Achieving Data Integrity in Cloud Storage Using BLAKE Hash Function

Prof. Hitesh Patel, Prof. Parin Patel, Prof Kiran Patel
Assistant Professor
Gandhinagar Institute of Technology, Gandhinagar, India

***Abstract* - In Cloud computing data security is biggest problem. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage applications and services) .User can store their data remotely without maintaining local copy of data. So the integrity verification of the data is major problem in cloud storage. There is issue of security and privacy of data storage because cloud provider is not completely trusted. Integrity Verification can be managed by the use of TPA (Third Party Auditor) or without TPA. In this paper I apply integrity check without using TPA and use cryptographic hash function BLAKE to generate the signature of file or massage. It is more efficient then other hash functions like md5, SHA-1, SHA-2 and SHA-3. I use more efficient Cryptographic encryption algorithm RSA, AES and DES is more secured and faster. So this proposed model achieves storage correctness, Data Confidentiality, Authentication, Integrity and Efficient Data Access (Sharing of file) in cloud's dynamic nature for maintaining low computation and communication cost.**

***Keywords*— Cloud Computing, Security, Integrity Verification, Data Storage Correctness, Privacy**

I. INTRODUCTION

Cloud computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy high quality applications and services from a shared pool of configurable computing resources. cloud computing has various security issues like data theft, data integrity on cloud server, secure transmission of data, integrity verification without much overhead/computation cost, access rights management and security while sharing file to other user. In cloud computing user can store data remotely on cloud server. Cloud server (Provider) is external entity is not completely trusted .Data can be altered/temper by unauthorized entity without permission of data owner on cloud server. How the data owner make sure that his data has not been modified by others (or may be by the Cloud provider itself, accidently or intentionally). So data storage correctness is required for detecting such kind of unlawful activities on data is an utmost priority issue.

Data storage correctness scheme classified in two categories

1. without use of third party auditor (Non TPA)
2. With use of third party auditor (TPA).

In case of using TPA, an external Third Party Auditor (TPA) that verifies the data integrity and sends report to user, some time in form of extra hardware or cryptographic coprocessor is required. This hardware scheme provides better performance due to dedicated hardware for the auditing process but has some drawbacks

1. Such as single TTP resulting into bottleneck in the system, TPA is supposed to be a central, independent & reliable component; it may become bottleneck to the entire system. Any unusual activity in TPA may cause entire cloud system to go down or reduction in the performance.
2. As the data sent from cloud data owner premise is in encrypted form and the required credentials to decrypt the same are kept hidden from cloud service provider, during regulatory compliance, laws which make the data owner responsible for protection of his data can be followed
3. Some time with the use of TPA extra hardware or cryptographic coprocessor is needed.
4. During any legal investigation, cloud service provider cannot handover the data to any statutory body without consulting to data owner.

To provide data security in cloud computing we use cryptographic techniques: Cryptography is the science of using mathematics to encrypt and decrypt information. Once the information has been encrypted, it can be stored on insecure media or transmitted on an insecure network (like the Internet) so that it cannot be read by anyone except the intended recipient. We use Symmetric key (AES, DES, 3DES) and Asymmetric key (RSA, Diffie-Hellman) algorithm for encryption and decryption of data.

In data storage correction data integrity verification can performed with use of hash function such as MD5, SHA1, SHA2, SHA3, BLAKE using this hash function we create unique signature of data for later verification of data integrity.

II. RELATED WORK

This section illustrates recent research in Cloud data storage correctness. Recently, few researchers have proposed approaches based on third party auditor (TPA). Wang at el. [5] propose an approach which enables public auditability for Cloud data storage security through external TPA, without demanding local copy of data or imposing extra online burden on Cloud. Gowrigolla at el. [12] outline a data protection scheme with public auditing which allows data to be stored in encrypted form on Cloud server without loss of accessibility or functionality for authorized users. Wang at el. [5] propose Homomorphic authenticator with

random masking token for verifying data integrity using TPA in cloud. Authors of [8] recommend a design of cryptographic cloud data storage and suggest various cryptographic approaches to achieve access control, authentication, Integrity, Availability, Reliability, Data sharing and confidentiality.

