Public Auditing for Secure and Efficient Cloud Data Storage: A Comprehensive Survey

[1]Ayesha Siddiqha Mukthar, [2]Dr. Jitendra Sheetlani
[1]Research Scholar, Sri Satya Sai University of Technology and Medical Sciences, Sehore
[2]Associate Professor, Sri Satya Sai University of Technology and Medical Sciences, Sehore

Abstract:

Nowadays storage of data is big problem because the huge generation of multimedia data likes images, audio, video etc. whose size is very large. For storing of these data size of conventional storage is not sufficient so we need remote storage such as cloud which is resilient infrastructure, reliable and high quality performance for the cloud users. In the cloud there is no direct physical control over the records because the cloud uses its resource pool for storing. Consequently data reliability fortification and auditing is not a modest task. The user prerequisites to depend on a Third Party Auditor (TPA) who is working as a public auditor for authenticating the data integrity and privacy. This paper presents the various auditing techniques of cloud computing for improving security and then future research challenges which need to be adopt by researchers to make system obvious.

Keywords: Auditing, Cloud Computing, Storage, TPA, Reliability, Integrity.

Introduction:

Cloud is offering the different services to its users. Data sharing between two organizations which common in many application areas. The current data sharing and integration among various organizations requires the central and trusted authority to collect data from all data sources and then integrate the collected data. In current trend, there is necessary condition which defines the data sharing while preserving privacy in cloud. With cloud computing, it is mandatory for data to be not only stored in the cloud, but also it shared across multiple users. For this purpose, defined many different data sharing techniques which are developed in cloud environment. This concept resolved the information assurance, network security, data protection, and privacy concerns. Computing is the process which describes the combination of a set of software infrastructure, framework, and middleware services. That allowed sharing and selection of resources.[1] Storing large amounts of data with cloud service providers (CSPs) raises concerns about data protection. Data integrity and privacy can be lost because of the physical movement of data from one place to another by the cloud administrator, malware, dishonest cloud providers, or other malicious users who might distort the data.[2] Hence, saved data corrections must be verified at regular intervals. Nowadays, with the help of cryptography, verification of remote (cloud) data is performed by third-party auditors (TPAs).[3] TPAs are also appropriate for public auditing, offering auditing services with more powerful computational and communication abilities than regular users.[4] In public auditing, a TPA is designated to check the correctness of cloud data without retrieving the entire dataset from the CSP. However, most auditing schemes don't protect user data from TPAs; hence, the integrity and privacy of user data are lost.[2] Our research focuses on various cloud auditing techniques for security of cloud storage, integrity and privacy issues faces by these techniques.

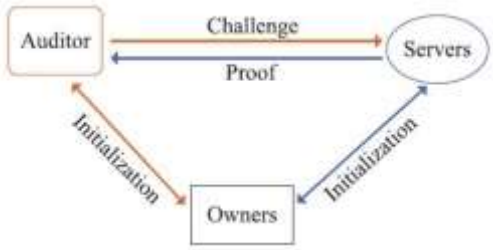


Fig.1: Third party auditing system workflow

1.1 Characteristics of Cloud Computing

There are basically five essential characteristics of Cloud Computing.

1. ***On-demand self-services:***
   The Cloud computing services does not require any human administrators, user themselves are able to provision, monitor and manage computing resources as needed.
2. ***Broad network access:***
   The Computing services are generally provided over standard networks and heterogeneous devices.
3. ***Rapid elasticity:***
   The Computing services should have IT resources that are able to scale out and in quickly and on as needed basis. Whenever the user require services it is provided to him and it is scale out as soon as its requirement gets over.
4. ***Resource pooling:***
   The IT resource (e.g., networks, servers, storage, applications, and services) present are shared across multiple applications and occupant in an uncommitted manner. Multiple clients are provided service from a same physical resource.
5. ***Measured service:***
   The resource utilization is tracked for each application and occupant, it will provide both the user and the resource provider with an account of what has been used. This is done for various reasons like monitoring billing and effective use of resource [30,33].

1.2 Challenges in Cloud Computing

As cloud provides many advantages but as every coin has 2 side, and cloud computing is no exception, it also has certain challenges. Every day, a fresh news item, latest publication, blog entry, highlights the cloud computing's challenges and issues. In each technology there are some security issues that affect the usage and the behavior below some of these concerns in the cloud: [5]

Access: When there is an unauthorized access to the data, the ability of altering on the client data arise.

Availability: The data must be available all the time for the clients without having problems that affect the storage and lead to the client data lose.

