Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-7, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

Achieving Availability and Efficient Audit Services in Cloud Computing

Pallavi R
Asst. Prof, Department of Computer Science and Engineering
Sri Venkateshwara College of Engineering,
Bangalore, Karnataka, India.

Abstract— *Cloud computing is the long dreamed version of computing as a utility in IT industry. By data outsourcing, users can be relieved from the burden of local data storage and maintenance. As cloud provides many advantages, it also brings certain challenges. Here, as the clients no longer have the physical possession of the data, they can face formidable risk for missing data without clients knowledge, cloud providers can modify or delete data which are either not used by client from long-time which occupies large space. Hence, audit services are important for ensuring the integrity and availability of outsourced data in order to reduce the security risks and to achieve credibility on cloud computing. Many schemes have been proposed to achieve audit services, such as PDP protocol, which is used to prevent dishonest of prover and the leakage of verified data. In this audit service, the third party auditor (TPA), known as the agent of the data owners, can issues a periodic verification to monitor the change of outsourced data by providing an optimized scheduled. Even though they have addressed various aspects such as Public verifiability, Dynamics, Scalability and Privacy preservation, they still lack in providing assurance of data when they experience data leakage attack and tag forgery attacks and also existing schemes are designed for public cloud. The existing schemes are not applicable for hybrid clouds. Hybrid cloud is a cloud computing environment in which organization provides and manages some internal resources and external. This new enviormnet cloud brings irretrievable losses to the clients due to a lack of data integrity and verification mechanism for distributed data outsourcing. Hence, we propose a collaborative provable data possession (CPDP) method for providing data integrity in hybrid cloud and provides dynamic scalability and data migration in hybrid cloud.*

Keyword— *Data integrity, Collaborative, Hybrid clouds and Data security.*

I. INTRODUCTION

CLOUD computing is most demanding and emerging technology throughout the world. Cloud computing had been envisioned as next generation information technology (IT) architecture for enterprises, due to its long list of advantages in the IT history: on- demand self-services ubiquitous network access, location independent resources pooling, rapid resources elasticity, usage-based price and transferences of risk. Although commercial cloud services have revolved around public clouds, the growth of private cloud on open- source cloud computing tools allows local users to have flexible and agile private infrastructure to run service workloads within their administrative domains. Private clouds are not exclusive for being public clouds and they can also support a hybrid cloud mode by supplementing a local infrastructure with computing capacity from an external public cloud. By using virtual infrastructures management (VIM), a hybrid cloud can allow remote access to its resources over the internet via remote interfaces, such as the web services interface that Amazon EC2 uses. Therefore a hybrid cloud puts more emphasis on cloud aggregation platform including private clouds and public clouds, combining the features of availability, scalability and low cost from public clouds and security from private clouds.

Although cloud computing envisioned as a promising service platform for the internet, the new data storage paradigm in "cloud" brings about many challenging design issues which have influence on the security and performance of the overall system. One of the biggest concerns with cloud data storage is that of data integrity and data verification at untrusted servers. For example, the storage services provides(or service provider, which experiences Byzantine failures occasionally, may decide to hide the data errors from the clients for the benefit of their own. For the purpose of saving money and storage space, the service provides might neglect to keep or delete rarely accessed data file which belongs to a client that may arise in disputes between clients and cloud service provides (CSPs). Therefore, it is indispensable for CSPs to provide secure management techniques to ensure their storage services. Traditional cryptographic techniques of data integrity and availability, based on hash functions and signature schemes (Hsiao et al, 2003; Yumerefendi and chase, 2007), cannot work on the outsourced data

without a local copy of data. It is also not a practical solution for data validation by downloading them due to expensive transaction, especially for large-size files. Moreover, the solution to audit the corrections of the data in cloud environment can be formidable and expensive for the cloud users (Armbrut et al, 2010). Therefore, it is crucial to realize public auditability for Cloud Storage Service (CSS), so that data owners may resort to a third party auditor (TPA), who has expertise and capabilities that a common user does not have periodically auditing the outsourced data. This audit service is significantly important for digital forensics and data assurance in clouds.