Researchers of [8] [21] address problem of access control mechanism using cryptographic techniques which degrades the performance and increase computation cost for management of key at user as well Cloud server side. They give solution by capability based access control scheme which gives surety that only valid user having rights to access data available on Cloud. They also propose and modified version of Diffie-Hellman key exchange scheme for sharing symmetric key securely. Motivated by these papers, we are proposing operational algorithms for achieving the above mentioned goals.

III. EXISTING SYSTEM

In existing system they use Third party auditor to check the integrity of data in this Scheme having three components:

1. Cloud User (CU)
2. Cloud Service Provider (CSP) & Cloud Server (CS)
3. iii)Third party Auditor (TPA)

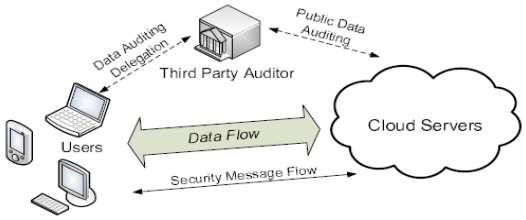


Fig1:Third Party Auditor Scheme[5]

Public Auditing Scheme using third party auditor (TPA)

They use the technique to uniquely integrate the homomorphic authenticator with random masking technique. In their system, the linear combination of sampled blocks in the server's response is masked with randomness generated by a pseudo random function (PRF).With random masking, the TPA no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the user's data content, no matter how many linear combinations of the same set of file blocks can be collected. Meanwhile, due to the algebraic property of the homomorphic authenticator, the correctness validation of the block-authenticator pairs will not be affected by the randomness generated from a PRF. A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, and VerifyProof). KeyGen is a key generation algorithm that is run by the user to setup the scheme.

SigGen is used by the user to generate verification metadata, which may consist of MAC, signatures, or other related information that will be used for auditing. GenProof is run by the cloud server to generate a proof of data storage correctness, while, VerifyProof is run by the TPA to audit the proof from the cloud Server.

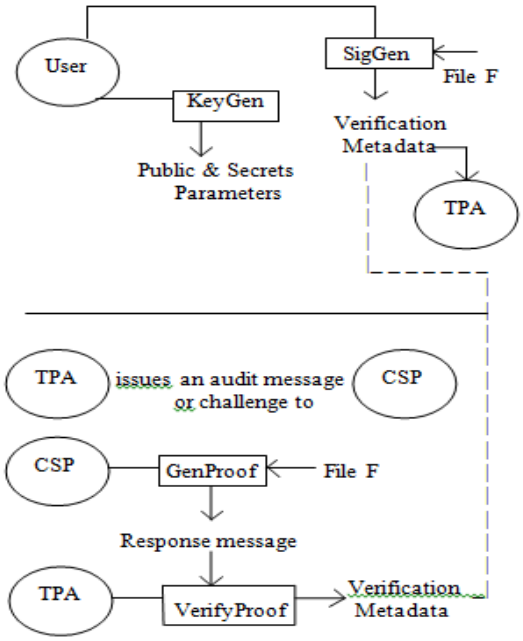


Fig2:Third Party auditing scheme

IV. PROBLEM IN EXISTING SYSTEM

1. TPA is supposed to be a central, independent & reliable component; it may become bottleneck to the entire system. Any unusual activity in TPA may cause entire cloud system to go down or reduction in the performance.