Network Load: The over load capacity on the cloud may drop the system out according to the high amount of data between the computers and the servers.

Integrity: The data correctness, legality and security is the most fields that influence on the cloud and have major lay on the service provider.

Data Location: The client does not know the actual place that the data saved or centered in because it distributed over many places that led to confusion.

One of the important concerns in the cloud computing that need to be addressed is to assure the customer of the integrity, accordingly in the next section will discuss about data integrity.

2. Cloud Storage Architecture

Cloud storage architectures are primarily about delivery of storage on demand in a highly scalable and multi-tenant way. [6] Generically, cloud storage architectures consist of a front end that exports an API to access the storage. In traditional storage systems, this API is the SCSI protocol; but in the cloud, these protocols are evolving. There, you can find Web service front ends, file-based front ends, and even more traditional front ends (such as Internet SCSI, or iSCSI). Behind the front end is a layer of middleware that I call the storage logic. This layer implements a variety of features, such as replication and data reduction, over the traditional data-

placement algorithms (with consideration for geographic placement). Finally, the back end implements the physical storage for data. This may be an internal protocol that implements specific features or a traditional back end to the physical disks.

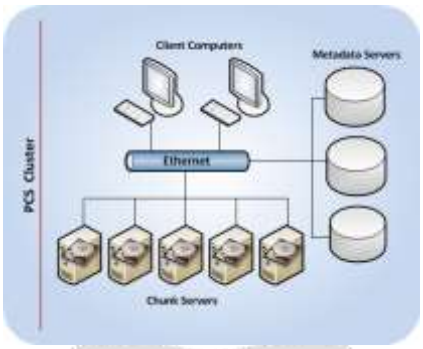


Fig. 2 Cloud Storage Architecture

3. Third Party Auditing Techniques in Cloud Storage

3.1 Hash based message authentication code (HMAC)

Hash based message authentication code is cryptographic hash function where the message and key are hash them together. By using secret key, we can calculate the hash function of message authentication code. SHA is a hash algorithm with is used to generate the authentication code for the message. It is used to check the message authentication by using secret key and verify the data integrity. The strength of HMAC is determined by strength of hash function, hash output size and key size. HMAC supports for has algorithms like MD5, SHA-1, SHA-256, etc.

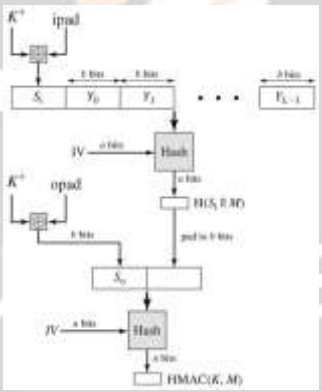


Fig. 3: Flow diagram of HMAC

Which implement the function:

$HMAC_k$= Hash [(K^+ XOR opad)] || Hash [(K^+XORipad) || M]

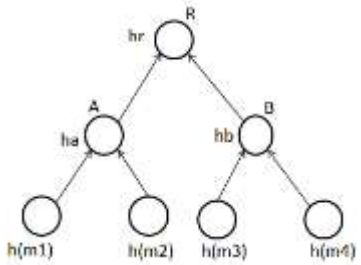
K^+ is K padded with zeros on the left so that the result is b bits in length ipad is a pad value of 36 hex repeated to fill block opad is a pad value of 5C hex repeated to fill block M is the message given as input to HMAC[7].

The output binary authentication code which equals in the length to that of the hash function digest. The data integrity of the file is checked by comparing the value of hash function in both user and auditor. In HMAC, the generation of authentication code uses secret and hash based algorithm. This code ensures the usage of hash function is expanded in many places. It conserves the performance of hash function.

3.2 Merkle Hash Tree (MHT)

A Merkle Hash Tree is a well-studied structure used for authentication purpose [8], which is intended to prove efficiently that a set of elements are unaltered and undamaged. It is used for decreasing the server computation time [9]. It is used by cryptographic methods to authenticate the file blocks. The tree is constructed as a binary tree where the leaf nodes are the hashes of the authentic data values i.e. the original file blocks. The idea used in this is to break the file up into a number of small pieces, apply hash to these pieces and the combine iteratively and rehash the resulting hashes in a treelike fashion until we get a tree with a single 'root hash'. The MHT is generated by the client and is stored at both the client and the server side. An example of the MHT structure is

as shown Fig 4. Among it, ha=h(h(m_1)||h(m_2)) and hb=h(h(m_3)||h(m_4)), where h is a secure one-way hash function.