To implement public auditability, the notions of proof of irretrievability (POR) (Juels, 2007) and provable data possession (PDP)(Ateniese et al,2007) have been proposed by some researchers. Their approach was based on probabilistic proof techniques for a storage provider to prove the client's data remain intact without downloading the stored data, which is called "verification" without downloading. Various PDP schemes have been recently proposed, such as Scalable PDP and Dynamic PDP, to work in publically verifiable way so that users can employ their verification protocol to prove the availability of stored data. Even though existing PDP schemes have addressed various aspects such as public verifiability, dynamic, scalability and privacy presentation, still they suffer from data leakage attacks and tag forgery attack.

However, these schemes focus on PDP issues at untrusted severs (public clouds), and are not applicable for a hybrid cloud environment due to the lack of support for heterogeneous multi- cloud storage and privacy protection mechanism. They also ignore the leakage problem of verified data via the interactive process of the verification protocol in a PDP scheme. Thus, when a public verification service does not have a strong security mechanism to data protection, a malicious attacks cloud easily exploit such as service to obtain private data. These drawbacks greatly affect the impact of cloud audit services. to overcome these issues, we provide an effective construction of collaborative provable data possession (CPDP) using homomorphic verifiable response and hash index hierarchy. This construction realizes security against data leakage attacks and tag forgery attacks considering transparent property for clients to store and manages resources in hybrid clouds. It uses homomorphic property, on which the responses of client's challenges computed from multiple CSPs can be combined into a single response as the final result of hybrid clouds. By using such a mechanism, clients can be conceived of data possession without knowing geographic locations where their files reside. In addition, a new hash index hierarchy is proposed to realize the client oriented transparency measures to store and manage client resources in hybrid clouds.

II. RELATED WORK

The basic idea of data integrity and data verification untrusted outsourced storage is as follows. The most direct way to enforce the integrity control is to employ cryptographic hash function (Yumerefendi and Chase, 2007; Hsiao et al., 2009) and signature schemes (Lietal, 2006; Ma et al., 2005; Xie et al., 2009), Yavuz and Ning, 2009), but they cannot work on the outsourced data without a local copy of data. Moreover, these traditional methods are not the practical solutions for data validation by downloading them due to the expensive communications especially for large files, to check the availability and integrity for the stored data without downloading it from storage space. Researchers have proposed two basic approaches called provable data possession (PDP) (Ateniese at al., 2009) and Proof of Retrievability (POR) (Juels, 2007). Ateniese et al., first proposed the PDP model for ensuring possession of files on untrusted storages and provided a RSA- based scheme for the static case that achieves the $O(1)$ communication cost.

They also proposed publically verifiable version, which allows anyone, not only just the data owners, to challenge the server for data possession. However, these schemes are insecure in dynamic scenarios because of the dependence on index of blocks. Moreover, they do not fit for the hybrid clouds due to loss of homomorphism in the verification process. These schemes also lack in providing dynamic data operations such as query, insertion, modification and deletion. In order to support dynamic operations, Ateniese et al., have proposed a dynamic PDP solution called Scalable PDP. They proposed a lightweight PDP scheme based on cryptographic hash function and symmetric key encryption, but the server can deceive the owner by using previous metadata or responses due to the lack of randomness in the challenge. Erway et al., introduced two dynamic PDP schemes with a hash function tree to realize the $O(\log n)$ communication and computational costs for a file consisting of n blocks. But, these schemes prevent any efficient extension to update data. Shancham and Waters proposed an improved version of this protocol called Compact POR, which uses homomorphic property to aggregate a proof into $O(1)$ authenticator value and O(t) computation cost for 't' blocks, but their solution is also static and exists the leakage of data blocks in the verification process. Wang et al., presented a dynamic scheme with O(log n) cost by integrating the above CPOR scheme and Merkle Hash Tree(MHT) in DPDP. Furthermore, several POR schemes have been proposed recently including (Bowers et al., 2009;

Dodis et al., 2009). Since the response of challenges has homomorphic property, the above schemes can leverage collaborative PDP construction in hybrid clouds, in which, multiple cloud service providers collaboratively store and maintain the client's data and also provides dynamic scalability and data migration.

