Privacy-Retaining Public Analyzing For Shared Data in the Cloud

Madhubala Rajendra Patil
ME Computer Engineering
G. S. MOZE College of Engineering Balewadi
Pune, Maharashtra India

Srinu Dharavath
Assistant Professor, Computer Engineering
G. S. MOZE College of Engineering Balewadi
Pune, Maharashtra India
sreevasmtech@gmail.com

Abstract—**In Cloud Storage, users can remotely store their data and have the demanded high quality apps and services. The originality of cloud data is subject to be a challenge due to the existence of hardware failures and software failures and human mistakes. Many systems have been implemented to allow both data owners and public verifiers to efficiently audit cloud data integrity. However, public analyzing on the integrity of shared data with these existing mechanisms will supports public analyzing on shared data stored in the cloud that exploit hash technique to compute verification metadata needed to audit the correctness of shared data. So that a third party auditor (TPA) is able to check the integrity of shared data for users with privacy retaining. Meanwhile, the identity of the signer on each block in shared data is kept private from the TPA in some special cases we can identify signer also we are able to perform multiple auditing tasks simultaneously instead of verifying them one by one .In this paper, we proved the freshness factor of data (proved the cloud possesses the latest version of shared data) while still retaining identity privacy. Our experimental result makes sure that retrieved data always gets the most recent updates and prevents roll-back attacks. Also we can add users dynamically and made it file centric instead of user centric to avoid obstacles. We have extended operations like upload, download, update to users other than owner by giving access.**

Keywords **Data Freshness, Public Auditing, Integirity Shared Data, Hash Technique, TPA, Hardware/Software Failures, Identify Signer.**

I. INTRODUCTION

With cloud computing we are able to use and to share resources offered by cloud service providers at a less cost. It is routine for users to have cloud storage services to use data with others in a group, as data sharing is unique feature in many cloud storage offerings, including Drop box, iCloud and Google Drive . The integrity of data in cloud storage, however, is subject to doubt and challenge, as data stored in the cloud can easily be lost or corrupted due to the inevitable hardware/software failures and human errors, The traditional approach for checking data correctness is to retrieve the entire data from the cloud, and then check data integrity by checking the correctness of the entire data. Certainly, this proper approach is able to successfully check the correctness of cloud data. However, the efficiency of using this previous approach on cloud data is not sure. The main obstacle is that the size of cloud data is large in general. Downloading the whole cloud data to check data integrity will cost or even waste users amounts of computation and communication sources, when data is corrupted in the cloud.

Recently, many mechanisms have been proposed to allow not only a data owner itself but also a public verifier to accurately perform integrity checking without downloading the whole data from the cloud, which is referred to as public analyzing. In these mechanisms, data is divided into many small blocks, where each block is uniquely signed by the owner; and a random combination of all the blocks instead of the whole data is retrieved during integrity checking. A public verifier could be a cloud data user who would like to utilize the owners data via the cloud or a third-party auditor (TPA) which provides perfect integrity verifying services. Existing public analyzing mechanisms can actually be extended to verify shared data integrity and data freshness.

However, a new important privacy issue introduced in this case of shared cloud data with the use of existing mechanisms is the leakage of identity privacy to public verifiers to protect the confidential information, it is important to preserve identity privacy from public verifiers during public auditing. To solve the above privacy issue on shared data, we propose novel privacy retaining public auditing mechanism so that a public verifier is able to verify the integrity of shared data without retrieving the entire data while the identity of the signer on each block in shared data is kept private from the public verifier. In addition, we have extended this mechanism which can perform multiple auditing tasks simultaneously and improve the efficiency of verification for multiple auditing tasks.

In existing system Ring signatures were used to compute verification metadata needed to audit the correctness of shared data. The identity of owner of each block in shared data was kept private from public verifiers so in some special cases we couldnt identify the author. Only data blocks were authenticated not their version. There were Static data means they were unable to add data dynamically.

II. LITERATURE SURVEY

B. Wang, , S.S. Chow,M. Li in Computing Encrypted Cloud Data Efficiently under multiple Keys, in 2013 had a mechanism named Secure Multiparty Computation (SMC) enables a set of users to evaluate certain functionalities on their respective inputs while keeping these inputs encrypted throughout the computation. In many scenarios, however, outsourcing these computations to an untrusted server is wanted to be done, so that the server can do the computation on behalf of the users. Unwantedly, the solutions we are having are either not efficient on user interaction, or needed the inputs to be encrypted by

the same key
M. Armbrust, A.D. Joseph, , A.Konwinski, G. Lee, D.A. Patterson,R.H. Katz, A. Rabkin, I. Stoica, and M. Zaharia, in A View of Cloud Computing, in April. 2010 described that it is the long waited dream of computing as a utility, has the potential to convert a major part of the IT industry, innovating software even more feasible as a service and giving the way IT hardware is designed and purchased. Developers with great new ideas for new Internet services are not require the large outlays in hardware to do their service or the human expense to perform it. They need not be worried about more provisioning for a service whose popularity does not meet their predictions, thus wasting resources which are more valuable, or under-provisioning for one that becomes wildly popular, thus not having potential customers and revenue.