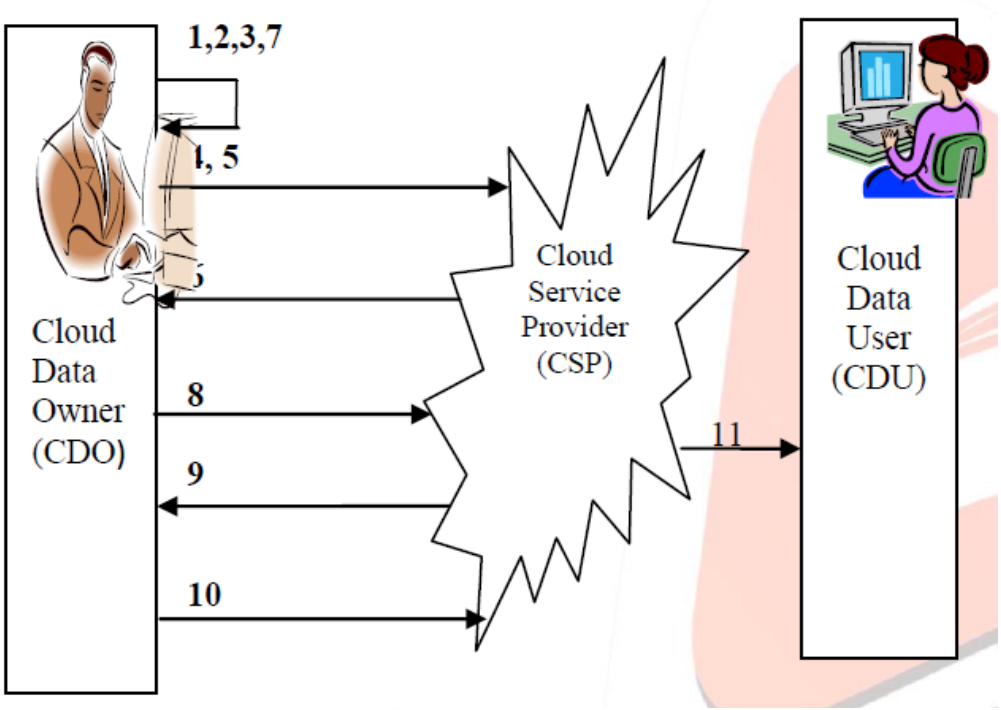
2. Cloud data owner can directly control the cryptographic operations to be performed on his data stored on cloud. Cloud data owner can specify privacy level of his data and also choose combinations of cryptographic operations from available options instead of TPA to decide what is good for his data.
3. Some time with the use of TPA extra hardware or cryptographic coprocessor is needed.
4. As the data sent from cloud data owner premise is in encrypted form and the required credentials to decrypt the same are kept hidden from cloud service provider, during regulatory compliance, laws which make the data owner responsible for protection of his data can be followed.
5. During any legal investigation, cloud service provider cannot handover the data to any statutory body without consulting to data owner.
6. No file sharing mechanism between CDO to CDU
7. High Computational and communication cost in

V. PROPOSED SYSTEM

I propose a data storage security model, which intends to solve the data security problem, Integrity verification and File sharing problem.

My Propose System Contains three stakeholders like:

1. Cloud data owner (CDO), who generates and owns the data. Possessing all rights about file operation, it can pass on the same to other Cloud data users.
2. Cloud service provider (CSP), which is the central core component of the whole system. It also acts as a data warehouse for CDO.
3. Cloud Data User (CDU), who uses the data based on credentials received from the cloud data owner (CDO).



Above scenario works as follow:

1. CDO generates key using Symmetric Key (DES, AES) and Asymmetric key generation (RSA) algorithm and store that key.
2. CDO encrypts file using Symmetric Key (DES, AES) and Asymmetric key (RSA) generation algorithm.
3. CDO creates Hash code (Signature) using cryptography hash functions Blake on Encrypted file and store that signature in database.
4. CDO upload encrypted file on cloud Service provider (CSP).
5. CDO request to CSP for file integrity verification.
6. CSP calculates hash code for the encrypted file which is uploaded by the CDO and sends it to CDO.
7. CDO compares the hash code received by CSP with the actual hash code to check the correctness of data which is stored on the CSP.
8. CDO requests for view/download the file.
9. CSP decrypt file using Symmetric key (DES, AES) and Asymmetric key (RSA) generation algorithm and send to CDO.
10. CDO Grant file Access Rights(Sharing of file) to CDU
11. CDU Access CDO File with Grant Access Rights.

