International Journal of Engineering Research and General Science Volume 3, Issue 2, Part 2, March-April, 2015
ISSN 2091-2730

Cooperative Provable Data possession for integrity verification in multicloud

Ms.Ashwini Mandale, Prof.Shriniwas Gadage

ME CE student -G.H. Raisoni College of Engg and Management, Pune, (ashwini.mandale@gmail.com), 8407975547

Abstract— To ensure the integrity of data in storage outsourcing Provable data possession (PDP) is a technique. In this paper, there is the construction of an efficient PDP scheme for distributed cloud storage to support the scalability of service and data migration, where the existence of multiple cloud service providers to cooperatively store and maintain the client's data. This paper presents a cooperative PDP (CPDP) scheme based on homomorphic verifiable response and hash index hierarchy.

In this paper proof of the security of CPDP scheme based on multi-prover zero-knowledge proof system, which can satisfy completeness, knowledge soundness, and zero-knowledge properties, is given.

Keywords— CPDP, Interactive Protocol, Multiple Cloud, POR, Provable Data Possession, Storage Security, Zero-knowledge.

INTRODUCTION

By providing a comparably low-cost, scalable, position-independent platform for clients' data, cloud storage service has become a faster profit growth point. The cloud computing environment is constructed based on open architectures and interfaces, so has the capability to incorporate multiple internal / external cloud services together to provide high interoperability. We call such a distributed cloud environment as a *multi/hybrid Cloud*.

There are various tools and technologies for multicloud, such as Platform VM Orchestrator, VMware vSphere, and Ovirt. These technologies and tools help cloud providers construct a distributed cloud storage platform (DCSP) for managing clients data. If such an important platform is vulnerable to security attacks, it would bring irretrievable losses to the clients. So, it is indispensable for cloud service providers (CSPs) to provide security techniques for managing their storage services.

Provable data possession (PDP) [2] or known as proofs of retrievability (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 Scalable PDP [4], Dynamic PDP [5] are not suitable for a multi-cloud environment.

To provide a low-cost, scalable, location independent platform for managing clients' data, current cloud storage systems adopt several new distributed file systems, Apache HDFS, GFS, Amazon S3 File System, CloudStore etc. These file systems share some similar features: a single metadata server provides centralized management by a global namespace; files are split into blocks or chunks and stored on block servers; and the systems are comprised of interconnected clusters of block servers. Those features enable cloud service providers to store and process large amounts of data. It is crucial to offer an efficient verification on the integrity and availability of stored data for detecting faults and automatic recovery. Moreover, this verification is necessary to provide reliability by automatically maintaining multiple copies of data and automatically redeploying processing logic in the event of failures.

Although existing schemes can make a false or true decision for data possession without downloading data at untrusted stores, and are not suitable for a distributed cloud storage environment as they were not originally constructed on interactive proof system. They use an authenticated skip list to check the integrity of file blocks adjacently in space. They did not provide any algorithms for constructing distributed Merkle trees that are necessary for efficient verification in a multi-cloud environment. When a client asks for a file block, the server needs to send the file block along with a proof for the intactness of the block. This process incurs significant communication overhead in a multi-cloud environment, since the server in one cloud typically needs to generate such a proof with the help of other cloud storage services, where the adjacent blocks are stored.

The schemes PDP [2], CPOR-I [5], and CPOR-II [6] are constructed on homomorphic verification tags by which the server can generate tags for multiple file blocks in terms of a single response value. However, that doesn't mean the responses from multiple clouds can be also combined into a single value on the client side. For lack of homomorphic responses, clients must invoke the PDP protocol repeatedly to check the integrity of file blocks stored in multiple clouds servers. Also, clients need to know the exact position of each file block in a multi-cloud environment. In addition, the verification process in such a case will lead to high communication overheads and computation costs at client sides as well. Therefore, it is of utmost necessary to design a cooperative PDP model to reduce the storage and network overheads and enhance the transparency of verification activities in cluster-based cloud storage

systems. Moreover, such a cooperative PDP scheme should provide features for utimely detecting abnormality and renewing multiple copies of data.

