7th International Conference on Recent Trends in Engineering, Science & Management
Genba Sopanrao Moze College of Engineering, Balewadi-Baner, Pune (ICRTESM-17)
01st-2nd April 2017, www.conferenceworld.in ISBN: 978-93-86171-12-2

PRIVACY PRESERVING AUDITING OF ENCRYPTED CLOUD DATA BY USING THIRD PARTY AUDITOR

Prof. Pradnya N. Gulhane

Computer Engg. Dept. G.S. Moze college of Engineering, Balewadi Pune

ABSTRACT

Cloud computing is an internet based computing. Using cloud computing, user can place their data remotely and enables sharing of services. Many users place their data on the cloud. Hence correctness and security of data is a prime concern. Physical possession of the outsourced data is restricted to users. It ensures the data integrity protection in Cloud Computing. Ensuring integrity is a difficult task for users with limited number of computing resources. With cloud computing user can use cloud storage like a local storage which verifies integrity. A third party auditor checks the integrity of a data stored on cloud and enables the public audit ability for cloud storage which is a crucial task. The auditing process does not bring any new vulnerability towards user data privacy and does not introduce any additional online burden to user to securely introduce an effective TPA. Here, proposed a secure cloud storage system using Third Party Auditor on encrypted cloud data which will preserve privacy. It is further extended to enable the TPA to perform audits for multiple users simultaneously and efficiently. Wide-spread security and performance analysis give an idea about the proposed scheme is secure and highly capable.

Keywords: **cloud computing, cloud service provider, cryptographic protocols, public audit ability , Third Party Auditor**

I. INTRODUCTION

Cloud computing is the result of evolution and adoption of existing technologies and paradigms. The goal of cloud computing is to allow users to take benefit from all existing technologies, without the need for wide knowledge. The main goal of cloud is to reduce cost and bring the users focus on their core business instead of being impeded by IT obstacles. There is a huge list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk [3]. Cloud computing as a technology with profound implications is transforming the nature of the way businesses use information technology. Data is being centralized or outsourced to the Cloud is one of the fundamental aspect of this paradigm shifting. Both IT enterprises and individuals, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: 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.[4]. While these advantages of Cloud Computing are more appealing than ever, it also brings new and challenging security threats towards users' outsourced data. Cloud Service Providers (CSP) are separate administrative entities and hence data outsourcing is actually relinquishing user's ultimate control over the fate of their data. It results into

putting at risk the correctness of the data in the cloud due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time [4]–[9]. Secondly, there exists various motivations for CSP to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For examples, CSP might reclaim storage for monetary reasons by discarding data that has rarely accessed or not accessed, or even hide some data loss incidents to maintain reputation [10]-[12].In short, even if it is true that outsourcing data to the cloud is economically attractive for long-term large-scale data storage, but it does not immediately offer any guarantee on data integrity and availability. In case this problem is not properly addressed, then it may impede the successful deployment of the cloud architecture.

Traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted [13] as users no longer physically possess the storage of their data. In particular, due to the expensiveness in I/O and transmission cost across the network, downloading all the data only for its integrity verification is not a practical solution. It is often insufficient to detect the data corruption only by accessing the data as it does not give users correctness assurance for the un-accessed data and might be too late to recover the data loss or damage. The tasks of auditing the data correctness in a cloud environment can be formidable and expensive for the cloud users considering the large size of the outsourced data and the user's constrained resource capability. Moreover, the overhead of using cloud storage should be minimized as much as possible, such that user does not need to perform too many operations to use the data (in additional to retrieving the data).

For example, it is desirable that users do not need to worry about the need to verify the integrity of the data before or after the data retrieval. More than one user can access the same cloud storage, say in an enterprise setting. TPA audit the outsourced data when needed to make the task easier. The TPA is an expertise as well as has capabilities that users do not. It can periodically check the integrity of all the data stored in the cloud on behalf of the users, which is the most easier and affordable way for the users to ensure their storage correctness in the cloud. It also helps to evaluate the risk of subscribed users cloud data services. Audit result from TPA would also be beneficial for the cloud service providers to improve their cloud based service platform. It also serves independent arbitration purposes [11].

It is summarized as the following:

It motivates the public auditing system of data storage security in Cloud Computing and provides a privacy-preserving auditing protocol. It enables an external auditor to audit user's outsourced data in the cloud without learning the data content.

2) It supports scalable and efficient public auditing in the Cloud Computing. Specifically, it achieves batch auditing where multiple delegated auditing tasks from different users can be performed simultaneously by the TPA.

