability of service and data migration, in which we consider the existence of multiple cloud service providers to cooperatively store and maintain the clients' data. In this work, we focus on the construction of PDP scheme for hybrid clouds, supporting privacy protection and dynamic scalability. We first provide an effective construction of Cooperative Provable Data Possession (CPDP) using Homomorphic Verifiable Responses (HVR) and Hash Index Hierarchy (HIH). This construction uses homomorphic property, such that the responses of the client's challenge computed from multiple CSPs can be combined into a single response as the final result of hybrid clouds. More importantly, a new hash index hierarchy is proposed for the clients to seamlessly store and manage the resources in hybrid clouds. The experimental results also validate the effectiveness of our construction. The code was written in C++ and the experiments were run on an Intel Core 2 processor with 2.16 GHz. All cryptographic operations utilize the QT and bilinear cryptographic library.

2.2. Title: On Verifying Dynamic Multiple Data Copies over Cloud Servers (2011)

Currently, many individuals and organizations outsource their data to remote cloud service providers (CSPs) seeking to reduce the maintenance cost and the burden of large local data storage. Christo Ananth et al. [2] discussed about a system,the effective incentive scheme is proposed to stimulate the forwarding cooperation of nodes in VANETs. In a coalitional game model, every relevant node cooperates in forwarding messages as required by the routing protocol. This scheme is extended with constrained storage space. A lightweight approach is also proposed to stimulate the cooperation. One fundamental advantage of using CC is pay-as-you-go pricing model, where customers pay only according to their usage of the services. This approach suffers from a severe drawback; the communication complexity is linear with the queried data size which is impractical especially when the available bandwidth is limited.

2.3. Title: PORs: Proofs of Retrievability for Large Files (2014)

In this paper, we define and explore proofs of retrievability (PORs). A POR scheme enables an archive or back-up service (prover) to produce a concise proof that a user (verifier) can retrieve a target file F, that is, that the archive retains and reliably transmits file data sufficient for the user to recover F in its entirety. The goal of a POR is to accomplish these checks without users having to download the files themselves. We explore POR protocols here in which the communication costs, number of memory accesses for the prover, and storage requirements of the user (verifier) are small parameters essentially independent of the length of F. In addition to proposing new, practical POR constructions, we explore implementation considerations and optimizations that b ear on previously explored, related schemes. In a POR, unlike a POK, neither the prover nor the verifier need actually have knowledge of F. PORs give rise to a new and unusual security definition whose formulation is another contribution of our work. We view PORs as an important tool for semi-trusted online archives This MAC-based approach is quite efficient in terms of file-expansion overhead, computational costs, and bandwidth. It has an important drawback, though: It does not permit the prover to return a digest of its responses, i.e., to hash or XOR them together.

2.4. Title: Proving Possession and Retrievability within a Cloud Environment: A Comparative Survey (2014)

In this paper we surveyed two core integrity proving schemes in detail along with different methods used for data integrity in both the schemes. Suppose that a CSP offers to store n copies of an owner's file on n different servers to prevent simultaneous failure of all copies. Thus, the data owner needs a strong evidence to ensure that the CSP is actually storing no less than n copies, all these copies are complete and correct, and the owner is not paying for a service that he does not get.The PB-PMDP scheme supports public verifiability. Generating unique differentiable copies of the data file is the core to design a multi-copy provable data possession scheme. Identical data copies enable the CSP to simply deceive the owner by storing only one copy and pretending that it stores multiple copies. Using a simple yet efficient way, the proposed scheme generates distinct copies utilizing the diffusion property of any secure encryption scheme.

2.5. Title: Provable Possession and Replication of Data over Cloud Servers (2010)

In this paper we address this challenging issue and propose Efficient Multi-Copy Provable Data Possession (EMC-PDP) protocols. We prove the security of our protocols against colluding servers. Through extensive performance analysis and experimental results, we demonstrate the efficiency of our protocols. The only advantage of the MR-PDP scheme is the reduction of the tag generation cost which is done only once during the life time of the outsourced storage system. Unfortunately, this contribution resulted in many various limitations that have been explained earlier, and one of these critical resulting limitations is the inability of the authorized users to access the file copies for the opaqueness nature of the CSP.