Registration Phase

CDO and CDU register themselves to Cloud before they start accessing data by providing their unique identification (Customer_ID) and password, Email Id, Birth date and other Information. This information will be stored in Cloud Customer Registration Master Table (table 1) maintained by the Cloud, for future customer verification by CSP.

Field Name	Field Details
Customer_ID	Unique Identification of each Customer on the Cloud, (Primary key)
Customer_Name	Customer Name
Email_ID	Customer Email for communication
Mobile	Customer Mobile no for communication
Birth date	Customer birth date
Password	Password of the customer to access data on cloud

Pre-Storage Phase

Prior to storing (encrypted) data into the Cloud, CDO needs to decide cryptographic primitives such as select encryption algorithms (AES, DES, RSA) and signature, Data file etc. It will be stored on file Access Control Policy Master Table (table 2).

Field Name	Field Details
File_ID	Unique identification of every file on Cloud. (Primary key)
Owner_ID	Unique Customer Identification as mentioned in table 1. (Foreign key)
Created_Date_Time	Date and time (time stamp) of file creation.
File_size	Uploaded File size in KB
File_type	Uploaded File type text, image etc
Encryption_Algo_Type	Type of encryption algorithm AES, DES, and RSA (Contains 0 if data is not sensitive)
Hash_Code	Hash code based on encoding algorithm selected by CDO.
Owner_Signature	CDO's signature for later verification
Encryption and Decryption Key	Data Encryption and Decryption for Symmetric (AES,DES) and Asymmetric Algorithm(RSA)

Verification Phase

Any time, CDO/CDU can use this phase to check integrity of data, by issuing QUERY to CSP and CSP returns answer in form of code REPLY, which will be compared by CDO's locally stored code of the same file (or can be re-computed). The integrity of the data is considered to be protected if they are same.

Grant Rights Phase

CDO share file to CDU using CDU unique Custmer_id

Field Name	Field Details
File_ID	Unique identification of every file on Cloud. (Primary key)
Owner_ID	Unique Customer Identification as mentioned in table 1. (Foreign key)
Created_Date_Time	Date and time (time stamp) of file creation.
File_size	Uploaded File size in KB
File_type	Uploaded File type text, image etc
Encryption_Algo_Type	Type of encryption algorithm. (Contains 0 if data is not sensitive)
Hash_Code	Hash code based on encoding algorithm selected by CDO.
Owner_Signature	CDO's signature for later verification
Encryption and Decryption Key	Data Encryption and Decryption for Symmetric (AES) and Asymmetric Algorithm
Share to	CDU Custmer_ID for file Access Rights

Table: 3 Verification Phase Table

Encryption Process: Performed at CDO's site or CDU's site, they can choose encryption algorithm along with appropriate key or they can use their custom-designed algorithms, too. Two broadly known options for encryption viz. symmetric key encryption (e.g. AES) and asymmetric key encryption (e.g. RSA) may be used here. The keys are to be stored and maintained by the data owner, per file, locally. (Alternatively, we can use a trusted third party, which takes care of storage and maintenance of these keys.)

Algoritham : Encryption	
1	Select File for Encryption
2	Select Algorithms for Encryption (AES,RSA,DES,3DES)
2	Generate Key (For RSA Asymmetric key Generate key pair Private Key and Public Key)
3	Enc_file←Enc(sk(File)
4	CHash←Hash(BLAKE(Enc_file)
5	CSP Store CHash , Key, Enc_file update D/B

Verifying Data Integrity: Simply downloading the data for integrity verification is not a practical solution due to expensiveness in I/O cost and unsafe files transfer across the network and may lead to new vulnerabilities [16]. Moreover, legal regulations, such as (HIPAA) [17], further demand the outsourced data not to be leaked to external parties (e.g. TPA). So applying encryption before outsourcing is the most preferred way to mitigate the privacy concern.

