International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol.2, No.6, December 2013, pp. 429~437
ISSN: 2089-3337

INTEGRITY VERIFICATION IN MULTI-CLOUD BY USING PROVABLE DATA POSSESSION AND ALERT NOTIFICATION

Ms.V.Mangaiyarkkarasi *, Mr.K.A.Dhamodaran **
* M.E. (CSE) Second Year, Erode Sengunthar Engineering College, Erode
** Faculty of Computer Science & Engg Dept, Erode Sengunthar Engineering College, Erode

Article Info

Article history:

Received Jan 16th, 2014
Revised Aug 20th, 2014
Accepted Aug 26th, 2014

Keyword:

Provable Data Possession
Multiple Cloud
Data Compression
Coorperative

ABSTRACT

Integrity verification in cloud computing can be ensured by using a technique called Provable Data possession (PDP). In PDP model, it allows client to store data at an untrusted server and to verify that the server possesses the original data without retrieving it. It preprocesses the data and then send it to an untrusted server, by keeping a small amount of meta-data. The efficient PDP can be constructed for multi-cloud storage to support scalability of service and data migration. In this paper, multiple storage service providers cooperatively store and maintain client's data based on homomorphic verifiable response and hash index hierarchy. The cooperation between the multiple service providers can be controlled and maintained by Trusted Third Party (TTP). In addition, the Data Compression technique is used to compress the data and reduce the storage cost for clients. The alert notification is automatically sent to client when any malicious attack of client's data is made by service provider along with the timing information.

Corresponding Author:

Ms.V.Mangaiyarkkarasi,
M.E. (CSE) Second Year,
Erode Sengunthar Engineering College, Erode
Email: mangaikumaran@yahoo.com

1. INTRODUCTION

Cloud storage service is a rapid growing field as it provides a low cost, archiving, backup, scalable and position independent platform to store client's data. Cloud computing have an open architecture and interfaces it have capability to incorporate multiple storage service and provide high interoperability. This distributed cloud environment is called multi-cloud. This interface abstracting away the complexities of direct hardware management. But this eliminates the direct oversight of client data and hence reliability and security at high service-level requirement is expected. The clients need to assure that their data stored in untrusted server have not been tampered or deleted. Storage outsourcing of very large files to remote servers have constraints. (i.e) Client should not download all the stored data to validate as it may prohibitive in bandwidth and time if they check frequently. The security assurance can be eroded by two basic approach to client verification of file availability and integrity.

The first approach is Provable Data Possession (PDP)[2], in which data is preprocessed by the client and the metadata for verification is produced. The data is sent to an untrusted storage server and client can delete their local file. The client can keep some secret information to check server's response. The server have to prove the client that the data sent by them is not get tampered by responding to the challenges created by client.

The second approach is Proofs of Retrievability (POR)[3] is a challenge-response protocol it enables the service provider to demonstrate the client that data is retrievable(i.e) data can be recoverable without any loss or corruption.

Virtual Infrastructure Management(VIM) [1], is a multi-cloud which allows clients to access their resources remotely through interfaces. Various tools and technologies for multicloud is Platform VM Orchestrator, VMware,vSphere, and Ovirt. It helps to construct a distributed cloud storage platform (DCSP). The low-cost, scalable, location independent platform for distributed file systems can be provided by, for

example, Apache Hadoop Distribution File System (HDFS)[15], Google File System(GFS), Amazon S3 File System etc. They have some similar features:

1. A single metadata server provides centralized management by a global namespace;
2. Files are split into blocks or chunks and stored on block servers;
3. 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 difficult to efficiently verify the integrity and availability of stored data for detecting faults and automatic recovery. This verification is necessary because it automatically maintaining multiple copies of data and automatically redeploying processing logic if any failure occurs.

There are schemes to make a decision for data possession without downloading data from untrusted stores, it is not suitable for a distributed cloud storage environment because it is not constructed on interactive proof system.