3) It improves the security and justify the performance of proposed schemes through concrete experiments and comparisons with the state-of-the-art.

II. RELATED WORK

There are two different way for integrity checking of cloud data. The first one is a MAC-based solution in which user selects the secret key for creating MAC for data which is used to store on cloud. It suffers from undesirable systematic demerits – bounded usage and stateful verification, which may pose additional online burden to users, in a public auditing setting. This somehow also shows that the auditing problem is still not easy to solved even we use TPA for auditing . The second one is a system based on homomorphic linear authenticators (HLA), which covers many recent proof of storage systems. All existing HLA-based systems are not privacy-preserving. The analysis of these basic schemes leads to main result, which overcomes all these drawbacks. Here main scheme is based on a specific HLA scheme.

Mac-Based Solution

There are two possible methods to make use of MAC to authenticate the data. In the first method, cloud users just store the data blocks with their MACs to the server, and sends the corresponding secret key *sk* to the TPA. Later, the TPA can randomly retrieve blocks with their MACs and check the correctness via sk. TPA requires the knowledge of the data blocks for verification and it can easily access the data on cloud and so leak the data on cloud.

In the second method, by restricting the verification which consists of only the quality checking, requirement of the data in TPA verification can be avoided. The procedure is given below: Cloud user selects s random message authentication code keys {sk$T$}1≤T_≤s before storing data on the cloud. It pre-computes *s* MACs, {MAC sk$T$ (F)}1≤ T ≤s for the entire data file F and forwards verification metadata to TPA. TPA reveals a secret key sk$T$ to the cloud server and asks for a fresh keyed MAC for comparison in each audit. It preserves the privacy until F is recovered in full given MACsk$T$ (F) and sk$T$. But, it has the following severe drawbacks:

1) Number of times a particular data file can be audited is limited by the number of secret keys that must be fixed a priori. Once all possible secret keys are exhausted, the user can retrieve data in full to re-compute and re-publish new MACs to TPA.

2) The TPA also has to maintain and update state between audits by keeping track on the revealed MAC keys. There will be large number of audit delegations from multiple users. To maintain such states for TPA can be difficult and error prone.

3) It can only support static data, and cannot efficiently deal with dynamic data at all. However, supporting data dynamics is also of critical importance for cloud storage systems.

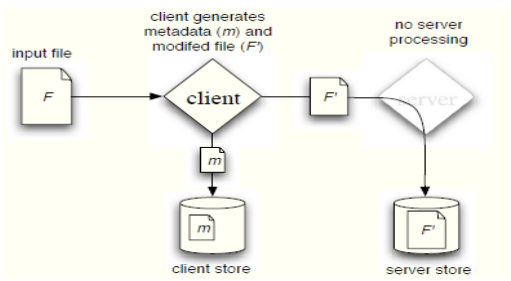

Fig 1: Pre-process and store [9]

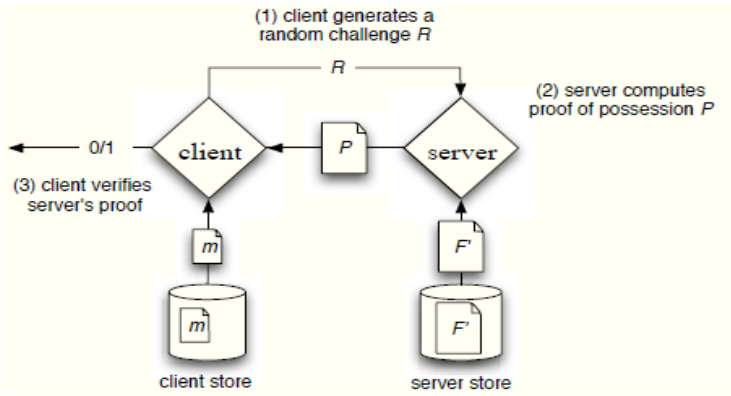

Fig 2: Verify server possession [9]

HLA-Based Solution

HLA technique [10], [12], [15] can be used to effectively support public audit ability without retrieving actual data on cloud. HLAs, like MAC's also use metadata to authenticate integrity of a data block. It uses homomorphic verifiable tags. Tags computed for multiple file blocks can be combined into a single value due to homomorphic property. The client pre-computes tags for each block of a file and then stores the file and its tags with a server. At a later time, the client can verify that the server possesses the file by generating a random challenge against a randomly selected set of file blocks. Using the queried blocks and their corresponding tags, the server generates a proof of possession. The client is thus convinced of data possession, without actually having to retrieve file blocks.