III. PROPOSED SYSTEM

In this section, we present a verification framework for hybrid clouds and a formal definition of collaborative PDP.

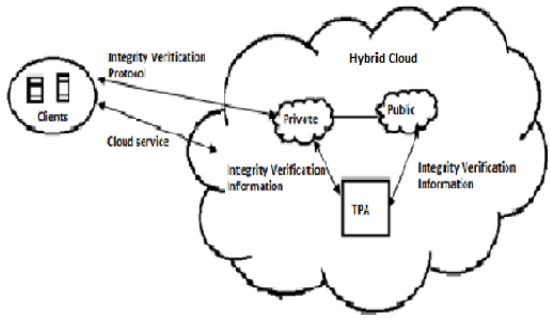

FIG 1.VERIFICATION ARCHITECTURE FOR HYBRID CLOUDS

As depicted in Fig. 1, verification architecture for data integrity in hybrid cloud.

A. *Veification frame work for hybrid clouds:*

Although PDP schemes evolved around public clouds offer a publicly accessible remote interface to check and manage the tremendous amount of data, the majority of today's PDP schemes is incapable of satisfying such an inherent requirement of hybrid clouds in the aspects of security, bandwidth and usability. To solve this problem, we consider a hybrid cloud storage service as illustrated in Fig. 1.

In this architecture, we consider a data storage service in a hybrid cloud involving three different entities:

i) Granted clients: who have large amount of data stored in a hybrid cloud and have the right to access and manipulate these stored data;

ii) Cloud Service Providers (CSPs): who work together to provide data storage service and have enough storage spaces and computational resources.

iii) Third Party Auditor (TPA): who is trusted to store verification parameters, including data and offers the query services for these parameters.

In Fig.1, a hash table is associated with TPA which contains data items such as data block position, access domain and hash value should be added to this table. In our scheme, one of the most important items is a cryptographic hash value, which is used to compress the record itself and supports data integrity verification in collaborative PDP services. More importantly, this
hash table is used to solve the heterogeneous storage problem.

In this architecture, we consider the existence of multiple CSPs to collaboratively store and maintain client's data. Moreover, a collaborative PDP is used to verify the data integrity and availability of their stored data in CSPs.

The verification process is described as follows: First, the client 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 TPA, transmits the file and some verification tags to CSPs and may delete its local copy; then, by using verification protocol for collaborative PDP the clients can issue a challenge for one CSP to check the integrity and availability of outsourced data in terms of public verification information stored in TPA.

B. *Definition of collaborative PDP:*

In order to prove the integrity of data stored in hybrid cloud in hybrid clouds, we define a framework for collaborative PDP based on multi-prover interactive proof system (MP-IPS).

Collaborative PDP: A collaborative PDP scheme 'S' is a collection of two algoriths and a MP-IPS, S=(K,T,P):

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

(b) TagGen(sk,F,P): takes as inputs a secret key sk, a file F and a set of cloud storage providers $P=\{P_k\}$, and a set of cloud storage providers $P=\{P_k\}$, and returns the triples(ζ,Ψ,σ), where ζ is the secret of tags, $\Psi=(u,H)$ is a set of verification parameters u and an index hierarchy H for F, $\sigma = \{\sigma^{(k)}\}_{P_k \in P}$ denotes a set of all tags, $\sigma^{(k)}$ is the tags of the fraction $F^{(k)}$ of F in P_k.

(c) Proof(P,V): is a protocol of proof of data possession between the CSPs ($P=\{P_k\}$) and a verifier (V), that is, $\sum_{Pk \in P} Pk(F^{(k)},\sigma^{(k)}),V(pk,\Psi)$, where each P_k 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 is the common input between P and V. At the end of protocol running, V returns a bit {0|1} denoting false and true

Where $\sum_{Pk \in P}$ denotes the collaborative computing in $P_k \in P$.

