New Approach for Outsourced Storages in Cloud: DAS

[1]Sonali D. Pardeshi , [2]Prof. S.M.Rokade
[1]Student ME Computer Engineering, [2]H.O.D. Dept. of Computer Engineering
[1]Savitribai Phule Pune University, [2]Savitribai Phule Pune University
[1]SVIT, Chincholi, [2] S.V.I.T. Chincholi
Nashik, India

***Abstract* - Cloud computing is a long dreamed vision of computing as utility, where data owner can remotely store their data in cloud to enjoy on demand high quality application and services form shared pool of computing resources. Data integrity protection in cloud computing is a mandatory task as users no longer have physical possession of the outsourced data users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Dynamic audit service is used for verifying the integrity of an untrusted and outsourced storage. This audit service is based on random sampling, index- hash table and fragment structure. It also performs anomaly detection at the regular intervals. Performance of audit services is enhancing by probabilistic query and periodic verification method. Hence our proposed audit system verifies the integrity with lower computation overhead also audit metadata does not require extra storage.**

***Keywords* - dynamic Audit, index hash table, random sampling, Storage Security**

I. INTRODUCTION

New computing paradigms keep emerging. One of the example is the cloud computing paradigm. It is a new computing model which uses advances in networking technology, where a client can take advantage of service provider's computing, storage or networking infrastructure. With the exponential growth rate of information, there is an increasing demand for outsourcing data storage to cloud services. Flexibility in on demand remote storage in cloud has numerous benefits Such as relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc. Cloud computing has very great impact on the businesses using information technology.

Along with the advantages of cloud, it brings, challenging threats towards users outsources data. basically cloud service providers (CSP) are separate administrative entities, thus it increases the risk of data security . Even though cloud have powerful infrastructures compare to personal computing devices they still face the broad range of both internal and external threats for data integrity.

Behavior of CSP toward the outsourced data can not be trusted. There are chances that CSPs might claim storage for monetary reasons by discarding data that have not been accessed, also they can even hide data loss incidents to maintain a reputation. Though cloud is fascinating for long-term large-scale storage, it does not give assurance regarding data integrity and data availability. This problem, if not properly addressed, may violate the success of cloud architecture. As users no longer physically possess the storage of their data, traditional cryptographic method for the data security protection cannot be directly adopted. Also simply downloading data for its integrity verification increases I/O and transmission cost across the network. It does not guarantee successful detection of corrupted data as it does not have any assurance for un accessed data.

Also users are reluctant for going through process of complexity in verifying data integrity. Consider example of enterprise where their may be more than one user accessing same cloud storage. So it is desirable that cloud only entertains verification request from a single designated party. Therefore dynamic audit service for integrity verification of untrusted and outsourced storages is introduced to addressed performance objectives such as public auditability dynamic operation , timely detection, effective forensic and lightweight processing.

Dynamic auditing service support auditing without downloading raw data. Using effective technique such as fragment structure, random sampling and index hash table, auditing can be perform along with timely anomaly detection. Probabilistic query and periodic verification are core methods for improving the performance of auditing. Feasibility and viability is evaluated using proof-of-concept prototype. the proposed audit system verifies the integrity with lower computation overhead also audit metadata does not require extra storage.

II. RELATED WORK

The hash function and signature schemes are used for checking cryptographic technique for the integrity checking of the data. They require the original copy of the data which is not solution as for doing the integrity checking or for auditing they download all the data and do the auditing of that data. In this case the TPA may behave unfaithful as for the monetary reason or for the personal repudiation reason, and also it not feasible that to download all the data as it increases the communication cost. Especially for the big size data.

To give assurance to the data owner it is very important that the TPA should behave faithful. Auditing is important task specially that should be done on behalf of the data owner. so the DO's completely rely on the TPA for the integrity checking and security of data.
To implement public auditability, the notions of proof of retrievability (POR) and PDP have been proposed by some researchers. These approaches were based on a probabilistic proof technique for a storage provider to prove that clients that data remain untamperd. Some POR/PDP schemes work on a publicly verifiable way, so that anyone can use the verification protocol to prove the availability of the stored data. To check the large amount of data over publically accessible interface POR/ PDP schemes are effective to use.
This Xie et al proposed very efficient method for auditing service using content comparability technique. But this technique is in efficient for dynamic auditing of outsourced storages in clouds regular data.
On the similar lines Wang et al developed new pubic auditing system using privacy preserving property. This architecture could not strongly support practical implementation as it is difficult to maintain extra storage.
To address such a problem, we introduce a fragment technique to improve the system performance and reduce the extra storage Another major concern is the security issue of dynamic data operations for public audit services. In clouds, one of the core design principles is to provide dynamic scalability for various applications. This means that clients can remotely stored data and dynamically update them through block operations such as modification, deletion and insertion. However, these operations may raise security issues in most of existing schemes, e.g., the forgery of the verification metadata (called as tags) generated by DOs and the leakage of the users secret key. Hence, it is crucial to develop a more efficient and secure mechanism for dynamic audit services, in which a potential adversary's advantage through dynamic data operations should be prohibited**.**

III. THE PROPOSED SCHEMES

Auditing system architecture for outsourced data is as shown in fig 1. In this architecture four entities are involved
1.Data owner :- who has large amount of data
2. CSP:- who provides data storage service and has sufficient storage space and computation resource
3 TPA:- who manages & monitor outsourced data.
4 AA :- who have right to access and manipulate stored data.