Dynamic Provable Data Possession (DPDP)[5] is an extension of PDP model to support provable updates on the stored data. A file may contain n blocks, and updation can be insertion of a new block or modification of an existing block or deletion of any block. DPDP solution is based on authenticated dictionaries, where rank information is used for dictionary entries. It also supports data possession for hierarchical file system. It is based on Merkle Hash tree (MHT) but it not provide any algorithms for constructing distributed Merkle trees which is needed for multi-cloud environment. If clients ask for any particular client file block, then the server needs to send that file block with a proof of intactness. Here the communication overhead in a multi-cloud occurs, because server in one cloud needs to generate proof with the help of other cloud storage services, where the adjacent blocks is getting stored.

The other schemes like CPOR-I, and CPOR-II [6] are constructed based on homomorphic verification tags, here the server can generate tags for multiple file blocks with a single response value. If there is no homomorphic responses, then clients must invoke the PDP protocol repeatedly to check the integrity of file blocks stored in multiple cloud servers. The clients should know the exact position of each file block in a multi-cloud and it increase high communication overheads and computation costs for clients.
Therefore the new scheme cooperative PDP model will reduce the storage and network overheads and increase the transparency of verification activities in multi-cloud systems. Its main features is detecting abnormality and renewing multiple copies of data. The existing scheme address public verifiability[2], dynamics [5], scalability [4], and privacy preservation [7]. There are two main attacks,

1. **Data Leakage Attack** - after running certain verification communication the adversary can easily obtain the stored data.
2. **Tag Forgery Attack -** by which a dishonest CSP can deceive the clients.

These two attacks may cause potential risks for privacy leakage and ownership cheating. A verification scheme for data integrity in distributed storage environments should have the following features:

· **Usability aspect**: A client should utilize the integrity check in the way of collaboration services. The scheme should conceal the details of the storage to reduce the burden on clients.

· **Security aspect**: The scheme should provide adequate security features to resist some existing attacks, such as data leakage attack and tag forgery attack.

· **Performance aspect**: The scheme should

Scheme	**Type**	**CSP Comp.**	**Client Comp.**	**Comm.**	**Frag.**	**Privacy**	**Multiple Clouds**	**Prob. Of Detection**
PDP[2]	$HomT$	$O(t)$	$O(t)$	$O(1)$		✓	#	$1-(1-\rho)t$
SPDP[4]	MHT	$O(t)$	$O(t)$	$O(1)$	✓	✓		$1-(1-\rho)t{\cdot}s$
DPDP-I[5]	MHT	$O(t \log n)$	$O(t \log n)$	$O(t \log n)$		✓		$1-(1-\rho)t$
DPDP-II[5]	MHT	$O(t \log n)$	$O(t \log n)$	$O(t \log n)$				$1-(1-\rho)\Omega(n)$
CPOR-I[6]	$HomT$	$O(t)$	$O(t)$	$O(1)$			#	$1-(1-\rho)t$
CPOR-II[6]	$HomT$	$O(t+s)$	$O(t+s)$	$O(s)$	✓		#	$1-(1-\rho)t{\cdot}s$
CPDP	$HomR$	$O(t + c \cdot s)$	$O(t+s)$	$O(s)$	✓	✓	✓	$1-\Pi Pk\in\mathcal{P}(1-\rho k)rk{\cdot}t{\cdot}s$

Table 1 : Comparision of various schemes for file consisting of n blocks

2 RELATED WORKS

The related works of this model under the various scenario is as follows,

2.1 Information dispersal:

The distributed information dispersal algorithms(IDA) tolerate the Byzantine server in both synchronous network and asynchronous one. In this algorithm file integrity is enforced in the server. It protects the client from sending the inconsistent data shares to different servers. HAIL[9] protocol provides a file-integrity checking of clients oor external service and avoid the communication among servers. It also provides the assurance of granularity of full file.

2.2 Universal Hash Functions:

It is also termed as algebraic signatures or homomorphic fingerprinting. It is used to construct the message-authentication codes(MAC). When pseudorandom functions(PRFs) is used along with UHF it can yield MACs. MAC can be aggregated over many data blocks and thus it support compact proof over large file samples.