C. Security model for collaborative PDP:

In cryptography, the collaborative PDP scheme is a multi-prover interactive proof system(MP-IPS) in nature. CPDP scheme should satisfy the following security requirements:

A pair of interactive machines ($\sum_{Pk \in P} P_k$, V) is called an available provable data possession for a file F if P={P_k} is a collection of probablistic algorithms, V is a deterministic polynomial-time algorithms and the following conditions hold for some polynomial $p_1(.),p_2(.)$ and all s $\in$ N:

i) Completeness: For every $\sigma \in$ TagGen(sk,F),

$P_r[\sum Pk \in P\ Pk(F^{(k)},\sigma^{(k)}), V(pk,\Psi)=1] \geq 1-1/p_1(k)$

ii) Soundness: For every $\sigma^* \in$ TagGen(sk,F), every interactive machine $P_k^* \in P$,

$Pr[\sum Pk \in P\ Pk^*(F^{(k)}, \sigma^{(k)}), V(pk,\Psi)=1] \leq 1/p_2(k)$

Here, the knowledge soundness could be regarded as the stricter definition of security of tag information. This means that the prover can forge file tags by means of a knowledge extractor M if soundness property does not hold.

For a private cloud, we concerned more about the disclosure of private information in the verification process. It is easy to find that data blocks and their tags could be obtained by the verifier in some existing schemes.In order to solve the problem, we introduce zero-knowledge notion into the CPDP scheme, as follows:

a) Zero-knowledge:

An interactive proof system for provable data ppossession problem is computational zero knowledge if there exists a probablistic polynomial-time algorithm S*(call a simulator) such that for every probablistic polynomial-time algorithm D, for every polynomial p(.), it hods that

$Pr[D(pk,\Psi,S^*(pk,\Psi))=1] -$

$Pr[D(pk,\Psi,\sum_{Pk \in P} Pk(F^{(k)},\sigma^{(k)},V^*(pk,\Psi))] = 1 \leq 1/p(s)$

Where S*(pk,Ψ) denotes the output of simulator.

Actually,zero-knowledge is a property that captures P's robustness against attempts to gain knowledge by interacting with it. For the PDP scheme, we use the zero-knowledge property to the security of data blocks and signature tags.

D. Verification process:

The verification process is performed by a five-more interactive proof protocol shown in Fig.2:

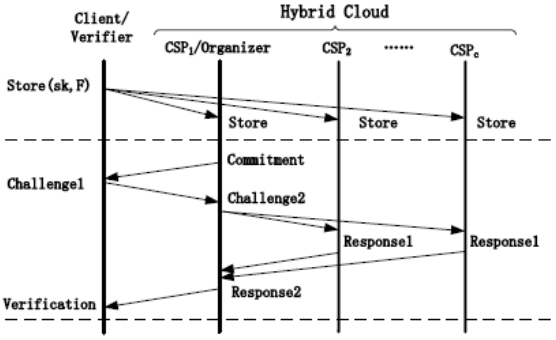

Fig.2: Flowchart of verification process

i) The organizer initiates the protocol and sends the commitment to the verifier.
ii) The verifier returns a challenge set of random index co-efficient pairs to the organizer.
iii) The organizer relays them into each Pi in P according to the exact position of each data block.
iv) Each Pi returns its response of challenge to the organizer.
v) The organizer synthesizes a final response from these 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.

a) Integrity audit services:

CPDP scheme is used to construct audit system architecture for outsourced data in hybrid clouds by using TPA as shown in Fig.1.

In this architecture, granted clients need to dynamically interact with CSPs to access or update their data for various application purposes. TPA, as trusted third party is used to ensure the storage security of outsourced data. TPA is reliable and independent and thus has no consideration to collude with either CSPs or users during the auditing process.

i. TPA makes regular checks on the integrity and availability of the delegated data at appropriate intervals.

ii. TPA will organize, manage and maintain the outsourced data instead of data owners and support dynamic data operations for the granted clients.

iii. TPA takes the evidences for the disputes about the inconsistency of data in terms of authentic records for all data operations.