A PDP protocol (Fig. 1) checks that an outsourced storage site retains a file, which consists of a collection of n blocks. The client C (data owner) pre-processes the file, generating a piece of metadata that is stored locally, transmits the file to the server S, and may delete its local copy. The server stores the file and responds to challenges issued by the client. As part of pre-processing, the client may alter the file to be stored at the server. The client may expand the file or include additional metadata to be stored at the server. Before deleting its local copy of the file, the client may execute a data possession challenge to make sure the server has successfully stored the file. Later on, (in fig. 2) the client issues a challenge to the server to establish that the server has retained the file. The client requests that the server compute a function of the stored file, which it sends back to the client. Client verifies the response using it's local metadata.

This scheme is a collection of four algorithms (KeyGen,TagBlock,GenProof , CheckProof)[8] such that:

KeyGen (1k) → the key generation algorithm is run by the client. It takes a security parameter k as input, and returns a pair of matching public and secret keys (pk, sk).

TagBlock (pk, sk,m) → Tm is an algorithm run by the client to generate the verification metadata. Input given to it: a public key pk, a secret key sk and a file block m and it returns the verification metadata Tm.

GenProof (pk, F, chal, Σ) →Server run this algorithm and generates a proof of possession V. It takes as inputs a public key pk, an ordered collection F of blocks, a challenge chal and an ordered collection Σ which is the verification metadata corresponding to the blocks in F. It returns a proof of possession V for the blocks in F which is determined by the challenge chal.

CheckProof (pk, sk, chal, V) → {"success", "failure"} is run by the client in order to validate a proof of possession. It takes input: a public key pk, a secret key sk, a challenge chal and a proof of possession V and returns whether V is a correct proof of possession for the blocks determined by chal.

A PDP system contains PDP scheme of two phases, Setup and Challenge:

Setup: The client C is in possession of the file F and runs (pk, sk) ← KeyGen(1k), followed by Tmi ← TagBlock(pk, sk,mi) for all $1 \leq i \leq n$. C stores the pair (sk, pk). C then sends pk, F and _ = (Tm1 , . . . , Tmn) to S for storage and deletes F and _ from its local storage.

Challenge: C generates a challenge chal to indicate the specific blocks for which C wants a proof of possession. C then sends chal to S. S runs V ←GenProof (pk, F, chal,_) and sends to C the proof of possession V. Finally, C can check the validity of the proof V by running CheckProof (pk, sk, chal, V).

In the Setup phase, tags for each file block will get computed by C. It stores them together with the file at S. In the Challenge phase, C requests proof of possession for a subset of the blocks in F. This phase can be executed an unlimited number of times in order to check whether S still possesses the selected blocks.

III. IMPLEMENTATION DETAILS

Problem Definition

As shown in fig 3 Cloud data storage contain different entities like: the cloud user (U), who want to store the large amount of data file on cloud, the cloud server (CS), which is managed by the cloud service provider (CSP) to provide data storage service. It has significant storage space and computation resources .The *third party auditor* (TPA) has capabilities that cloud users do not have. It is trusted to assess the cloud storage service reliability on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance. For various application purposes , TPA's may dynamically interact with the CS to access and update their stored data. Cloud users may rely on TPA to ensure the storage integrity of outsourced data which helps to save computation resource as well as online burden along with keeping their data private from TPA.

Here considered the existence of a semi-trusted CS [16] which does not deviate from the prescribed protocol execution. But CS might neglect for it's own benefit to keep or delete rarely accessed data files which belong to ordinary cloud users. Moreover, the CS may decide to hide the data corruptions caused by server hacks. Also it may hide Byzantine failures to maintain reputation. Here TPA, who is in the business of auditing independent but not reliable. It will interact with CSP during the auditing process. However, it may harm the user if the TPA could learn the outsourced data after the audit. To authorize the CS to respond to the audit delegated to TPA's, the user can sign a certificate granting audit rights to the TPA's public key, and all audits from the TPA are authenticated against such a certificate.

In existing system, before uploading file it is divided into 7 blocks creating its authenticator for each block and tag for file then upload all data blocks its authenticators and tag on cloud. For auditing TPA access all authenticators from cloud as well as access all authenticators generated by the cloud service provider. Out of this any 5 authenticators are linearly verified by TPA. By using existing system the integrity of file is not checked 100%.Also For downloading file first it all data blocks are integrated into single main block then downloaded this block . It require more time to upload and download data.

