Advanced Cooperative Provable Data Possession based Data Integrity Verification for Multi-Cloud Storage

Vaibhav Bharati
PG Student, Computer Dept.,
SKNCOE, Pune,
University of Pune

M. R. Patil
Professor, Computer Dept.,
SKNCOE, Pune,
University of Pune

ABSTRACT
Cloud computing is an emerging field which is raising its importance in the field of storage of data. Every individual, firm or organization wants to store data that has the limited storage capacity. For that purpose, there is need to outsource data to some Cloud Service Provider, there arises a problem of security of data. Security comes with 3 main parameters confidentiality, integrity and availability. For these issues, there is term called Provable Data Possession, it is nothing but the proof given by service provider to the data owner when there is demand by owner. Paper proposes an effective PDP model for integrity verification of data on distributed multi-cloud storage by using web-servers and Trusted Third Party. Here proposed the use of techniques such as, Advanced Cooperative Provable Data Possession (Advanced CPDP), Homomorphic Verifiable Response and Hash Index Hierarchy for Multiprover Zero Knowledge Proof System which is an Interactive Proof System.

General Terms
Security, Data Integrity on multi cloud

Keywords
Multi-cloud storage, Web Server based Multi-cloud, Advanced Cooperative Provable Data Possession, Zero Knowledge Property, Hash Index Hierarchy, Homomorphic Verifiable Response.

1. INTRODUCTION

In this paper, concept focussed on the need of security of owner's data which is outsourced at the multiple cloud based web servers. For outsourcing the data the most preferable technology is the cloud. Cloud has services on the basis of pay per use. As data owner outsources data to the CSP, the issue of the security arises because misuse of the outsourced data can be done by CSP itself. There are many ways to prevent misuse of data at the small scale like cryptography.

Security is nothing but the intersection of confidentiality, integrity and availability, failing of anyone parameter to achieve leads to data to become vulnerable. Here all the parameters are assessed but the focus is mostly on the issue of integrity. Confidentiality can be achieved through the authentication while the availability can be achieved through the redundant storage of the data on multiple cloud based web servers.

The main aim is integrity, which can be achieved by the technique called PDP. In this technique, the data owner challenges to the CSP for providing the guarantee of the integrity of the outsourced data. The challenge is in the form of query. Query contains some credentials of the uploaded data calculated before uploading. Then the CSP calculates the credentials according to the challenge of the data owner in the form of response, the response is given to the owner. Owner crosschecks the original credentials with the response, if equality holds the owner is satisfied with the provided service and integrity the outsourced data. As mentioned above is already existing technique as per paper [7]. But the overhead and the role of the Data Owner is reduced by inducing the third actor in the picture as TTP. All the above stuff taken place in communication in between CSP and Data Owner is happens through TTP.

In further sections, the dynamic storage of the data on the multiple cloud based web-servers as a distributed storage has been discussed.

2. RELATED WORK

2.1 Survey
All material An Ateniese, et al [2] proposed a PDP model which supports problems of static files. This model works optimally for static case with constant complexity by the principle of blocks and tags. It gives the private authentication and the linear storage for Data Owner. But, this model has some limitations which are detected by later results as, expensive server computation, no security guarantee for data possession, vulnerable to replay attacks.

A. Juels, et al [3] proposed a POR model which also works on static storage with the constant complexity. This varies with the previous model in the processing on the data. Data is modified by inserting some sentinel blocks in between which are used to verify the integrity of the file by checking the correctness of sentinel blocks. This model has both communication and computation complexities constant. It has some limitations on number of times one can challenge for integrity as sentinels are one time labels. Also, the model depends on the large preprocessing of data.

Shacham and Waters [4] proposed Compact POR model which is an improved version of POR. This model uses homomorphic property to aggregate a proof of challenge with authenticity complexity of O(1) and with computation

complexity of O(t). In this model also some limitations with respect to current models as its solution is for static storage and vulnerable to the leakage of data in verification.