IV. SECURITY ANALYSIS

To enable privacy preserving public auditing for cloud data storage under this architecture, our protocol design should achieve the following security and performance guarantee:

- *Public auditability*: to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data.
- *Verification correctness*: to ensure there exists no cheating CSP that can pass the audit from TPA without indeed storing users data intact.
- *Verification Transparency*: to enable TPA with secure and efficient auditing capability to cope with auditing delegations from possibly large number of different CSPs simultaneously.
- *Privacy-preserving*: to ensure that there exists no way for TPA to derive users data from the information collected during auditing process.
- *Lightweight*: to allow TPA to perform auditing with minimum storage, communication and computation overhead.

V. CONCLUSION

In this paper, we addressed the construction of collaborative integrity verification mechanism for the distributed data outsourcing in hybrid clouds. Based on the homomorphic verifiable responses and index property, we proposed a collaborative provable data possession scheme to support dynamic scalability on multiple cloud storage services providers. It also provides security properties required by the zero-knowledge interactive proof system so that it can resist various attacks even if it is deployed as a public verification service. More importantly, our solution conceals the details of outsourced storage to reduce the burden on verifiers and verifiers cannot even distinguish whether the verified data is in a hybrid cloud or a single cloud.

VI. REFERNCES

[1] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson,D.A., Rabkin, A., Stoica, I., Zaharia, M., 2010. A view of cloud computing.
Commun. ACM 53 (4), 50–58.

[2] Ateniese, G., Burns, R.C., Curtmola, R., Herring, J., Kissner, L., Peterson, Z.N.J., Song,D.X., 2007. Provable data possession at untrusted stores. In: Proceedings of the
2007 ACM Conference on Computer and Communications Security, CCS 2007,pp. 598–609.

[3] Ateniese, G., Pietro, R.D., Mancini, L.V., Tsudik, G., 2008. Scalable and efficient provable data possession. In: Proceedings of the 4th International Conference on
Security and Privacy in Communication Networks, SecureComm, pp. 1–10.

[4] Barreto, P.S.L.M., Galbraith, S.D., O'Eigeartaigh, C., Scott, M., 2007. Efficient pairingcomputation on supersingular abelian varieties. Des. Codes Cryptography. 42 (3),
239–271.

[5] Beuchat, J.-L., Brisebarre, N., Detrey, J., Okamoto, E., 2007. Arithmetic operators for pairing-based cryptography. In: Cryptographic Hardware and Embedded Systems
– CHES 2007, 9th International Workshop, pp. 239–255.

[6] Boneh, D., Boyen, X., Shacham, H.,2004. Short group signatures. In: In Proceedings of CRYPTO 04, LNCS Series. Springer-Verlag, pp. 41–55.

[7] Boneh, D., Franklin, M., 2001. Identity-based encryption from the weil pairing. In: Advances in Cryptology (CRYPTO'2001). Vol. 2139 of LNCS, pp. 213–229.

[8]Bowers, K.D., Juels, A., Oprea, A., 2009. Hail: a high-availability and integrity layer for cloud storage. In: ACM Conference on Computer and Communications Security,
pp. 187–198.

[9] Cramer, R., Damgård, I., MacKenzie, P.D., 2000. Efficient zero-knowledge proofs of knowledge without intractability assumptions. In: Public Key Cryptography, pp.
354–373.

[10] Dodis, Y., Vadhan, S.P., Wichs, D., 2009. Proofs of retrievability via hardness amplification. In: Reingold, O. (Ed.), Theory of Cryptography, 6th Theory of Cryptography
Conference, TCC 2009. Vol. 5444 of Lecture Notes in Computer Science. Springer, pp. 109–127.

