Data Integrity Verification in Cloud Storage without using Trusted Third Party Auditor

Rana M Pir
Lecturer
Leading university, sylhet Bangladesh
Ranapir@yahoo.com

***Abstract*— Data security is biggest problem. In cloud storage 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. We ensure the data not tampered with other 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 we review integrity verification with TPA and non TPA with security.**

***Index Terms*— Cloud Computing, Security, Integrity Verification, Data Storage Correctness, Privacy, Third party auditor**

I. INTRODUCTION

Cloud computing is a utility, where users can remotely store their data into the cloud storage so as to enjoy high quality applications and services. Cloud computing involves delivering hosted services over the internet. These services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). A cloud service has three distinct characteristics that differentiate it from traditional hosting. The advantage of cloud is cost savings. The prime disadvantage is security

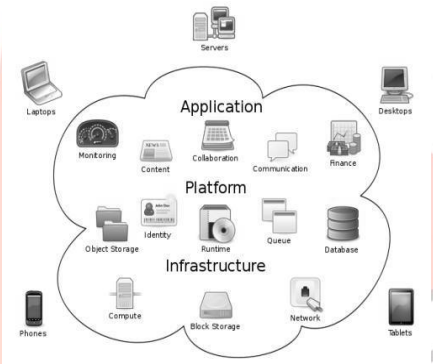


Fig: 1.1 Cloud Computing [7]

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 (a) without use of third party auditor (Non TPA) (b) 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.

Data Security & Integrity Verifying with Hash Function

1. **Using Cryptographic Algorithm**
   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, Deffi-Hellman) algorithm for encryption and decryption of data.
2. **Using Hash Function (Message Digest)**
   A cryptographic hash operation produces a fixed-length output string (often called a digest) from a variable-length input string. For all practical purposes, the following statements are true of a good hash function:
   Collision resistant: If any portion of the data is modified, a different hash will be generated.
   One-way: The function is irreversible. That is, given a digest, it is not possible to find the data that produces it.
   These properties make hash operations useful for authentication purposes. For example, you can keep a copy of a digest for the purpose of comparing it with a newly generated digest at a later date. If the digests are identical, the data has not been altered.
3. **Using MAC (Message Authentication Code)**
   A MAC operation uses a secret key and cipher algorithm to produce a value (the MAC) which later can be used to ensure the data has not been modified. Typically, a MAC is appended to the end of a transmitted message. The receiver of the message uses the same MAC key, and algorithm as the sender to reproduce the MAC. If the receiver's MAC matches the MAC sent with the message, the data has not been altered.
4. **Using HMAC (Hash MAC)**
   An HMAC operation uses a cryptographic hash function and a secret shared key to produce an authentication value. It is used in the same way a MAC is used.

Hash functions, also called message digests and one-way encryption, and are algorithms that, in some sense, use no key. Instead, a fixed-length hash value is computed based upon the plaintext that makes it impossible for either the contents or length of the plaintext to be recovered. Hash algorithms are typically used to provide a digital fingerprint of a file's contents often used to ensure that the file has not been altered by an intruder or virus. Hash functions are also commonly employed by many operating systems to encrypt passwords. Hash functions, then, provide a measure of the integrity of a file. Hash algorithms that are in common use today include: Message Digest (MD) algorithms: A series of byte-oriented algorithms that produce a 128-bit hash value from an arbitrary-length message. We mainly used MD2,MD4 and MD5 Message Digest .

Secure Hash Algorithm (SHA): Algorithm for NIST's Secure 180- 1 Hash Standard (SHS). SHA-1 produces a 160-bit hash value and was originally published as FIPS and (aka SHA-2) describes five algorithms in the SHS: SHA-1 plus SHA-224, SHA-256, SHA-384, and SHA-512 which can produce hash values that are 224, 256, 384, or 512 bits in length, respectively.

II. 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)

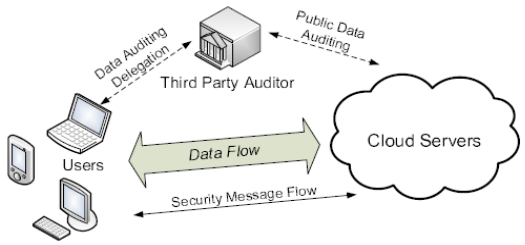


Fig1:Third Party Auditor Scheme[5]

Public Auditing Scheme using third party auditor (TPA)

Basic Scheme I The cloud user pre-computes MACs σi= MACsk(i||mi) of each block mi (i∈ {1, . . . , n}), sends both the data file F and the MACs {σi}1≤i≤n onto the cloud server, and releases the secret key sk to TPA. During the Audit phase, the TPA requests from the cloud server a number of randomly selected blocks and their corresponding MACs to verify the correctness of the data file. The insight behind this approach is that auditing most of the file is much easier than the whole of it. However, this simple solution suffers from the following severe drawbacks:

1. The audit from TPA demands retrieval of users' data, which should be prohibitive because it violates the privacy-preserving guarantee;

2. Its communication and computation complexity are both linear with respect to the sampled data size, which may result in large communication overhead and time delay, especially when the bandwidth available between the TPA and the cloud server is limited.