Algorithm : Verifying Data Integrity	
1	Select File for verify Data Integrity
2	Send {File_name} to CSP
2	CSP Calculate CSP_Hash, and Send to CDO
3	CDO Compare {CDO_Hash ,CSP_Hash}
4	If Match Hash than Integrity Verification Successes
5	Else Data may be Corrupted or Tampered Integrity Verification Failed

Decryption process:

When CDO wants that Encrypted file from cloud server they perform the Decryption process so file is decrypted using decryption algorithms and key after CDO get that original file.

Algorithm : Decryption	
1	Select File for Decryption
2	Retrieve Decryption algorithm and Key to CDO
2	Decryption_ file ←_Enc_file(key, Enc_Algo)
3	Decryption_ file Downloaded to CDO or CSP.

VI. EXPERIMENTAL RESULT

I have implement system using PHP on cloud using window server 2012 using HYPER-V.

Fig3:File upload on Cloud using RSA encryption algoritham

Fig4: Check Integrity of uploaded file on cloud server

TABLE II COMPARISON OF AES, DES AND RSA IN TIME AND SECURITY

Data		Algo Type	Time(s)
File (80KB)	1	AES	2.4
		DES	1.9

	RSA	9.24
File 2 (140KB)	AES	3.2
	DES	2.5
	RSA	11.4

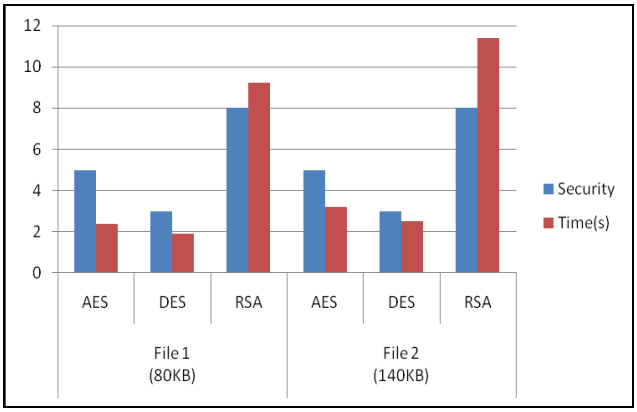


Fig. 5 Comparisons of AES, DES and RSA in Time and Security

TABLE II COMPUTATIONAL COST

Parameters	Existing system[6]		proposed System	
Sampled blocks c(bytes)	460	300	460	300
Server comp. time (ms)	335.17	219.27	340.56	215.1
TPA comp. time (ms)	530.6	357.53	0	0
Non TPA comp. time (ms)	0	0	350.63	156.38
total computational cost	865.77	576.8	691.19	371.48

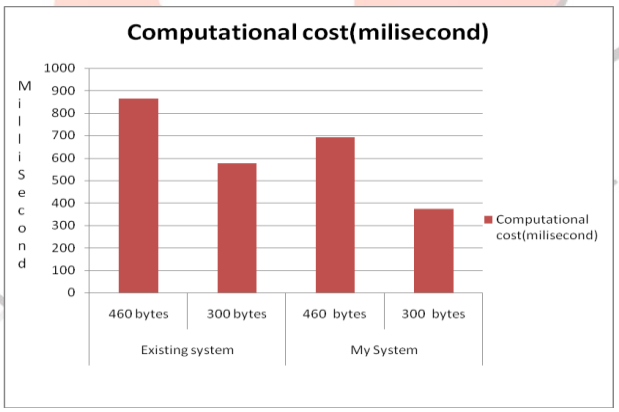


Fig. 6 Computational Cost

VII. CONCLUSIONS

I analyzed Data storage Correctness issues in cloud computing and provides lightweight integrity verification scheme using non third party approach (non TPA). In this Scheme provides encrypt and decrypt data using Symmetric (AES, DES) and Asymmetric (RSA) Algorithms and use BLAKE hash function for generating hash code. My proposed system is provides high security, lightweight data integrity verification, data hiding and secure access right to other cloud data file requester.