2.6. Title: Efficient Dynamic Provable Possession of Remote Data via Balanced Update Trees (2013)

In this work we develop a novel and efficient scheme, computation and communication over-head of which is an order of magnitude lower than those of other state-of-the-art schemes. Our solution has a number of new features such as a natural support for operations on ranges of blocks, and revision control. The advantages come at the cost of requiring the client to maintain a data structure of modest, but non-constant size. A disadvantage of Merkle hash tree based solutions is that the trees becomes unbalanced after a series of insert and delete requests.

3. PROPOSED SYSTEM

Proposed system consists of the main entities:

Data Owner - That can be an organization or an individual originally possessing sensitive data to be store in the cloud.

CSP - Who manages Cloud Servers (CSs) and provide paid storage space on its infrastructure to stores files.

Authorized Users - Users who have right to access remote data.

Verifier - It may be Data Owner or Third Party Auditor or Authorized User.

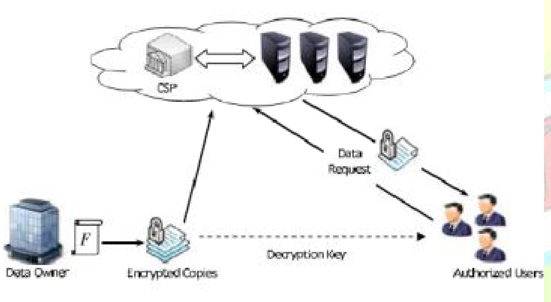

Fig 1. Cloud Computing Data Storage SystemModel

3.1 MAP-BASED PROVABLE MULTICOPY DYNAMIC DATA POSSESSION (MB-PMDDP) SCHEME

The proposed scheme consists of seven polynomial time algorithms: KeyGen, CopyGen, TagGen, Prepare- Update, ExecUpdate, Prove, and Verify. The data owner runs the algorithms KeyGen, CopyGen, TagGen, and PrepareUpdate. The CSP runs the algorithms ExecUpdate and Prove, while a verifier runs the Verify algorithm.

$(pk, sk) \leftarrow$ KeyGen(). This algorithm is run by the data owner to generate a public key *pk* and a private key *sk*.

The private key *sk* is kept secret by the owner, while *pk* is publicly known.

$\tilde{F} \leftarrow \text{CopyGen}(CNi\ ,\ F)1 \leq i \leq n$. This algorithm is run by the data owner. It takes as input a copy number *CNi*

and a file *F*, and generates *n* copies $\tilde{F} = \{\tilde{F}i\ \}1 \leq i \leq n$. The owner sends the copies $\tilde{F}$ to the CSP to be stored on

cloud servers.

$\emptyset \leftarrow \text{TagGen}(sk, \tilde{F})$. This algorithm is run by the data owner. It takes as input the private key *sk* and the file

copies $\tilde{F}$, and outputs tags/authenticators set $\emptyset$, which is an ordered collection of tags for the data blocks. The

owner sends $\emptyset$ to the CSP to be stored along with the copies $\tilde{F}$.

(D', UpdateReq) $\leftarrow$ PrepareUpdate(*D*, *UpdateInfo*). This algorithm is run by the data owner to update the outsourced file copies stored by the remote CSP. The input parameters are a previous metadata *D* stored on the owner side, and some information *UpdateInfo* about the dynamic operation to be performed on a specific block. The outputs of this algorithm are a modified metadata *D'* and an update request *UpdateReq*. This request may contain a modified version of a previously stored block, a new block to be inserted, or a delete command to delete a specific block from the file copies. *UpdateReq* also contains updated (or new) tags for modified (or inserted/appended) blocks, and it is sent from the data owner to the CSP in order to perform the requested update.