Table 1. Evaluation of Related Work

Algorithm	Description	Evaluation
PDP (Anteniese, et al.)	Ensuring possession of files on untrusted storages and provided an RSA-based scheme for a static case that achieves the O (1) communication cost.	Insecure against replay attacks in dynamic scenarios Do not fit for multi-cloud storage
POR **(A. Juels, et al.)**	Publicly verifiable version, which allows anyone to challenge the server for data possession, and greatly extends application areas of PDP.	Relies largely on pre-processing that the Data Owner conducts before sending a file to a CSP
Compact POR (Shacham and Waters)	Uses homomorphic property for aggregating a proof in authenticator value with O (1) and t challenge blocks O (t) computation cost	Supports only for static data and could not prevent the leakage of data blocks in the verification
Scalable PDP (Ateniese, et al.)	Suitable for the limited dynamic nature and Require pre-computed answers as metadata which allows limited and fixed a priori number of updates and challenges.	Requires lot off pre computations to improve the performance and Supporting only append type insertions.
DPDP **(C. Erway, et al.)**	Based on PDP model for dynamic files which can be updated online.	Complexity of the order of O (log n)
Improved DPDP **(Feifei Liu, et al.)**	Improved the model based on DPDP model, and reduces the computational and communication complexity to constant.	-----
Cooperative PDP [1] **(Yan Zhu, et al.)**	Provable data possession in distributed cloud environments from the aspects: high security, transparent verification, and high performance.	Model is evaluated on simulator by using hadoop file system.

All the above techniques are useful for the static storage only. Ateniese, et al [5] proposed Scalable PDP which is the first technique to give the dynamicity in the outsourcing of data. This model works on the principle of the random oracle model. It takes the pre-computed answers as metadata so it limits on the number of times the updates and the challenges can possible. This model supports only append operation on data and does not provide the in between modifications in the already uploaded data, which also puts limitations on the dynamicity of the model.

C. Erway, et al [6] proposed a DPDP model on the basis of PDP model for dynamic storage of files and also which can be updated online. Even after this modification one can verify the integrity of the file. It uses the skip-lists for maintaining tags and is stored at Data Owner side to avoid replay attacks. This model has computational and communication complexity both up to log(n). The server requires the whole path of data block to access it, because this model does not maintain the numbering to the data blocks. In this case, if the file is too large then both the above complexities are considerable.

Feifei Liu, et al [7] proposed Improved DPDP model which is the improvement over the previous model [6]. It does partition of original file into blocks. Tag is generated for each block and hash value is computed for each tag. These tags are used to verify the integrity of the file by using the skip-lists, and the integrity of the tags is verified by the hash values. In this model computational and communication complexities are reduced from log(n) to constant.

2.2 Motivation

In cloud computing scenario data owner outsource data on cloud. CSP stores that data on the single available server, but there can be the chances of crashing the server. It leads into loss of valuable data which threatens the availability. Also there are chances of directly or indirectly corruption of owner's data. CSP can access the valuable data and also do the misuse of it, which threatens the integrity of data. So this problem can be solved by storing data on multiple clouds. The proposed Advanced CPDP model is totally based on the CPDP Model [1] along with enhancement in the form of real implementation with replacing simulator.

3. PROPOSED MODEL

3.1 Model

The proposed system will make use of multi cloud storage to upload data on the cloud. Loss of the data during corruption of any server can be prevented by retrieving it from other servers. Here the concept of distributed file system is used in the form of web servers. Data owner will request the CSP for the space for data storage; CSP will check the availability of space and then store that data on the cloud, but there is no guarantee for security of data stored on the cloud.

As data owner cannot fully trust the CSP, TTP facilitates guarantee of security of outsourced data. In this system the two algorithms are to be used, first is Multiprover ZKPS which supports completeness, knowledge soundness and zero-knowledge properties and second one is CPDP which supports dynamic scalability storage of data on multiple servers cooperatively based on HIH [1] and HVR [1] with high security, transparent verification, and high performance.

Fig. 1 shows the architecture of the Advanced CPDP model. There are three main actors as Data Owner, CSP and the TTP. Data Owner calculates the data credentials in the form of tags and stores them on TTP before uploading the data on the central CSP. The integrity of tags is maintained by the technique of HIH. Then central CSP partitions the original whole data on multiple storages which are in the form of cloud. Each cloud has separate CSP to maintain the data storage and security. The security maintained by the CSP in the form of confidentiality, availability and integrity is monitored by the TTP. TTP has the whole access of data stored on the cloud. It keeps the overlook on stored data to prevent any change in contents and provide the security.