2.3 PDP:

The PDP[2] scheme is an optimal protocol for the static case that achieves O(1) costs for all the complexity measures and it uses RSA-based scheme. This protocol lack in either they require expensive server computation or communication over the entire file, linear storage for the client, or do not provide security guarantees for data possession. It is based on cryptographic hash function and symmetric key encryption, but the servers can deceive the owners by using previous metadata or responses due to the lack of randomness in the challenges Dynamic scenario is insecure due to replay attacks. To avoid replay attacks, an authenticated tree structure that incurs logarithmic costs must be employed and thus constant costs are not feasible in a dynamic scenario.

2.4 DPDP:

Dynamic PDP(DPDP)[5] schemes uses a hash function tree to realize $O(\log n)$ communication and computational costs for a n-block file. The basic scheme, called DPDP-I, which retains the drawback of Scalable PDP, and in the 'blockless' scheme, called DPDPII, the data blocks can be leaked by the response of a challenge when selecting a random challenge value. Moreover, these schemes are not effective for a multi-cloud environment because the verification path of the challenge block cannot be stored completely in a cloud [8].

2.5 POR:

It is the static archival storage of large files and its effectiveness rely on preprocessing steps that the client conducts before sending a file F to the server: "sentinel" blocks are randomly inserted to detect corruption, F is encrypted to hide these sentinels, and error-correcting codes are used to recover from corruption[10]. This codes improve the error-resiliency of their system. But it prevent any efficient extension to support updates, beyond simply replacing F with a new file F′. The number of queries a client can perform is limited and fixed.

2.6 CPOR:

Compact POR [6], is an improved version of POR. It come up with all future challenges during setup and store pre-computed answers as metadata. The metadata at the client, or at the server as in authenticated and encrypted manner. In this approach, the number of updates and challenges a client can perform is limited and fixed a priori. Also, one cannot perform block insertions anywhere (only append-type insertions are possible). Then for each update requires re-creating all the remaining challenges, which is problematic for large files. It works in the random oracle model.

2.7 RAID:

It is a distributed cryptographic system that allows a set of servers to solve the PDP problem.This system is based on an integrity-protected error correcting code (IP-ECC)[8], which improves the security and efficiency of existing tools, like POR. 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 a cloud storage.

Organization: The rest of this paper is organized asfollows. In Section 3, the formal definition of CPDP and the underlying techniques, which are utilized for the construction of CPDP scheme is described. In section 4, the details of cooperative PDP scheme for multicloud storage is described. In section 5, the security analysis for CPDP scheme is discussed. Then the section 6 which concludes this paper.

3. STRUCTURE AND TECHNIQUES

In this section, verification framework for multi-cloud storage and a CPDP definition is discussed. Two fundamental techniques for constructing CPDP scheme,

1.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.

2.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.

3.1. Verification Framework for Multi-Cloud

The existing PDP schemes provide a publicly accessible remote interface for checking and managing the large amount of data, but it is incapable to satisfy the inherent requirements from multiple clouds for communication and computation costs. A data storage service involves three different entities:

1. **Clients-** Have a large amount of data to be stored in multiple clouds and have the permissions to access and manipulate stored data
2. **Cloud Service Providers** (CSPs)-They work together to provide data storage services and have enough storages and computation resources
3. **Trusted Third Party (TTP)-** They are trusted one to store verification parameters and offer public query services for these parameters.In the verification architecture, there exists a multiple CSPs to cooperatively store and maintain the clients' data.Cooperative PDP is used to verify the integrity and availability of their stored data in all CSPs. The verification procedure is,
   - 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
   - Then thay transmits the file and some verification tags to CSPs, and may deleteits 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.

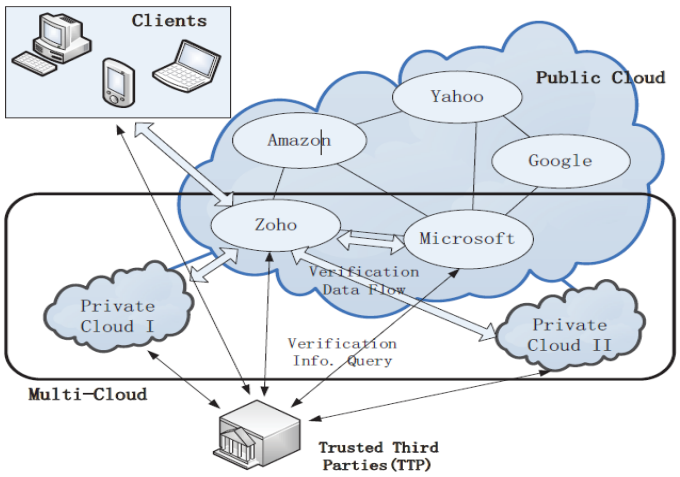

Fig. 1. Verification architecture for data integrity.