In future we can provide additional File Sharing facility like Google Drive, Sky Drive etc and provide upload facility with large data file.

REFERENCES

[1] Cloud Security Alliance, "Security Guidance for critical areas of focus in Cloud Computing V3.0" https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf

[2] National Institute of Standards and Technology- Computer Security Resource Center www.csrc.nist.gov

[3] http://en.wikipedia.org/wiki/Cloud_computing

[4] Hiren B. Patel, Dhiren R. Patel, BhaveshBorsania, Avi Patel, "Data Storage Security Mode for Cloud Computing", in Third International Conference on Advances in Communication, Network, and Computing – CNC 2012 organized by ACEEE. February, 2012.
[5] Cong Wang, Student Member, IEEE, Sherman S.-M. Chow, Qian Wang, Student Member, IEEE,KuiRen, Member, IEEE, and Wenjing Lou, Member, IEEE "Privacy-Preserving Public Auditing for Secure Cloud Storage, IEEE-2012
[6] Cong Wang1, Qian Wang1, Kui Ren1, and Wenjing Lou2," Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing", IEEE INFOCOM 2010, San Diego, CA, March 2010
[7] Ms. Vaishnavi Moorthy1, Dr. S. Sivasubramaniam2," Implementing Remote Data Integrity Checking Protocol for Secured Storage Services with Data Dynamics and Public Verifiability In Cloud Computing, IOSR Journal of EngineeringMar. 2012, Vol. 2(3) pp: 496-500
[8] C. Hota, S. Sanka, M. Rajarajan, S. Nair, "Capability-based Cryptographic Data Access Control in Cloud Computing", in International Journal of Advanced Networking and Applications, Volume 01, Issue 01, 2011.
[9] Rosario Gennaro and Daniel Wichs, Fully Homomorphic Message Authenticators IBM Research, T.J. Watson, May 23, 2012
[10] K. Kajendran, J. Jeyaseelan, J. Joshi, "An Approach for secures Data storage using Cloud Computing" In International Journal of Computer Trends and Technology- May to June Issue 2011
[11] W. Luo, G. Bai, "Ensuring the Data Integrity In Cloud computing" In Proceedings of IEEE CCIS, 2011.
[12] S. Sanka, C. Hota, and M. Rajarajan, "Secure data access in cloud computing," in 2010 IEEE 4th International
[13] http://en.wikipedia.org/wiki
[14] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. of Asiacrypt 2008, vol. 5350, Dec 2008, pp. 90–107.
[15] 104th United States Congress, "Health Insurance Portability and Accountability Act of 1996 (HIPPA)," Online at http://asp.ehhs.gov/admnsimp/pl104191.htm, 1996.
[16] Amazon.com, "Amazon s3 availability event: July 20, 2008,"Online at http://status.aws.amazon.com/s3-20080720.html,2008.
[17] K. Raen, C. Wang, Q. Wang, "Security Challenges for the Public Cloud", Published by IEEE Computer Society, Jan/Feb 2012
[18] J.-P. Aumasson, L. Henzen, W. Meier, and R. Phan, "SHA-3 proposal BLAKE," December 2010.
[19] GALS System Design: Side Channel Attack Secure Cryptographic Accelerators
[20] AES encryption and decryption http://www.iis.ee.ethz.ch/~kgf/acacia/c3.html
[21] Kamara, S., Lauter, K.: "Cryptographic cloud storage". In: Proceedings of the 14th international conference on Financial cryptography and data security, FC'10, pp. 136-149. Springer-Verlag, Berlin, Heidelberg (2010)