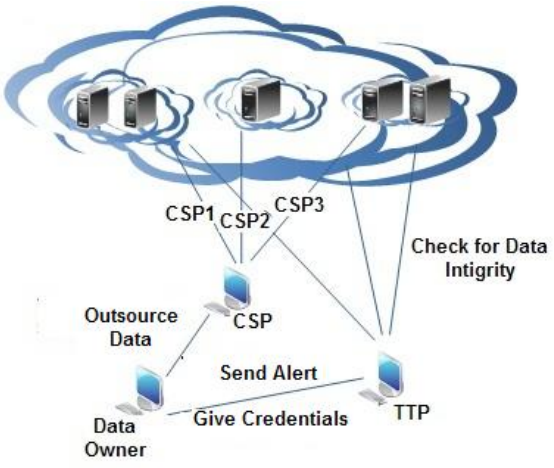

Fig 1: Architecture Diagram of Advanced CPDP Model

There is any change in the data on the clouds at any instant; the CHALLENGE is done by TTP in the form of query. On that TTP calculates the RESPONSE containing the credentials as tags. As in this model there are multiple clouds, all calculated RESPONSEs are in the heterogeneous form. All these responses have to merge such a way that to form a single homomorphic RESPONSE and which is done by HVR technique. Then TTP verifies the RESPONSE generated by HVR technique and the credentials stored initially on it. If there is mismatch in the comparison, then the integrity of the data is compromised, else the data is safe with its integrity. If the data loses its integrity TTP generates the alert in the form of automated mail to notify to the Data Owner.

3.2 Advanced CPDP Algorithm

Step 1: Input file F

Step 2: Using Hash Index Hierarchy concept to split the file.

Step 3: Split file into number of blocks {m1, m2, m3…}

Step 4: Generate key [7] for each block

$KeyGen(1^k) \rightarrow \{s_k, p_k\}$

Compute, $s_k = k_1$, $k_1 \leftarrow \{0, 1\}^k$,

$p_k = (N, g)$

Where, N=p*q, product of two primes

g=high order in Z_N

Table 2. Symbols in Model

Symbol	Representation
N	number of blocks in file
S	number of sectors in each block
t	number of index coefficient pairs in a query
c	number of clouds to store a file
F	file with n*s sectors; F= $\{m_{i,j}\}$, i ∈ [1,n], j ∈ [1,s]
σ	set of tags; σ = $\{\sigma_i\}$, i ∈ [1,n]
H	Set of hash values; H = $\{h_i\}$, i ∈ [1,n]
V	set of number of times i^{th} block is modified; i ∈ [1,n]
Q	set of index coefficient pairs; Q = $\{(i, v_i)\}$, i ∈ [1,t]
Θ	response for the challenge Q

Step 5: Generate Tag [7] for each block

$TagBlock(s_k, p_k, m_i, v_i, i) \rightarrow \{T_i, h_i\}$

Compute, $T^*_i = g^{mi} \bmod N$

$H^*_i = H_{k1}(T_i \parallel f(v_i) \parallel i)$

Where, H=cryptographic hash function

f = pseudo random function

Step 6: Store all the credentials in the form of tags and hash on the TTP before uploading the data on clouds..

Step 7: Response calculation by TTP from clouds:

Challenge: ***Query(c)*** ➔ ***chal*** [7]

i. Input: number of blocks to be challenged, *b*
ii. *k2, k3* ➔random numbers.
index: $i_j = \pi_{k2}(j)$ *for* $1 \le j \le b$
Coeff.: $a_j = f_{k3}(j)$; *π= pseudo random permutation*
iii. Output: $chal = \{(i_1, i_2, \ldots, i_b), (a_1, a_2, \ldots, a_b)\}$

Proof: ***Prove(chal, F)*** ➔ ***P*** [7]

i. Input: Query chal, File F
ii. Search h_{ij}, T_{ij}, m_{ij} For $i_j \in \{i_1, i_2, \ldots, i_b\}$

Use the HVR concept [1] to integrate the responses from multiple clouds.

Homomorphism mapped as, f: P ➔ Q
Where, P and Q are two different groups
Such that,
$f(g_1 \oplus g_2) = f(g_1) \otimes f(g_2)$; for all g1, g2 ∈ P
Where, $\oplus$:= operation in P
$\otimes$:= operation in Q

The above homomorphism concept is used for calculating Homomorphic Verifiable Tags such that, T_i and T_j for messages m_i and m_j respectively, then by combining T_i and T_j get T' for the $m_i + m_j$.