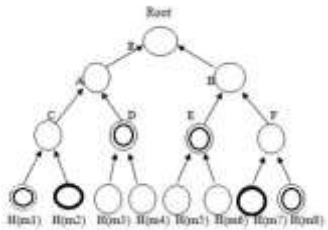


Fig. 4 Merkel hash tree (MHT) Fig.5.Authentication of data elements using MHT

The authentication of the file blocks is done by the client by requesting the server to send block related information for generating the tree. This information is called as the Auxiliary Authentication Information (AAI). For example, consider the MHT in Fig.5. The verifier with the authentic root h_r requests for {m_2,m_7} and requires the authentication of the received blocks. The AAI Ω_2= and is Ω_7= are provided by the prover to the verifier. The verifier can now verify m_2 and m_7 by computing h(m_2), h(m_7), h_d=h((h(m_3)||h(m_4)), h_e=h(h(m_5)||h(m_6)), h_a=h(h_c||hd), hb=h(h||hf) and hr=h(h_a||h_b).

3.3 Message Authentication Code (MAC) Method

Assume the outsourced data file F consists of a finite ordered set of blocks m1; m2; . . .; mn. One straightforward way to ensure the data integrity is to precompute MACs for the entire data file. Specifically, before data outsourcing, the data owner precomputes MACs of F with a set of secret keys and stores them locally. During the auditing process, the data owner each time reveals a secret key to the cloud server and asks for a fresh keyed MAC for verification.[10] This approach provides deterministic data integrity assurance straightforwardly as the verification covers all the data blocks. However, the number of verifications allowed to be performed in this solution is limited by the number of secret keys. Once the keys are exhausted, the data owner has to retrieve the entire file of F from the server in order to compute new MACs, which is usually impractical due to the huge communication overhead. Moreover, public auditability is not supported as the private keys are required for verification.[10]

3.4 BLS Signature

BLS digital signature also known as Boneh–Lynn–Shacham[11] (BLS) is cryptographic signature scheme which allows a user to verify that a signer is authentic. The scheme uses a bilinear pairing for verification, and signatures are elements of an elliptic curve group. Working in an elliptic curve group provides some defense against index calculus attacks (with the caveat that such attacks are still possible in the target group G_T of the pairing), allowing shorter signatures than FDH signatures for a similar level of security.
Signatures produced by the BLS signature scheme are often referred to as short signatures, BLS short signatures, or simply BLS signatures.[12] The signature scheme is provably secure (the scheme is existentially unforgeable under adaptive chosen-message attacks) assuming both the existence of random oracles and the intractability of the computational Diffie–Hellman problem in a gap Diffie–Hellman group.[13]

3.5 Homomorphic Encryption

A Homomorphic Encryption system is used to perform operations only on encrypted data not on decryption data. While performing operations, it does not know the users private key. It knows only the users secret key. While performing the calculation on raw data it should be same as decryption data.[32]
Definition: An encryption is Homomorphic, if: from Enc(a) and Enc(b) it is possible to compute Enc(f (a, b)), where f can be: +, ×, ⊕ and without using the private key. Homomorphic encryption has Additive Homomorphic encryption and multiplicative Homomorphic encryption. Additive Homomorphic encryption is the Pailler[14] and multiplicative Homomorphic encryption is the RSA[15] and ElGammal cryptosystems[16].
For calculating any calculation in the cloud fully Homomorphic encryption is used based on encrypted data not on decrypted data. Fully Homomorphic encryption is an important for cloud for providing Cloud Computing security and keeps the data more confidential. Decryption is also based on client secret key. By working with cloud server uses virtual platform ESX and a VPN network that links to the client. By simulating different scenarios using the Computer Algebra System Magma tools [17].

It focusing on

- The size of the public key and its impact on the size of the encrypted message.
- The server delay of the request treatment according to the size of the encrypted message.
- The result decrypting time of the request according to the cipher text size sent by the server.