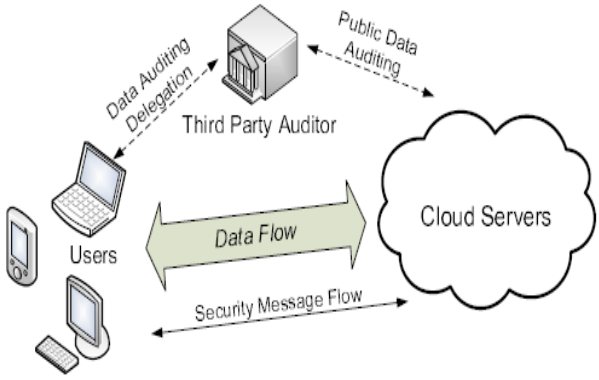

Fig 3: The architecture of cloud data storage service [1]

Proposed Algorithm

For its implementation, The proposed algorithm is divided into two phases. First phase is set up phase which is before uploading file, it is divided into maximum 3 blocks. For each block authenticator is created and for each file tag is created. Then all data blocks with their authenticators and tag are uploaded on cloud. If the file size is 2.99 MB then only one block is created, If the file size is 5.99 MB then two blocks are created, if file size is greater than or equal to 6 MB then three blocks are created.So maximum 3 authenticators are created. It allows uploading and downloading of max three blocks.

. Second phase is Audit phase which is used for checking integrity of data stored on cloud without accessing actual data through TPA and CSP. Detailed description of each phase given bellow.

Setup Phase:

First user will divide file F into three blocks F= (m1,m2,m3).The cloud user runs KeyGen to generate secret parameter (*Sk*).By using Sk all three blocks are encrypted[2].

Given a data file F = (Em1,Em2,Em3), the user runs SigGen to compute authenticator (digest D) for each block m*i*.This digest D again encrypted by user using private key Pk and create new encrypted digest ED.The last part of SigGen is for ensuring the integrity of the unique file identifier name. One simple way to do this is to compute t = name||SSigssk(name) as the file tag for F, where SSigssk(name) is the signature on name under the private key ssk. Now user will upload all data (F,ED,Tag t) on cloud and delete its local copy.

For simplicity, it is assumed that the TPA knows the number of blocks n. The user then sends F along with the verification metadata to the server and deletes them from local storage.

Audit Phase:

As shown in Fig 4, TPA first retrieves the file tag t. According to the mechanism described in setup phase, TPA verifies the signatureSSigssk(name) via spk, and quits by FALSE if the verification fails. Otherwise, the TPA recovers name. Now it comes to the "core" part of the auditing process.User generate the challenge message "chal", for the audit and send to TPA, the TPA picks tag t and secret key Sk and send it with chal message to CSP. Upon receiving challenge chal = {(t,sk)} , the CSP runs GenProof to generate a response proof of data storage correctness. Now cloud service provider first searches file blocks using tag t .It creates digest D for a file block. This digest is encrypted by secret key Sk and create encrypted digest ED . It then

sends {t,ED} as the response proof of storage correctness to the TPA. When TPA gets response from CSP, it runs VerifyProof to validate the response. It checks whether the encrypted digest stored on cloud is as same as encrypted digest send by cloud service provider. If it is same then alert message will send to the user.

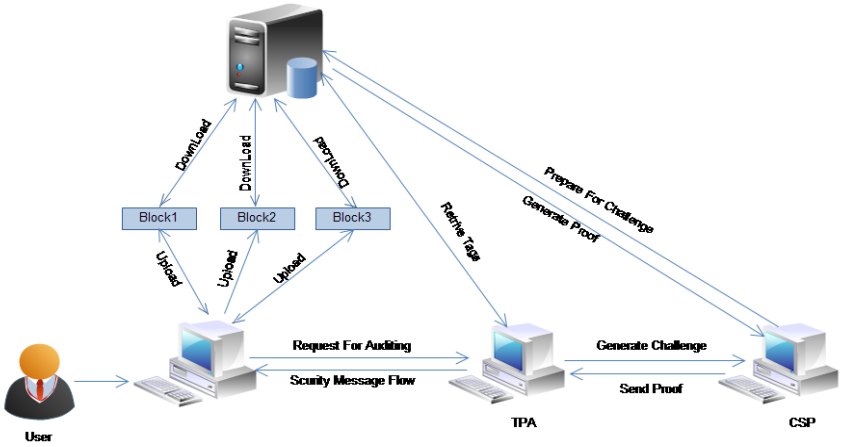

Fig 4: System architecture diagram

Here TPA want to verify 1, 2 or maximum 3 authenticators . So the time required to verifyproof algorithm is also reduced. Table 1 shows comparative result of existing and proposed system in term the file size, required time for verification in exsting system,