Basic Scheme II To avoid retrieving data from the cloud server, one may improve the above solution as follows: Before data outsourcing, the cloud user chooses s random message authentication code keys $\{skt\}1 \leq t \leq s$, pre-compute MACs for the whole data file F, and publishes these verification metadata to TPA. The TPA can each time reveal a secret key skt to the cloud server and ask for a fresh keyed MAC for comparison, thus achieving privacy-preserving auditing. However, in this method: 1) the number of times a particular data file can be audited is limited by the number of secret keys that must be a fixed priori. Once all possible secret keys are exhausted, cloud user then has to retrieve data from the server in order to re-compute and republish new MACs to TPA. 2) The TPA has to maintain and update state between audits, i.e., keep a track on the possessed MAC keys. Considering the potentially large number of audit delegations from multiple users, maintaining such states for TPA can be difficult and error prone.

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.

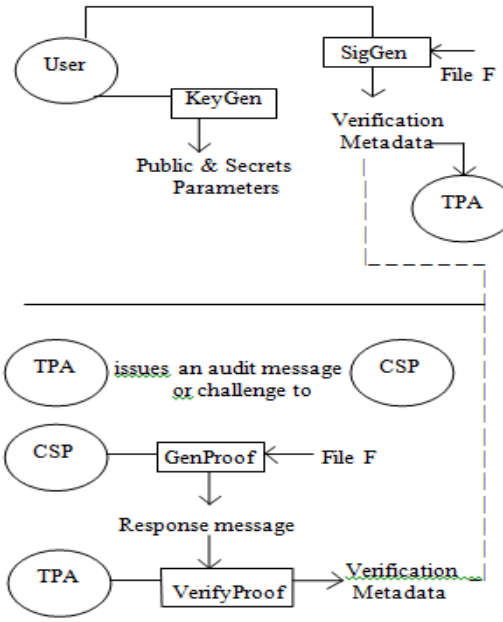


Fig2:Third Party auditing scheme

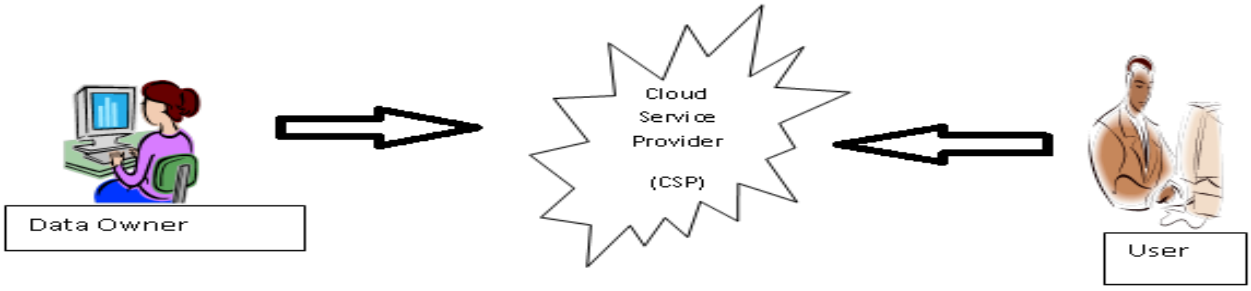
III. PROBLEM IN EXISTING SYSTEM

many researchers have attended the issue of data storage security in cloud which we broadly categorize into two groups one which make use of trusted third party auditor (TTPA) and other that do not. Normally TTPA is a reliable independent component which is trusted by both the cloud users and server many researchers recommend the support of trusted third party (TTP). By leaving the resource consuming cryptographic operations on TTP for achieving confidentiality and integrity, cloud users can be worry-free. But issues such as TTP becoming bottleneck, data leakage, introduction of new vulnerabilities, scalability, accountability, performance overhead, dynamic data support, extra hardware cost incurred etc. have motivated many researchers to address the data storage security problems without using trusted third party auditor

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 cloud user.
7. High Computational and communication cost in