$(F, \emptyset) \leftarrow$ ExecUpdate($\tilde{F}$, $\emptyset$, *UpdateReq*). This algorithm is run by the CSP, where the input parameters are the file copies $\tilde{F}$, the tags set $\emptyset$, and the request *UpdateReq*. It outputs an updated version of the file copies $\tilde{F}$ $\emptyset$ along with an updated tags set $\emptyset'$. The latter does not require the private key to be generated; just replacement/insertion/deletion of one item of $\emptyset$ by a new item sent from the owner.

P $\leftarrow$ Prove*($\tilde{F}$, $\emptyset$, chal)*. This algorithm is run by the CSP. It takes as input the file copies$\tilde{F}$, the tags set $\emptyset$, and

a challenge *chal* (sent from a verifier). It returns a proof P which guarantees that the CSP is actually storing *n* copies and all these copies are intact, updated, and consistent.

{1, 0} $\leftarrow$ Verify*(pk,* P,*D)*. This algorithm is run by a verifier (original owner or any other trusted auditor). It takes as input the public key *pk*, the proof P returned from the CSP, and the most recent metadata *D*. The output

is 1 if the integrity of all file copies is correctly verified or 0 otherwise.

3.2 Proposed system achieves the following main objectives:

1. Implement the system which allows multi owner facility for dynamic data with notification.
2. Allow to reconstruct the corrupted copies using existing duplicate file copies.
3. Allow shared access authority by anonymous access request matching mechanism with security and privacy consideration.

System will be consisting of following module, which is formed on the basis of the functionalities that are found in the system.

Registration

The user can store the file into the cloud storage only if he/she is a registered owner of this web application. The registration can be made as either free or a paid registration depending on the organization's requirement.

Copies Generation

File copies are created by Data Owner side. Propose system allows a user to stores all copies of a file in a storage system. Each copy of file can be produced at the time and it will store into storage system with tag of each file copies.

File Division

User's file is divided into data blocks of different sizes for improving the efficiency of storage and as well as to improve security of file.

File Upload

File is uploaded on the cloud storage with the help of CSP. Files store on cloud storage infrastructure which is location independent. While we upload file ultimately CSP store all copies of file which is agreed on service.

View Files

File content can be viewed in original format by Data Owner and Authorizer User but file content is viewed in encrypted format by CSP and Verifier.

File Modify

File can be modified only by the File Owner. Modification will be done by inserting, appending, editing or changing the data.

File Deletion

The Uploaded file can be deleted by the File Owner or CSP.

File Download

Only the verified Files can be downloaded by the File Owner and Authorized User. But they don't know which copy of file is downloaded.

File Verification

Public key shared by data owner for the verification process. File cannot be downloaded by the verifier side at the time of verification process.

File Storage

File is stored in the cloud is an Encrypted format using the private key which is generated by data owner. File copies can be stored in multiple servers with unique copy.

Verifier

Verifier is one of the users in this application. Verifier is used to verify the copies of file that are stored into cloud storage. Verifier randomly checks the integrity of all file copies by sending challenge to the CSP.

Integrity Verification

Verifier randomly send a challenge to the CSP to check integrity and consistency of file copies then CSP send proof of that challenge and finally verifier check is it correct or not without downloading of files copies.

Advantages:

- Utilisatin is very effective and efficiency.
- Proof for the utilization of the spaces allocated.