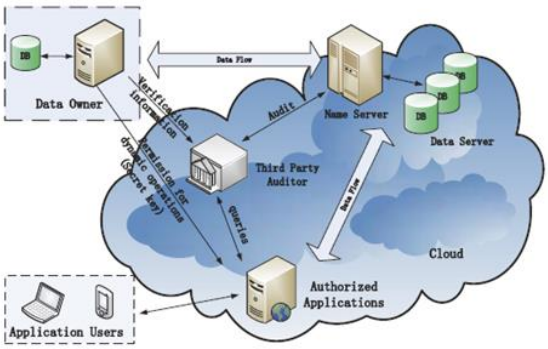

fig 1: architecture of proposed system

A reliable TPA has to perform following auditing function
1: regular checks on integrity and availability of data.
2: organize manage and maintain outsourced data.
3: support dynamic data operations.
4:it should be able to take evidences in case of disputes about inconsistency of data.
Hence to relies this function auditing service consist of three process:

1. Fragment structure and secure tag: Auditing system uses fragment structure to maximize storage efficiency. In the process of fragment structure outsourced file F is split into n blocks. Such as {m1,m2,m3,…….mn} and each of this block is again split into s sectors such as $\{m_{i1},m_{i2},\ldots..m_{is}\}$
The frame work consist of block- tag paring {mi,бi}
Here бi is signature tag of block n. this signature is generated by some secrets Ʈ = Ʈ1,Ʈ2 …..Ʈs. when TPA gives the challenge for verification, we can use tags and corresponding data to create a response.
This response is verified without raw data.
CSP stores this block tag pair and encrypted secrets are stored in TTP.
As the file is split into nxs sectors and each block corresponds to a tag. Hence, if we increase sectors the storage signature tag can be reduce. Which enhances audit performance.

2 Periodic Sampling: Random sampling checks reduces workload of audit service in comparison with whole checking. In fact of performing sample check, auditing achieves effective results. Given a randomly chosen challenge Q={(i,v_i)}where I =subset of block index $v_{i=}$ random coefficient, this algorithm is used to produce constant size Response=(µ1, µ2, µ3…. µs,б') where µi come from $\{m_{k,i},V_k\}$.and б' come from $\{б_k v_k\}$.Since single sampling check algorithm may overlook small number of data

abnormalities, we prefer periodic sampling approach. Using this approach audit activity are efficiently schedule in audit period.
3 index hash table: index hash table are used to support dynamic data operations. This hash table record changes of file blocks and generate hash value for each block index hash table have similar structure as of file block allocation in file systems. Index hash table consist of serial number , block number version number and random integer. All records in index hash table differ from each other. Index hash table provides higher assurance to monitor behavior of CSP as well as valuable evidence for computer forensic.

IV. MATHEMATICAL MODEL

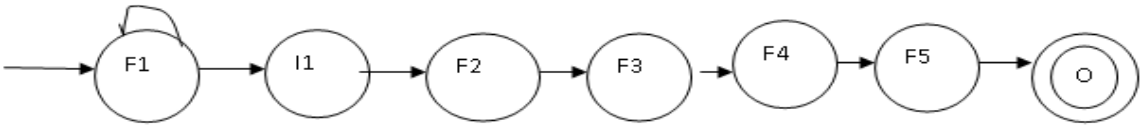

fig 2 : mathematical model

The Figure 2 shows the mathematical model of propose system:
The Mathematical Model of proposed system shown in Figure 2 In which f1 is function for Authentication in which user login if user login successfully then user inserts the data into various fields where I1 is the Patient Data therwise he remains in same state that is f1. After that f2 is function for Middleware that is Upload the data onto cloud then f3 is function for generating signature of data. Data that is to be sent onto cloud is I encrypted format where f4 is function for TPA to authenticate data on request or to authenticate data Dynamically . F5 is function to check integrity of the data and finally O is the Output function which will return the result whether The integrity of the data is change or not.
The parameter are explain bellow

Input(I) Parameter
I= [I1]
Where I is a set of Input.
I1= Patient Data.