Existing PDP schemes have various security properties such as public verifiability, dynamics, scalability and privacy preservation. There are some potential attacks:

1) Data Leakage Attack: Here an adversary can easily obtain the stored data through verification process after running or wiretapping sufficient verification communications.

2) Tag Forgery Attack: a dishonest CSP can deceive the clients. These attacks may cause potential risks for privacy leakage and ownership cheating and can more easily compromise the security of a distributed cloud system than a single cloud system.

LITERATURE SURVEY

To check the availability and integrity of outsourced data in cloud storages two basic approaches called Provable Data Possession (PDP) [2] and Proofs of Irretrievability (POR) [3].

PDP: Ateniese et al. [2] first proposed the PDP model for ensuring possession of files on untrusted storages and provided an RSA-based scheme. They also proposed a publicly verifiable version, which allows anyone to challenge the server for data possession. This property greatly extended application areas of PDP protocol due to the separation of data owners and the users. But these schemes are insecure against replay attacks in dynamic scenarios because of the dependencies on the index of blocks. And they do not fit for multi-cloud storage due to the loss of homomorphism property in the verification process.

Scalable PDP/dynamic PDP: In order to support dynamic data operations a lightweight PDP scheme based on cryptographic hash function and Symmetric key encryption is proposed. But the servers can deceive the owners by using previous metadata or responses Due to the lack of randomness in the challenges. The numbers of updates and challenges are limited and fixed in advance and users cannot perform block insertions anywhere.

DPDP-I and DPDP-II: There are two Dynamic PDP schemes with a hash function tree to realize $O(\log n)$ communication and computational costs for a n-block file. The basic scheme, called DPDP-I, retains the drawback of Scalable PDP, and in the 'blockless' scheme, called DPDPII,the data blocks $\{mij\}j\in[1,t]$ can be leaked by the response of a challenge, $M = \Sigma tj=1\ ajmij$, where aj is a random challenge value.

All above schemes are not effective for a multi-cloud environment because the verification path of the challenge block cannot be stored completely in a cloud.

POR scheme: It relies largely on preprocessing steps that the client conducts before sending a file to a CSP. Unfortunately, these operations prevent any efficient extension for updating data.

Compact POR: It is an improved version of POR protocol. It uses homomorphic property to aggregate a proof into $O(1)$ authenticator value and $O(t)$ computation cost for t challenge blocks, but their solution is also static and could not prevent the leakage of data blocks in the verification process.

A dynamic scheme with $O(\log n)$ cost: By integrating the Compact POR scheme and Merkle Hash Tree (MHT) into the DPDP.

A distributed cryptographic system: It allows a set of servers to solve the PDP problem. This system is based on an integrity-protected error correcting code (IP-ECC), which improves the security and efficiency of existing tools. However, a file must be transformed into l distinct segments with the same length, which are distributed across l servers. Hence, this system is more suitable for RAID rather than cloud storage.

PROPOSED WORK

For addressing the problem of provable data possession in distributed cloud environments from the following aspects: high security, transparent verification, and high performance.

To achieve this verification framework for multi-cloud storage along with two fundamental techniques: hash index hierarchy (HIH) and homomorphic verifiable response (HVR) is proposed.

Then possibility of constructing a cooperative PDP (CPDP) scheme without compromising data privacy based on modern cryptographic techniques, such as interactive proof system (IPS) is proposed. Then an effective construction of CPDP scheme using above-mentioned structure is introduced. Then security analysis of our CPDP scheme from the IPS model is given. As this construction is a multi-prover zero-knowledge proof system (MP-ZKPS) [11], which has completeness, knowledge soundness, and zero-knowledge properties which ensure that CPDP scheme can implement the security against data leakage attack and tag forgery attack.

A. Verification framework for multi-cloud storage:

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 consider a multi-cloud storage service as shown in Fig 1.