Neither we assume the CSP nor we assume the data owner to be a trusted one after error has been found. A TTP server is constructed as a core trust base on the cloud for the sake of security. TTP is not directly involved in the CPDP scheme in order to reduce the complexity of cryptosystem. TTP is reliable and independent through several functions [12],

- To setup and maintain the CPDP cryptosystem
- To generate and store data owner's public key
- To store the public parameters for verification protocol in the CPDP scheme.

3.2. Definition of Cooperative PDP-To prove the integrity verification in a multi-cloud environment, CPDP is based on interactive proof system (IPS)[12] and multi-prover zero-knowledge proof system (MPZKPS)[11].

- **Cooperative-PDP:** A cooperative provable data possession $\mathcal{S} = (KeyGen, TagGen, Proof)$ is a collection of two algorithms $(KeyGen, TagGen)$ and an interactive proof system $Proof$.
- ***KeyGen* (1^k):** It takes a security parameter κ as input, and returns a secret key sk or a public-secret keypair(pk, sk);
- ***TagGen* (sk, F,):** It takes as inputs a secret key sk, a file F, and a set of cloud storage providers $\mathcal{P} = \{Pk\}$, and returns the triples $(\zeta,, \sigma)$, where ζ is the secret in tags, $\psi = (u,\mathcal{H})$ is a set of verification parameters u and an index hierarchy $\mathcal{H}$ for F, $\sigma = \{\sigma(k)\}_{Pk\in\mathcal{P}}$ denotes a set of all tags, (k) is the tag of the fraction $F(k)$ of F in Pk;
- ***Proof* ($\mathcal{P}$, V):** is a protocol of proof of data possession between CSPs ($\mathcal{P} = \{Pk\}$) and a verifier (V) where each Pk 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 are 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,$\Sigma_{Pk\in\mathcal{P}}$ denotes cooperative computing in $Pk \in \mathcal{P}$.

3.3. Hash Index Hierarchy for CPDP-To support distributed cloud storage, has a hierarchy structure which resembles a representation of file storage. The hierarchical structure $\mathcal{H}$ consists of three layers to represent relationships among all blocks for stored resources. They are,

- **Express Layer**: It provides an abstract representation of the stored resources
- **Service Layer**: It offers and manages cloud storage services
- **Storage Layer**: It realizes data storage on many physical devices

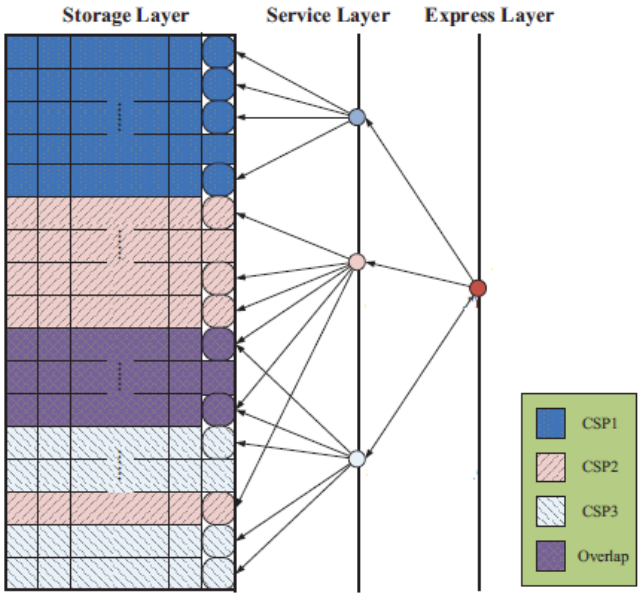

Fig. 2. Index-hash hierarchy of CPDP model.