By working on the same principle the HVR can be calculated from different responses from different clouds in CPDP scheme and stored at TTP.

Compute, $M=a_1.m_{i1}+a_2.m_{i2}+.....+a_c.m_{ib}$
$h=h_{i1}.h_{i2}...h_{ib}$
iii. Output: $P= \{M, h, (T_{i1},T_{i2},....,T_{ib})\}$

Step 7: Verify the collective response with the RESPONSE stored at TTP.

Verification: $Verify(s_k, chal, P, \Omega) \rightarrow \{Accept, Reject\}$ [7]

i. Compute, $T^* = g^m \bmod N$
$T=T_{i1}^{a1} * T_{i2}^{a2} * .... * T_{ib}^{ab}$.
Search, $\Omega=\{v_{i1}, v_{i1}, ..., v_{ib}\}$ for $i_1....i_b$
$H=H_{k1}(T_{i1}||f(v_{i1})||i_1)*...*H_{k1}(T_{ib}||f(v_{ib})||i_b)$
ii. Check,
$T^* ?= T$, & $H^*? = h$
Where, T^* and H^* are nothing but the original credentials.
If both holds, output Accept;
Else, output Reject.

Step 8: If there is mismatch, generate alerts.

Step 9: Otherwise, SUCCESS.

4. CONTRIBUTION

In this paper proposed the concept of an effective PDP model for integrity verification of data on distributed multi-cloud storage by using web-servers. This system will support the multiple private clouds by using homomorphic verifiable response technique and hash index hierarchy concept. System proposes an Advanced CPDP model to satisfy the data-owner by challenge-response method induced on the added new actor in system which is trusted third party. Trusted third party gives more effectiveness by confidentiality and the verification process of the outsourced data. Also it removes the overhead and the processing done by the Data Owner. Therefore, this concept can be considered as some enhancement in the related work till yet.

In future, this Advanced CPDP model can be applied to hybrid clouds i.e. combinations of private and public clouds.

5. ACKNOWLEDGMENTS

For proposing this model referred the IEEE Transaction paper under the title "Cooperative Provable Data Possession for Integrity Verification in Multi-cloud Storage" published in IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 12, DECEMBER 2012. This paper contains the results as per given in simulator, are to be implement in real time scenario using Web servers.

6. REFERENCES

[1] Yan Zhu, Hongxin Hu, Gail-Joon Ahn, Mengyang Yu, "Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage", *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 12, DECEMBER 2012.*

[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," *Proc. 14th ACM Conf. Computer and Comm. Security (CCS'07), pp. 598-609, 2007.*

[3] A. Juels and B.S.K. Jr., "Pors: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, 2007.

[4] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), pp. 90-107, 2008.

[5] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," *Proc. Fourth Int'l Conf. Security and Privacy in Comm. Netowrks (SecureComm '08), pp. 1-10, 2008.*

[6] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession", *In CCS '09, pp. 213-222, April 24,2012.*

[7] Feifei Liu, Davu Gu, Haining Lu,"An Improved Dynamic Provable Data Possession", *Proceedings of IEEE CCIS2011, pp 290-295, 2011.*

[8] Zhifeng Xiao and Yang Xiao, "Security and Privacy in Cloud Computing", *The University of Alabama, Tuscaloosa,* 24 March 2012.

[9] Venkatesa Kumar V, Poornima G, "Ensuring Data Integrity in Cloud Computing", *Journal of Computer Applications ISSN: 0974 – 1925, Volume-5, Issue EICA2012-4,* February 10, 2012.

[10] 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," *Proc. ACM Symp. Applied Computing, pp. 1550-1557, 2011.*

[11] Yashaswi Singh, Farah Kandah, Weiyi Zhang, "A secured cost effective multi-cloud storage in cloud computing", IEEE INFOCOM 2011 Workshop on Cloud Computing.

[12] 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, Sept. 2009.*

[13] Y. Zhu, H. Hu, G.-J. Ahn, Y. Han, and S. Chen, "Collaborative Integrity Verification in Hybrid Clouds," *Proc. IEEE Conf. Seventh Int'l Conf. Collaborative Computing: Networking, Applications.*

[14] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," *Proc. 14th European Conf. Research in Computer Security (ESORICS '09), pp. 355-370, 2009.*