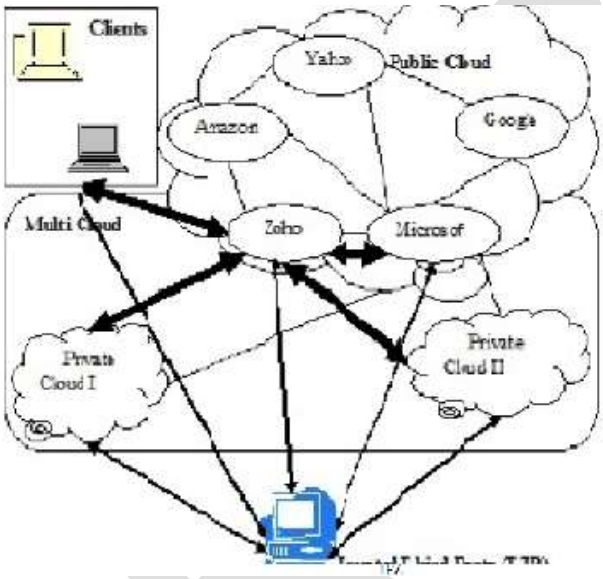

Fig 1: Architecture for data integrity in multicloud environment

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. Trusted Third Party (TTP) who is trusted to store verification parameters and offer public query services for these parameters.

This architecture consist the existence of multiple CSPs to cooperatively store and maintain the clients' data alongwith CPDP to verify the integrity and availability of their stored data in all CSPs.

The verification procedure:

I.A client (data owner) uses the secret key to pre-process a file which consists of a collection of n 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.

II.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.

B.Defination of Co-operative PDP:

A cooperative provable data possession scheme S′ is a collection of two algorithms and an interactive proof system, S′ = (K, T, P).

KeyGen (1k): It takes a security parameter k as input, and returns a secret key sk or a public-secret key pair (pk, sk).

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

Proof(P, V): It is a protocol of proof of data possession between the CSPs (P = {Pk}) and a verifier (V), that is,(ΣPk∈ PPk(F(k), σ(k)), V) (pk, ψ), where each Pk takes as input a file F(k) and a set of tags σ(k), and a public key pk and a set of public parameters ψ is 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, ΣPk∈ P denotes the collaborative computing in Pk ∈ P.

C. Hash Index Hierarchy for CPDP:

This work addresses the construction of an efficient PDP scheme for distributed cloud storage to support data migration and scalability of service, where the existence of multiple cloud service providers to cooperatively store and maintain the clients' data is considered. It presents a *cooperative*-PDP (CPDP) scheme based on homomorphic verifiable response and hash index hierarchy. Multi-prover zeroknowledge proof system is used to prove the security of this scheme, which can satisfy knowledge soundness, completeness and zero-knowledge properties.

a. Hash index hierarchy: Architecture used in cooperative PDP scheme to support distributed cloud storage as shown in Fig 2.

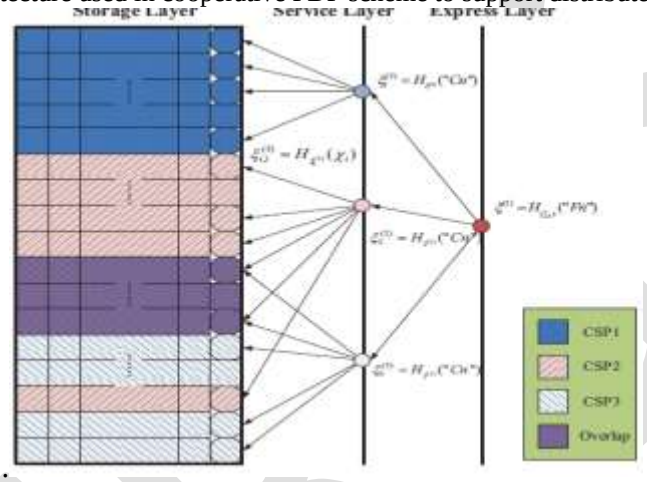

Fig 2: Index-hash hierarchy of CPDP model.