It is a simple hierarchy to organize data blocks from multiple CSP services into a large size file by shading their differences among these cloud storage systems. In Fig.2. the resources in Express Layer are split and stored into three CSPs, that are indicated by different colors, in Service Layer. Each CSP fragments and stores the assigned data into the storage servers in Storage Layer. Different colors used to distinguish different CSPs. Storage Layer provides a special functions for data storage and management, e.g., there may exist overlaps among data blocks and discontinuous blocks but these functions may increase the complexity of storage management. It also defines a common fragment structure that provides probabilistic verification of data integrity for outsourced storage. The fragment structure is a data structure that maintains a set of block-tag pairs, allowing searches, checks and updates in (1) time.
Implementation of the CPDP framework in the multiple clouds involves:

- A file is split into $n \times s$ sectors and each block corresponds to a tag, so that the signature tags can be reduced by the increase of s;
- A verifier can verify the integrity of file in random sampling approach, which is of utmost importance for large files;
- It rely on homomorphic properties to aggregate data and tags into a constant size response, which minimizes the overhead of network communication
- The hierarchy structure provides a virtualization approach to conceal the storage details of multiple CSPs.

3.4. Homomorphic Verifiable Response for CPDP- When provable data possession is considered as a challenge-response protocol, Homomorphic Verifiable Responses(HVR), is used to integrate multiple responses from the different CSPs in CPDP scheme. 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 $Qi \cup$. Homomorphic verifiable response is the key technique of CPDP because it not only reduces the communication bandwidth, but also conceals the location of outsourced data in the distributed cloud storage environment.

4. COOPERATIVE PDP SCHEME

In this section, CPDP scheme for multicloud system based on the above-mentioned structure and techniques is proposed. It is constructed on collision-resistant hash, bilinear map group, aggregation algorithm, and homomorphic responses.

4.1. Notations and Preliminaries Collision-Resistant Hash:

A hash family $\mathbb{H}$ is (t, ϵ)-collision-resistant if no t-time adversary has advantage at least ϵ in breaking collision resistance of $\mathbb{H}$. The bilinear map with properties like 1) Bilinearity[17] 2) Non-degeneracy 3) Computability is efficiently computable.

4.2. CPDP Scheme

A CPDP scheme is a collection of two algorithms ($KeyGen$, $TagGen$) and an interactive proof system $Proof$. This scheme is described as follows,

- .$KeyGen$- which obtain the public/private key pairs for CSPs and users.
- $TagGen$- where the clients generate the tags of outsourced data
- $Proof$- 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[11] 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 Q to the organizer;
    3. The organizer forwards them to each Pi in $\mathcal{P}$ according to the exact position of each data block;
    4. Each Pi returns its response of challenge to the organizer;
    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 cloud environment, CPDP differs from two aspects:

- Tag aggregation algorithm: In stage of commitment, the organizer generates a random set and returns its commitment to the verifier. This assures that the verifier and CSPs do not obtain the value. This guarantees only the organizer can compute the final value by using response that is received from CSPs. After final value is computed, it should be transfer to the organizer in stage of "Response1". To ensure the security of transmission of data tags, El-Gamal encryption[14] method is used.
- Homomorphic responses: Because of the homomorphic property, the responses computed from CSPs in a multi-cloud can be combined into a single final response

5. SECURITY ANALYSIS

The security analysis of CPDP scheme is constructed from multi-prover zero-knowledge proof system (MPZKPS)[11],which satisfies the following properties of given assertion *L*:

1. **Completeness** - whenever $x \in L$, provers that convinces the verifier
2. **Soundness** - whenever $x \not\in L$, provers will not convince the verifier that $x \in L$;
3. **Zero-knowledge** - no cheating verifier can learn anything other than the veracity of the statement.

These properties can protect from various attacks such as data leakage attack (privacy leakage) and tag forgery attack (ownership cheating). Along with the MPZKPS properties the alert notification property for verification of security is also included.The various security analysis is described as.

5.1. Collision resistant for index-hash hierarchy

In CPDP scheme, the collision resistant for index-hash hierarchy is based on random oracle model. Hash function is a collision resistant model,but it also produce a forged tag when the same hash value is reused multiple times, e.g., a true client modifies the data or repeats to insert and delete data blocks of outsourced data. It can be overcome by using indexhash hierarchy by providing a collision resistant, in which client generates $\sqrt{2p.ln.\frac{1}{1-\varepsilon}}$ files with same file name and cloud name. The client can repeat this $\sqrt{2^{L+1}.ln.\frac{1}{1-\varepsilon}}$ to modify, insert and delete the data blocks.