4. CONCLUSION AND FUTURE SCOPE

The proposed system can be helpful to create multiple copies of sensitive data in different server. Also, it verifies integrity where CSP prove all copies are intact. In addition it identifies corrupted copies and reconstruct before dynamic operation performs. It also discussed to share access authority by providing security and privacy. Outsourcing data to remote servers has become a growing trend for many organizations to alleviate the burden of local data storage and maintenance. In this work we have studied the problem of creating multiple copies of dynamic data file and verifying those copies stored on untrusted cloud servers. We have proposed a new PDP scheme (referred to as MB-PMDDP), which supports outsourcing of multi-copy dynamic data, where the data owner is capable of not only archiving and accessing the data copies stored by the CSP, but also updating and scaling these copies on the remote servers. To the best of our knowledge, the proposed scheme is the first to address *multiple* copies of *dynamic* data. The interaction between the authorized users and the CSP is considered in our scheme, where the authorized users can seamlessly access a data copy received from the CSP using a single secret key shared with the data owner. Moreover, the proposed scheme supports public verifiability, enables arbitrary number of auditing, and allows *possession-free* verification where the verifier has the ability to verify the data integrity even though he neither possesses nor retrieves the file blocks from the server. Through performance analysis and experimental results, we have demonstrated that the proposed MB-PMDDP scheme outperforms the TB-PMDDP approach derived from a class of dynamic single-copy PDP models. The TB-PMDDP leads to high storage overhead on the remote servers and high computations on both the CSP and the verifier sides. The MB-PMDDP scheme significantly reduces the computation time during the challenge-response phase which makes it more practical for applications where a large number of verifiers are connected to the CSP causing a huge computation overhead on the servers. Besides, it has lower storage overhead on the CSP, and thus reduces the fees paid by the cloud customers. The dynamic block operations of the map-based approach are done with less communication cost than that of the tree-based approach. A slight modification can be done on the proposed scheme to support the feature of identifying the indices of corrupted copies. The corrupted data copy can be reconstructed even from a complete damage using duplicated copies on other servers. Through security analysis, we have shown that the proposed scheme is provably secure.

REFERENCES

[1]. R. Buyya, C.S. Yeo, S.Venugopal, J. Broberg , and I. Brandic, "Cloud computing and emerging IT platforms", Future generation computer system, vol. 25, no. 6, pp. 599-616, 2009.

[2]. Christo Ananth, M.Muthamil Jothi, A.Nancy, V.Manjula, R.Muthu Veni, S.Kavya, "Efficient message forwarding in MANETs", International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE), Volume 1,Issue 1, August 2015,pp:6-9

[3]. F. Sebé, J. Domingo- Ferrer, A. Martinez-Balleste, Y. Deswarte and J.-J. Quisquater,"Efficient remote data possession checking in critical information infrastructures," IEEE Trans. Knowl. Data Eng. vol. 20, no. 8, pp. 1034–1038, Aug. 2008

[4]. Giuseppe Ateniese, Roberto Di Pietro, Luigi V.Mancini and Gene Tsudik, "Scalable and Efficient Provable Data Possession", in Proceedings of the 4th international conference on Security and privacy in communication, 2008.

[5]. C. Wang, Q. Wang, K. Ren, and W.Lou.(2009)."Ensuring data security in cloud computing", http:// eprint.jacr. org/

[6]. C. Erway, C. Papamanthou, and R. Tamassia, " Dynamic provable data possession", USA, 2009, pp. 213-222.

[7]. Decio Luiz Gazzoni Filho and Paulo Sergio Licciardi Messeder Barreto, "Demonstrating data possession and uncheated data transfer", 2006

[8]. Ayad F. Barsoum and M. Anwar HAsan "Provable Multi- Copy Dynamic Data Possession in Cloud Computing Systems" IEEE trans. On information foraensics and security VOL10, NO. 3, March 2015.

[9]. Giuseppe Ateniese, Roberto Di Pietro, Luigi V.Mancini and Gene Tsudik, "Scalable and Efficient Provable Data Possession", in Proceedings of the 4th international conference on Security and privacy in communication,

2008.

[10]. RezaCurtmola, Osama Khan, Randal Burns and Giuseppe Ateniese, "MR-PDP: Multiple-Replica Provable Data Possession", in 28thInternational Conference on Distributed Computing Systems, 2008.

[11]. Ayad F. Barsoum and M. Anwar HAsan "Provable Multi- Copy Dynamic Data Possession in Cloud Computing Systems" IEEE trans. On information foraensics and security VOL10, NO. 3, March 2015.