This hierarchical structure $\mathcal{H}$ consists of three layers to represent relationships among all blocks for stored resources.

a) **Express Layer**: offers an abstract representation of the stored resources.

b) **Service Layer**: offers and manages cloud storage services.

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

This hierarchy used to organize data blocks from multiple CSP services into a large size file by shading their differences among these cloud storage systems. The resource in Express Layer are split and stored into three CSPs in Service Layer. After that each CSP fragments and stores the assigned data into the storage servers in Storage Layer. It follows the logical order of the data blocks to organize the Storage Layer.

b. Homomorphic Verifiable Response:

A response is called homomorphic verifiable response in a PDP protocol, if given two responses θi and θj for two challenges Qi and Qj from two CSPs, there exists an efficient algorithm to combine them into a response θ corresponding to the sum of the challenges $Qi \cup Qj$. It is the key technique of CPDP as it reduces the communication bandwidth as well as conceals the location of outsourced data in the distributed cloud storage environment.

***c. Security Analysis*:**

Multi-prover zero-knowledge proof system is directly used for security, which satisfies following properties:

1) Collision resistant for index-hash hierarchy: The index hash hierarchy in CPDP scheme is collision resistant, even if the client generates files with the same file name and cloud name collision doesn't occur there.

2) Completeness property of verification: In this scheme, the Completeness property implies public verifiability property. Due to this property allows client as well as anyone other than client (data owner) can challenge the cloud server for data integrity and data ownership without the need for any secret information.

3) Zero-knowledge property of verification: This paper makes use of the zero-knowledge property to preserve the privacy of data blocks and signature tags. Initially, randomness is adopted into the CSPs' responses in order to resist the data leakage attacks.

4) Knowledge soundness of verification: The soundness means that it is infeasible to fool the verifier to accept false statements. Often, the soundness can also be considered as a stricter notion of unforge ability for file tags to avoid cheating the ownership. This denotes that the CSPs, even if collusion is tried, cannot be tampered with the data or forge the data tags if the soundness property holds. Thus CPDP scheme can resist the tag forgery attacks to avoid cheating the CSPs' ownership.

ACKNOWLEDGMENT

I would like to thanks to Prof.Shriniwas Gadage, Adj.faculty Computer Engg-G.H.Raisoni College of Engg and Management, Pune for his help, Encouragement and intellectual influence, which made this paper possible. His invaluable guidance in the successful completion of this paper work.

CONCLUSION

In this paper the construction of an efficient PDP scheme for distributed cloud storage is described. Based on homomorphic verifiable response and hash index hierarchy, this paper showed that CPDP scheme provided all security properties required by zeroknowledge interactive proof system, so that it can resist various attacks even if it is deployed as a public audit service in clouds.

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. 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 IEEE Conference on the 7th International Conference on Collaborative Computing: Networking 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.

[15] O. Goldreich, Foundations of Cryptography: Basic Tools. Cambridge University Press, 2001.

[16] 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.

[17] J.-L. Beuchat, N. Brisebarre, J. Detrey, and E. Okamoto, "Arithmetic operators for pairing-based cryptography," in CHES, ser. Lecture Notes in Computer Science, P. Paillier and I. Verbauwhede, Eds., vol. 4727. Springer, 2007, pp. 239–255.

[18] H. Hu, L. Hu, and D. Feng, "On a class of pseudorandom sequences from elliptic curves over finite fields," IEEE Transactions on Information Theory, vol. 53, no. 7, pp. 2598–2605, 2007.

[19] A. Bialecki, M. Cafarella, D. Cutting, and O. O'Malley, "Hadoop: A framework for running applications on large clusters built of commodity hardware," Tech. Rep., 2005. [Online]. Available: http://lucene.apache.org/hadoop/

[20] E. Al-Shaer, S. Jha, and A. D. Keromytis, Eds., Proceedings of the 2009 ACM Conference on Computer and Communications Security,CCS 2009, Chicago, Illinois, USA, November 9-13, 2009. ACM, 2009