[11] Erway, C.C., Küpc¸ ü, A., Papamanthou, C., Tamassia, R., 2009. Dynamic provable data possession. In: Proceedings of the 2009 ACM Conference on Computer and Communications Security, CCS 2009, pp. 213–222.

[12]Fu, K., Kaashoek, M.F., Mazières, D., 2002. Fast and secure distributed read-only file system. ACM Trans. Comput. Syst. 20 (1), 1–24. Goldreich, O., 2001. Foundations of Cryptography: Basic Tools. Vol. Basic Tools. Cambridge University Press.

[13] Hsiao, H.-C., Lin, Y.-H., Studer, A., Studer, C., Wang, K.-H., Kikuchi, H., Perrig, A., Sun, H.-M., Yang, B.-Y., 2009. A study of user-friendly hash comparison schemes. In: ACSAC, pp. 105–114.

[14] Hu, H., Hu, L., Feng, D., 2007. On a class of pseudorandom sequences from elliptic
curves over finite fields. IEEE Trans. Inform. Theory 53 (7), 2598–2605.

[15] Juels Jr., A., Kaliski, B.S., 2007. Pors: proofs of retrievability for large files. In: Proceedings
of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, pp. 584–597.

[16] Ko, R.K.L., Lee, B.S., Pearson, S., 2011. Towards achieving accountability, auditability and trust in cloud computing. In: Abraham, A., Mauri, J.L., Buford, J.F., Suzuki, J., Thampi, S.M. (Eds.), Advances in Computing and Communications. Vol. 193 of Communications in Computer and Information Science. Springer,Berlin/Heidelberg, pp. 432–444.

[17] Li, F., Hadjieleftheriou, M., Kollios, G., Reyzin, L., 2006. Dynamic authenticated index structures for outsourced databases. In: Chaudhuri, S., Hristidis, V., Polyzotis, N.(Eds.), SIGMOD Conference. ACM, pp. 121–132.

[18] Ma, D., Deng, R.H., Pang, H., Zhou, J., 2005. Authenticating query results in data publishing. In: Qing, S., Mao, W., Lopez, J., Wang, G. (Eds.), ICICS. Vol. 3783 of
Lecture Notes in Computer Science. Springer, pp. 376–388.

Schnorr, C.-P., 1991. Efficient signature generation by smart cards. J. Cryptol. 4 (3), 161–174.

[19]Shacham, H., Waters, B., 2008. Compact proofs of retrievability. In: Advances in Cryptology – ASIACRY, 2008, 14th International Conference on the Theory and
Application of Cryptology and Information Security, pp. 90–107.

[20] Tchifilionova, V., 2011. Security and privacy implications of cloud computing c lost in
the cloud. In: Camenisch, J., Kisimov, V., Dubovitskaya, M. (Eds.), Open Research Problems in Network Security. Vol. 6555 of Lecture Notes in Computer Science. Springer, Berlin/Heidelberg, pp. 149–158.

[21]Wang, C., Wang, Q., Ren, K., Lou, W., 2010. Privacy-preserving public auditing for data storage security in cloud computing. In: INFOCOM, 2010 Proceedings IEEE,
pp. 1–9, 14-19.

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

[23] Xie, M., Wang, H., Yin, J., Meng, X., 2007. Integrity auditing of outsourced data. In: Koch, C., Gehrke, J., Garofalakis, M.N., Srivastava, D., Aberer, K., Deshpande, A., Florescu, D., Chan, C.Y., Ganti, V., Kanne, C.-C., Klas, W., Neuhold, E.J. (Eds.), VLDB. ACM, pp. 782–793.

Yavuz, A.A., Ning, P., 2009.

[24] Baf: An efficient publicly verifiable secure audit logging scheme for distributed systems. In: ACSAC, pp. 219–228.

[25] Yumerefendi, A.R., Chase, J.S., 2007. Strong accountability for network storage. ACM Trans. Storage (TOS) 3 (3).

[26] Yan Zhua,b,∗, Hongxin Huc, Gail-Joon Ahnc, Stephen S. Yauc ,.Efficient audit service outsourcing for data integrity in clouds.