Function(F) Parameter
F=[f1,f2,f3,f4,f5]
Where F is a function for processing.
f1= Authentication.
f2=Middleware to upload the data on cloud.
f3= To generate signature.
f4=integrity check .
f5=generate report.

Output(O)Parameter
O=[O1]
Where O is the Output.
O1=Report generated by TPA

V. IMPLEMENTATION

Algorithm for dynamic auditing service is as follows:
KeyGen (1k) takes a security parameter k as an input, and returns a public key pair (pk, sk)
TagGen (sk; F) takes a secret key sk and a file F,and returns a triple (T,Ψ, б),
where T denotes the secret used to generate verification tags,Ψ is a set of PVPs and IHT X i.e., = (u, X), and б denotes a set of tags.
A DO or AAs save the secret key sk that is, the secrets of process file Ƭ are discarded after the tags are generated.
Suppose a DO wants to store a file in a storage server, and maintains a corresponding authenticated index structure at a TPA, user uses KeyGen( ), to create public or secrete key pair (pk, sk) . after key generation public key pk to TPA. after that owner chooses random secret Ƭ blic verification information Ψ = (u,ɤ) and signature tag б . finally owner sends Ψ & (F,б) to TPA and CSP.

Supporting Periodic Sampling Audit
TPA perform integrity check as follows: initially TPA queries database to obtain Ψ and also initializes interactive protocol proof(CSP, Clients). After that it performs commitment challenge and respond and verifies interactive data to get the result.
The proof process is defined as proof (CSP,TPA) as an interactive protocol between CSP and TPA, <CSP(F,б),TPA> (Pk,Ψ).at the end of protocol TPA returns either 0 or 1 value.
Where 1 means file is correctly stored on the server.

Supporting Dynamic Data Operations

Dynamic data operations are define as follows:

Update(sk; Xi; mi) i) is an algorithm run by AA to update the block of a file mi at the index i by using sk, and it returns a new verification metadata.

Delete(sk; Xi;mi) is an algorithm run by AA to delete the block mi of a file mi at the index i by using sk, and it returns a new verification metadata (Xi, i)

Insert(sk; Xi;mi) is an algorithm run by AA to insert the block of a file mi at the index i by using sk, and it returns a new verification metadata (Xi, i).

DO or AA can only perform the dynamic data operation as they have secrete key sk, which ensure the security. All operations are perform on data blocks. Index hash table is updated to implement the audit service. Basically an AA obtains public verification information Ψ from TPA. After receiving the information it invoke dynamic data operation update delete and Insert functions. Thus it receives new values б' and Ψ and forward it to CSP and TPA respectively. Finally CSP verifies validity of updated data and TPA modifies audit records after confirmation message from CSP.

VI. RESULT ANALYSIS

The proposed system is compared and analyze with previous system in terms of communication and computation overhead.

1. Communication overhead for uploading data block.
2. Computational overhead for tag generation and verification tag evaluation.

Following are notations used.

u_k = user who tries to update
d =no. of user in group
L=no. of challenge data block
S=no. of sectors.
id= identifier
q_{id} = request for identifier
u_{info} = update information
f_{prp} =peudo random permutation
H_G= {0,1}* to G
Pair= Pairing opration
EXP_G= exponentiation in G
MUL_G=multiplication in G
MUL_{ZP}=multiplication in Z_P
Add_{zp}=addition in Z_P

the way to get an identifier of updated data block depends on which entity manages an index table. In Wang et al.'s work, *uk* needs one round-trip communication to request and receive the identifier. Zhu et al. utilize a way that the TPA manages the index table. It needs an additional connection between *uk* and the TPA for the identifier and a report of update information. Yang and Jia utilize a way that a user manages the index table by himself. Although it is suitable when the outsourced data is managed by a single user, it requires more communication costs for synchronization of the index tables when the data is shared by multiple users.

Communication costs are summarized in Table 1. In this costs for uploading a data block and a corresponding tag is omitted for simplicity. Although the proposed scheme seems to
require the same cost as Zhu et al.'s approach, there may be update delays caused by concentration of communications to the TPA. On the other hand, the proposed scheme removes this delay via a direct
acquisition of the identifier from the CSP. To synchronize the index tables of users and the TPA, *uk* needs to broadcast extra update information. Considering this circumstance, additional communications caused by broadcast might be added into Table 1.

Table 1: communication costs for updating a data block.