5.2. Completeness property of verification

In CPDP scheme, the completeness property is a public verifiability property, which allows anyone to challenge the cloud server for *data integrity* and *data ownership* without the need for any secret information. In this process, anyone can obtain the owner's public key and the corresponding file parameter from TTP to execute the verification protocol, hence this is a public verifiable protocol. Moreover, for different owners, the secrets values hidden in their public key are also different, determining that a success verification can only be implemented by the real owner's public key. In addition, the file parameter is used to store the file-related information, so an owner can employ a unique public key to deal with a large number of outsourced files.

5.3. Zero-knowledge property of verification

In CPDP scheme, a Multi-Prover Zero-knowledge Proof (MP-ZKP) system [11], is an extension of an interactive proof system (IPS). In MP-ZKP, a polynomial-time bounded verifier interacts with several provers whose computational powers are unlimited. In a *Simulator* model, in which every cheating verifier has a simulator that can produce an interaction between a honest prover and a cheating verifier. Zero-knowledge is a property that achieves the CSPs' robustness against attempts to gain knowledge by interacting with them. In CPDP, zero-knowledge property preserves the privacy of data blocks and signature tags. Randomness is adopted into the CSPs' responses in order to resist the *data leakage attacks.*

This means that the cheating verifier cannot obtain any information from random integer and randomize verification tag.

5.4. Knowledge soundness of verification

In CPDP scheme, it has the knowledge soundness property which make use of $\mathcal{P}*$ to construct a knowledge extractor $\mathcal{M}$, which gets the common input and rewindable blackbox accesses to the prover $P*$, and then attempts to break the computational Diffie-Hellman (CDH) problem in $\mathbb{G}$: given G,$G1$ = Ga,$G2$ = $Gb \in R\ \mathbb{G}$, output $Gab \in \mathbb{G}$. But it is unacceptable because the CDH problem is regarded as an unsolved problem in polynomial-time. Soundness means that it is infeasible to fool the verifier to accept false statements. It is also regarded as a stricter notion of unforgeability for file tags to avoid cheating the ownership. This means that the CSPs, even if collusion is attempted, cannot be tampered with the data or forge the data tags if the soundness property holds. Thus in CPDP scheme, it can resist the *tag forgery attacks* to avoid cheating the CSPs' ownership.

5.5. Alert notification for verification

In CPDP scheme, along with MPZKPS property, the another property for security analysis is alert notification for verification of security analysis. In this alert notification for CPDP scheme, whwn any cloud storage providers try to deceive the clients by tampering or deleting the client's data then the alert notification along with timing information is automatically sent to client. Hence the client can be make sure that their data is successfully stored in server and if any malpractice is made by the service providers they will get the notification.

6. CONCLUSION

The new cooperative PDP scheme has been designed mainly for distributed cloud storage. It is an efficient PDP because it is based on homomorphic verifiable response and a hash index hierarchy model to support multiple storage servers with dynamic scalability. The compression technique provides an effective storage cost for clients. The notification to clients during any modification made by cloud service providers or third party auditor made clients to be confident that their data is not get tampered or deleted by anyone and it has been successfully stored in cloud. The security of this scheme is enhanced by ensuring all zero knowledge properties and hence it resist from various attacks even it can be deployed as public cloud. This cooperative PDP can be good source for data integrity verification in distributed cloud storage. In future work, the more effective CPDP can be constructed. There are three main problem in constructing the CPDP. First is it is affected by bilinear mapping operation because of its high complexity. It can be overcome by RSA algorithm but existing RSA scheme have too many restrictions on performance and security. Second is, it is difficult to integrate the CPDP scheme with the existing system. Because in cluster network model it is difficult to dynamically update the CPDP parameters. Third is, generation of tags with length irrelevant to size of data blocks. It can be overcome by supporting the variable-length block verification.

REFERENCES (10 PT)

[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 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] 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.

[14] 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.

[15] 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/