PROPOSED SYSTEM

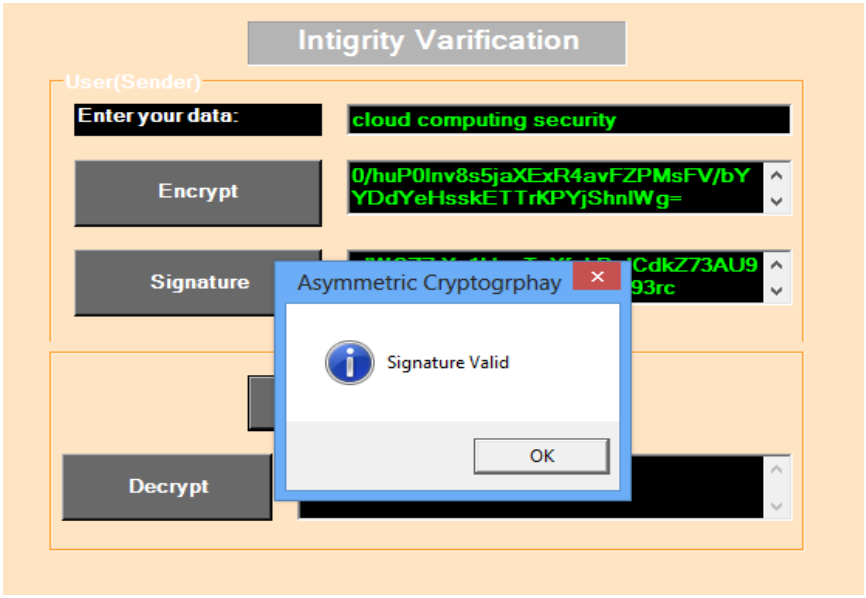


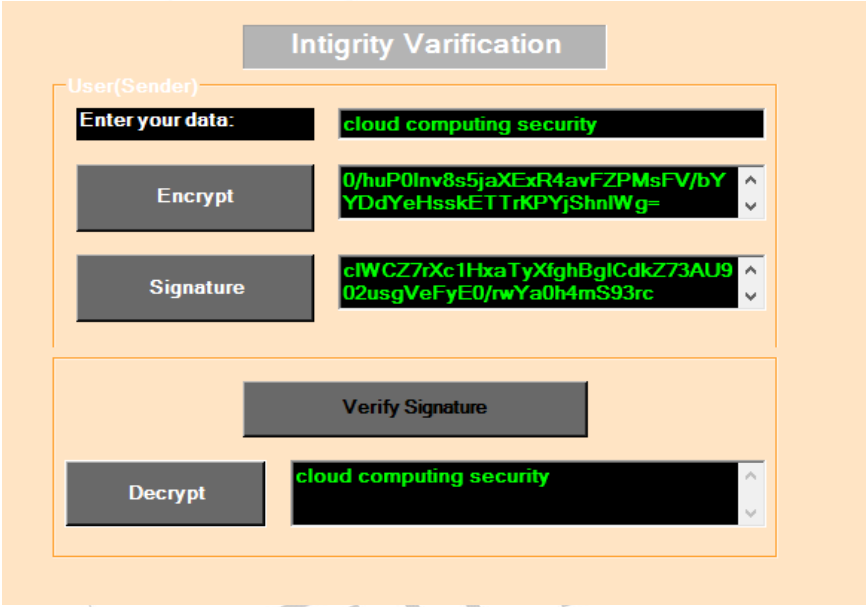
I propose a data storage security model, which intends to solve the data security problem, Integrity verification and File sharing problem.

My Propose System Contains three stakeholders like:

A. Data owner, who generates and owns the data, possessing all rights about file operation, it can pass on the same to other Cloud data users.
B. Cloud service provider (CSP), which is the central core component of the whole system. It also acts as a cloud data server.
C. User, who uses the data based on credentials received from the data owner.

1. Data owner generates key using Symmetric Key (DES, AES) and Asymmetric key generation (RSA) algorithm and store that key , and encrypt and decrypt data using that key stored in database. Data owner generate hash code (Signature) using cryptography hash functions Blake on Encrypted file and store that signature in database.
2. Data owner upload encrypted file on cloud Service provider (CSP). If later data owner want to verify that file on CSP they send request to CSP. So CSP calculates hash code for the encrypted file which is uploaded by the DO and sends it to DO.
3. DO compare the hash code received by CSP with the actual hash code to check the correctness of data which is stored on the CSP.
4. CSP decrypt file using Symmetric key (DES, AES) and Asymmetric key (RSA) generation algorithm and send to DO. And DO requests for view/download the file.
5. DO Grant file Access Rights (Sharing of file) to other cloud user.





VI. CONCLUSIONS

In this Scheme provides encrypt and decrypt data using Symmetric (AES, DES) and Asymmetric (RSA) Algorithms and use hash function for generating hash code.This system provides high security, lightweight data integrity verification, data hiding and secure access right to other cloud data file requester.In future this proposed scheme should be enhance for large data style. We provide mechanism for Cloud Data Requester to access file on Cloud Server. This scheme work faster and secure to check integrity of data on cloud server.

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)