3.6 Provable Data Possession (PDP)

A PDP scheme checks that a remote cloud server retains a file, which consists of a collection of n blocks. The data owner processes the data file to generate some metadata to store it locally. The file is then sent to the server, and the owner delete the local copy of the file. The owner verifies the possession of file in a challenge response protocol.
Characteristics of PDP Schemes are: In PDP Scheme the client verifies the data stored at a server and still possesses the data without retrieving it.[20] In PDP scheme the server does not actually have to access the file blocks, supporting Blockless verification. PDP scheme not includes error-correcting codes to address concerns of corruption. PDP scheme lends itself more naturally to data that may undergo slight changes as dynamic expansion is supported. Ateniese et al. [25] are the first to consider public auditability in their defined "provable data possession" model for ensuring possession of files on untrusted storages. In their scheme, they utilize Homomorphic Verifiable Tags for auditing outsourced data, thus public auditability is achieved. However, Ateniese et al. do not consider the case of dynamic data storage, and the direct extension of their scheme from static data storage to dynamic case may suffer design and security In their subsequent work [26], Ateniese et al. proposed a dynamic version of the prior PDP scheme problems. However, the system imposes a priori bound on the number of queries and does not support fully dynamic data operations, i.e., it only allows very basic block operations with limited functionality, and block insertions cannot be supported. In [21], Wang et al. consider the proposed challenge-response protocol can both determine the data correctness and locate possible errors. Similar to [26], they only consider partial support for dynamic data operation. Erway et al. [22] were the first to explore constructions for dynamic provable data possession. They extend the PDP model in [25] to support provable updates to stored data files using rank-based authenticated skip lists. This scheme is essentially a fully dynamic version of the PDP solution. To support updates, especially for block insertion, they eliminate the index information in the "tag" computation in Ateniese's PDP model [25] and employ authenticated skip list data structure to authenticate the tag information of challenged or updated blocks first before the verification procedure. However, it's computational and communication complexity is both up to log (n). Feifei Liu[24] were proposed an improved dynamic model that reduce the computational and communication complexity to constant by using Skip-List, Block, Tag and Hash method.

3.7 Proof of Retrievability (PoR)

In PoR Scheme a cloud server proves to a data owner that a target file is intact, in the sense that the client can retrieve the entire file from the server with high probability. Hence, PoR guarantees not only correct data possession but it also assures retrievability upon some data corruptions. Characteristics of PoR Scheme are:
This scheme purposed only works with static data sets. It is more suited for storage of large, unchanging data.
This scheme also includes error-correcting codes to address concerns of corruption. It supports only a limited number of queries as a challenge since it deals with a finite number of check blocks (sentinels).
Juels and Kaliski [18] describe a "proof of retrievability" model, where spot-checking and error-correcting codes are used to ensure both "possession" and "retrievability" of data files on archive service systems. Specifically, some special blocks called "sentinels" are randomly embedded into the data file F for detection purpose, and F is further encrypted to protect the positions of these special blocks. The number of queries a client can perform is also a fixed priori, and the introduction of precomputed "sentinels" prevents the development of realizing dynamic data updates. In addition, public auditability is not supported in their scheme. Shacham and Waters [11] design an improved PoR scheme with full proofs of security in the security model defined in [19]. They use publicly verifiable homomorphic authenticators built from BLS signatures [23], based on which the proofs can be aggregated into a small authenticator value, and public retrievability is achieved. Still, the authors only consider static Data files.

3.8 Cooperative provable data possession (CPDP)

This technique address the construction of an efficient PDP scheme to gain the scalability of service and data migration in distributed cloud storage system, in which this technique consider multiple cloud service providers to collectively store and maintain the clients' data [29][30].

3.9 Dynamic provable data possession (DPDP)

It expands the PDP model to allow provable updates to stored data. It uses a new version of authenticated dictionaries which are based on rank information. The value of dynamic updates is a performance change from O(1) to O(log n) (or O(nlog n)), to detect the misbehavior with the same probability for a file with n blocks. Their experiments show that this slowdown is very low in practice.

4. Conclusion

The security and privacy of cloud data storage becomes very challenging nowadays and for maintaining it various techniques has been developed and proposed by the researchers. In this paper we have discusses various cloud auditing techniques such as Merkel Hash Tree (MHT), MAC, BLS Signature, Homomorphic Encryption etc. After study, it is observed that in every techniques some improvement has been found to improve the security over cloud storage but together with they also added some limitation like computation cost and overhead increases which degraded the performance of system and consume lots of resources. So in future work, we do develop such method which can efficiently improve the performance and also overcome these limitations of above techniques.