	Wang et. Al	Zhu et.al	Yang and jia	Our scheme
CONNECTION	uk↔CSP	uk↔TPA	Uk with other user	uk↔TPA
		uk↔CSP	uk↔TPA	uk↔CSP
			uk↔CSP	
Extra communication cost	$\|q_{id}\|+\|id\|$	$\|q_{id}\|+\|id\|+u_{info}$	d x u_{info}	$\|q_{id}\|+\|id\|+u_{info}$

Computational Overhead. Computation costs for tag generation and verification are described in Table 2. The proposed scheme requires a single Exp$G$ operation in tag generation, while the others require Exp$G$ and Mul$G$ operations which are proportional to the number of sectorsin a data block. When the TPA verifies a proof received from the CSP, one Exp$G$ and one Mul$G$ operations

are required in the proposed scheme regardless of the number of challenged data blocks. However, the others require Exp$G$ and Mul$G$ operations linear to the number of challenged data blocks, which cause a significant overhead to the TPA.

Table2: computation costs for a tag generation and verification

	Tag generation	Verification
Wang et.al	(s+1)·Exp_G+S.MUL_G+H G	$2 \cdot \text{Pair} + (L+S) \cdot \text{Exp}G + (L+S-1) \cdot \text{Mul}G + L \cdot HG$
Zhu et.al	(s+2)·Exp_G+S.MUL_G+H G	$3 \cdot \text{Pair} + (L+S) \cdot \text{Exp}G + (L+S) \cdot \text{Mul}G + L \cdot HG$
Yang et.al	(s+1)·Exp_G+S.MUL_G+H G	$2 \cdot \text{Pair} + (L+1) \cdot \text{Exp}G + L \cdot \text{Mul}G + L \cdot \text{Mul}Zp + L \cdot HG$
		$2 \cdot \text{Pair} + \text{Exp}G + \text{Mul}G$
Our scheme	Exp_G+S.MUL_{ZP}+ADD_Z $_PF_{prp}$	$L \cdot \text{Mul}Zp + (L-1) \cdot \text{Add}Zp + L \cdot F\text{prp}$

VII. CONCLUSION

Hence it is concluded that this system is effective for integrity check of the cloud storage . it also reduces computation and communication cost. Zero knowledge property is also preserved . performance of TPA and CSP are enhanced periodic sampling audit.

VIII. ACKNOWLDGEMENT

I take this opportunity to express my hearty thanks to all those who helped me in the completion of the Paper. I express my deep sense of gratitude to my guide Prof. S.M.Rokade, HOD and Asst.Prof., Computer Engineering Department, Sir Visvesvaraya Institute of Technology, Chincholi for his guidance and continuous motivation. I great fully acknowledge the help provided by him on many occasions, for improvement of this paper with great interest. I would also like to thank to Prof. M.M.Naoghare P.G. Co-ordinator for her great support and excellent guidance.

IX. REFERENCES

[1] C. Wang, Q. Wang, K. Ren, and W. Lou, Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing, Proc. IEEE INFOCOM, pp. 1-9, 2010

[2] M. Xie, H. Wang, J. Yin, and X. Meng, Integrity Auditing of Outsourced Data, Proc. 33rd Intl Conf. Very Large Databases (VLDB), pp. 782-793, 2007.

[3] Y. Zhu, H.Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S. Yau, E_cient Provable Data Possession for Hybrid Clouds, Proc. 17th ACM Conf. Computer and Comm. Security, pp. 756-758, 2010.

[4] A.R. Yumerefendi and J.S. Chase, Strong Accountability for Network Stor- age, Proc. Sixth USENIX Conf. File and Storage Technologies (FAST), pp. 77-92, 2007.

[5] Amazon Web Services, Amazon S3 Availability Event: July 20, 2008, http://status.aws.amazon.com/s3-20080720.html, July 2008.

[6] M. Mowbray, The Fog over the Grimpen Mire: Cloud Computing and the Law, Technical Report HPL-2009-99, HP Lab., 2009.

[7] A.A. Yavuz and P. Ning, BAF: An E_cient Publicly Veri_able Secure Audit Logging Scheme for Distributed Systems, Proc. Ann. Computer Security Applications Conf. (ACSAC), pp. 219-228, 2009.

[8] 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, pp. 598-609, 2007.

[9] Gartner, Gartner Says Cloud Adoption in Europe Will Trail U.S. by at Least Two Years, http://www.gartner.com/it/page. jsp?id=2032215, May 2012.

[10] Ming Li, Shucheng Yu, Yao Zheng, Kui Ren, and Wenjing Lou, "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 24, NO. 1, JANUARY 2013.

[11] R. Rivest, A. Shamir, and L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,Comm. ACM, vol. 21, no. 2, pp. 120-126, 1978.

[12] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, Scalable and E client Provable Data Possession, Proc. Fourth Intl Conf. Security and Privacy in Comm. Netowrks (SecureComm), pp. 1-10, 2008.

[13] S. Bugiel, S. Nurnberger, A.-R. Sadeghi, and T. Schneider, Twin Clouds: Secure Cloud Computing with Low Latency,Proc. 12th IFIP TC 6/TC 11 Intl Conf. Comm. and Multimedia Security (CMS 11), pp. 32-44, 2011.