Table 1: verification time checking

File size for audit	Required time in existing system(in mSec)	Required time in proposed system(in mSec)
0.5	89	29
1	74	23
2	64	32
3	92	27
4	121	43
5	105	21
10	188	58
15	178	77
20	179	85
25	205	93

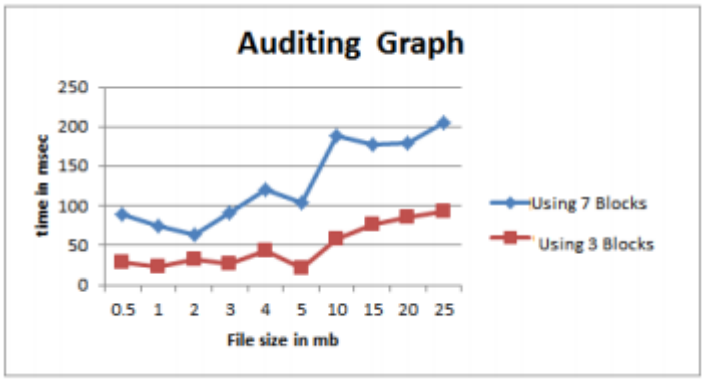

Figure 5: Result graph for audit

IV. RESULTS

With this system privacy is preserve of cloud data. User first wants to upload the file on cloud server stored securely (by encrypted file contents and create digest of it and this digest is again encrypted) With this system no one will access knowledge about data stored on cloud. With this technique integrity of cloud data is checked 100% and time required for file upload, download and auditing is minimized. Figure 5 shows auditing time difference for using 3 blocks and using 7 blocks.

V. CONCLUSION

In this work privacy-preserving public auditing system for data storage security in Cloud Computing . We utilize the homomorphic linear authenticator to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users fear of their outsourced data leakage. By using this technique required time for upload or download file is also minimize. File is divided into maximum 3 blocks. So the maximum 3 proofs are created for a file. Hence it reduces the time required to TPA for proof verification.

VI. FUTURE WORK

Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multiuser setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are provably secure and highly efficient.

REFERENCES

[1] Cong Wang, Sherman S.-M. Chow, Qian Wang, Kui Ren, Wenjing LouPrivacy-Preserving Public Auditing for Secure Cloud Storage IEEE TRANSACTIONS ON CLOUD COMPUTING YEAR 2013

[2] pradnya godhankar and Deepak gupta **"**Privacy Preserving Auditing of Encrypted Cloud Data by Using Third Party Auditor" cPGCON 2014 Vol 3 ISBN: 9789351072928

[3] P. Mell and T. Grance, "Draft NIST working definition of cloud computing," Referenced on June. 3rd, 2009 Online at http://csrc.nist.gov/groups/SNS/cloud-computing/index. html, 2009.

[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech.Rep.

[5] M. Arrington, "Gmail disaster: Reports of mass email deletions," Online at http://www.techcrunch.com/2006/ 12/28/gmail-disasterreports-of-mass-email-deletions/,December 2006.

[6] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors," Online at http://www.techcrunch.com/2008/07/10/mediamaxthelinkup-closes-its-doors/, July 2008.

[7] Amazon.com, "Amazon s3 availability event: July 20, 2008,"Online at http://status.aws.amazon.com/s3-20080720.html,2008.[6] S. Wilson, "Appengine outage," Online at http://www.cio-weblog.com/50226711/appengine outage.php, June 2008.

[8] S. Wilson, "Appengine outage," Online at http://www. cio-weblog.com/50226711/appengine outage.php, June 2008.

[9] B. Krebs, "Payment Processor Breach May Be Largest Ever,"Online at http://voices.washingtonpost.com/securityfix/2009/01/payment processor breach may b.html, Jan. 2009.

[10] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner,Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 598–609.

[11] M. A. Shah, R. Swaminathan, and M. Baker, "Privacypreserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008.

[12] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. of ESORICS'09, volume 5789 of LNCS. Springer-Verlag, Sep. 2009, pp. 355–370.

[13] A. Juels and J. Burton S. Kaliski, "Pors: Proofs of retrievability for large files," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 584–597.

[14] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing," 2009, http://www.cloudsecurityalliance.org.

[15] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. of Asiacrypt 2008, vol. 5350, Dec 2008, pp. 90–107.

[16] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan,"Auditing to keep online storage services honest," in Proc. ofHotOS'07. Berkeley, CA, USA: USENIX Association, 2007, pp.1–6.

[17] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure,scalable, and fine-grained access control in cloud computing,"in Proc. of IEEE INFOCOM'10, San Diego, CA, USA, March2010.