5. References

[1] Sangeetha, Saranya, "Survey of Security Auditing Issues in Cloud Computing", International Journal of Electrical Electronics & Computer Science Engineering Volume 1, Issue 5 (October 2014), ISSN : 2348 2273.
[2] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," Proc. IEEE 5th Int'l Conf. Cloud Computing (CLOUD 12), vol. 2, no.1, 2012, pp. 295–302.
[3] C. Wang et al., "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc. 29th Conf. Information Communications (INFOCOM 10), 2010, pp. 525–533.
[4] C. Wang et al., "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, 2013, pp. 362–375.
[5] Nandini J., Sugapriya N. P., M. S. Vinmathi, "Secure Multi-Owner Data Storage with Enhanced TPA Auditing Scheme in Cloud Computing", International Journal of Advances in Computer Science and Cloud Computing, ISSN: 2321-4058, Vol. 2, Issue: 1, MAY 2014.
[6] Gurudatt Kulkarni et al., "Cloud Storage Architecture", International Conference on Telecommunication Systems, Services, and Applications (TSSA), 2012. In proceeding of IEEExplore.
[7] Ramya. D, Dr. Raja. K, Srinivasan. S, "An Analysis of Third Party Auditing Techniques In Cloud Computing", International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 11, November 2014.
[8] Cong Wang, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing", IEEE Transactions on Parallel and Distributed Systems, May 2011.
[9] P. Golle, S. Jarecki, and I. Mironov "Cryptographic primitives enforcing communication and storage complexity". In Financial Cryptography, pages 120-135, 2002.
[10] T S Khatri, G B Jethava, "Survey on data Integrity Approaches used in the Cloud Computing", International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 9, November - 2012
ISSN: 2278-0181.
[11] Barreto, Paulo S. L. M.; Kim, Hae Y.; Lynn, Ben; Scott, Michael (August 2002). "Efficient Algorithms for Pairing-Based Cryptosystems". Advances in Cryptology — CRYPTO 2002. CRYPTO '02: Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology. Springer-Verlag Berlin Heidelberg. pp. 354–369.
[12] Jump up to: a b "Ethereum 2.0 Phase 0 -- The Beacon Chain : BLS Signatures". 28 July 2020. Retrieved 4 September 2020.
[13] Jump up to:a b Dan Boneh; Ben Lynn & Hovav Shacham (2004). "Short Signatures from the Weil Pairing". Journal of Cryptology. 17 (4): 297–319. CiteSeerX 10.1.1.589.9141. doi:10.1007/s00145-004-0314-9.
[14] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In 18th Annual Eurocrypt Conference
(EUROCRYPT'99), Prague, Czech Republic , volume 1592, 1999.
[15] R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. Communications of the ACM, 21(2) :120-126, 1978. Computer Science, pages 223- 238. Springer, 1999.
[16] TaherElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information
Theory, 469-472, 1985.

[17] WiebBosma, John Cannon, and Catherine Playoust. The Magma algebra system I: The user language. J. Symbolic Comput., 24(3-4): 235-265, 1997. Computational algebra and number theory, London, 1993.

[18]A. Juels and B.S. Kaliski Jr., "Pors: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, 2007.
[19]Bo Chen and Reza Curtmola. "Robust Dynamic Provable Data Possession," 1545-0678/12 $26.00 © 2012 IEEE.
[20] B. Priyadharshini and P. Parvathi, "Data Integrity in Cloud Storage", ISBN: 978-81-909042-2-3 ©2012 IEEE
[21]C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS '09), 2009.
[22]C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia,"Dynamic Provable Data Possession," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), 2009.
[23] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '01), pp. 514-532, 2001.
[24]Feifei Liu, Dawu Gu, Haining Lu," An Improved Dynamic Provable Data Possession Model," 978-1-61284-204-2/11/$26.00 ©2011 IEEE
[25]G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 598-609, 2007.
[26] 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. Networks (SecureComm '08), pp. 1-10.
[27]He-Ming.Ruan, Yu-Shian Chen, Chin-Laung Lei, "Data Integrity on Remote Storage for On-line Coworking," 978-0-7695-4776-3/12 $26.00 © 2012 IEEE
[28]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.
[29] Y. Zhu, H. Hu, G. Ahn, and M. Yu, "Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage," IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 12, pp. 2231-2244, Dec. 2012.
[30] M. Ravi kumar and E . Madhusudhana Reddy "Auditing Framework Service for Efficient Secure Data Storage in Multi- cloud" (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (2) , 2015, 1181-1183.
[31] Dr Salimath, Dr C Kavitha, Dr Sheetlani, "Analysis of Resource Management and Security Management in Cloud Computing Environment", Analysis of Resource Management and Security Management in Cloud Computing Environment (May 17, 2019).
[32] Nagesh Salimath, Satishkumar Mallappa, Neelamadhab Padhy, Jitendra Sheetlani, "Scrambling and descrambling of document image for data security in cloud computing", Smart Intelligent Computing and Applications, 2020, pp 283-290.
[33] Harsh Pratap Singh, Rashmi Singh, Vinay Singh, "Cloud Computing Security Issues, Challenges and Solutions", EasyChair, 2020, Issue 2533