III. PROPOSED APPROACH FRAMEWORK AND DESIGN

A. Problem Definition

Now a day there has been increase in use of Cloud data services. These are used everywhere for many purposes. The integrity of cloud data is main concerned so we need to check integrity of data. We check integrity of data to check correctness of data. Having the large size of the cloud data and the users capability of resources, the tasks of analysing the data correctness in a cloud environment can be difficult and expensive for the cloud users. However, the overhead of using cloud data storage should be kept minimum as much as possible, so that a user does not need to perform too many operations to use the data (in addition to retrieving the cloud data). In specific, users may not want to go through the hazard in verifying the data integrity. Apart from this, there may be more than one user accesses the same cloud data, say in an enterprise setting. For easier access, it is feasible that cloud only entertains verification request from only one designated party. There are many systems to check the correctness of data but no identity privacy is provided in any mechanism. In Public auditing there used to be reveal of confidential information. So privacy-retaining mechanism is proposed.

B. Proposed Architecture and Design

In this project, we propose the techniques providing more security by

- Creating Partitions: file is divided in to small parts which are called partitions. Partitions are created randomly for security purposes.
- Encrypting file: after creating partitions now encryption of file is taken place.
- Generating Hash and not signature: encrypted file goes under hashing. So hashes are created and those are stored for analyzing privacy.

To improve the efficiency of system independent third-party auditor (TPA) is used for all transactions. We are giving Access to more than one user this is possible because we are making our system files centric unlike users centric or group centric so we can add more no of users dynamically .Data blocks were authenticated as well as the latest version of shared data were added.

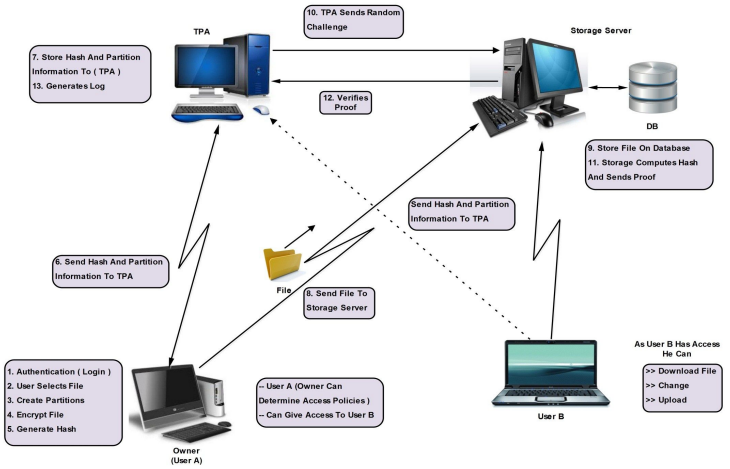

Fig. 1. System Architecture.

1) **Need of proposed system:**
There are some of the basic needs which are satisfied by our system
   - Data Sharing is standard feature of every organization so it is widely used everywhere.
   - Privacy Preserving of personal data as well as shared data is successfully done through our system.
   - Correctness of shared data is checked after particular period of time to check the data integrity.
   - Identity of signer in particular cases is shown by system.
   - More than one user is added dynamically and Access of one users files to another is given. Owner can decide which files which accesses to which users.

2) **Problems Solved Using Project:**
our system solves many problems some of them are as follows
   - Integrity of data checking is done with less space & time
   - We can add users afterwards dynamically and give them access of many files.
   - We have disclosure on personal data.
   - Insert, delete, update additionally upload & download operations on file are possible by all the users.
   - We dont need to recomputed whole signature if any small mistake takes place as we are using Hashes now.

3) **Social and Commercial Need or Aspect of project:**
   - The integrity threats of data are doubt & scrutiny so we need assured of data correctness.
   - There is easily loss or corrupt of data because of manual and hardware errors.
   - We use cloud services like Drop box, iCloude,

Google Drive very often.
- Cloud service providers unwilling to inform data errors to maintain reputation and avoid losing profits
- It is essential to check the integrity of data before data utilization such as search or computation over cloud data

4) **Advantages:**

- Our system is file centric & no group centric so dynamic users have access to all the files.
- We can have Static & Dynamic data.
- It focuses on personal & public data.
- Overall efficiency of project increased.
- It supports traceability as no ring signature.
- It makes freshness verification extremely efficient.
- It tempts nominal invisibility.
- Here in our system data blocks & their versions both are authenticated

5) **Features:**

- Public Analysing with traceability possible.
- Accuracy.
- Forge ability of data.
- Identity Privacy of data.
- Adaptable & tractable of system.
- Trusted Proxy of owner's data.
- Public auditing of Group users easy & lesser size due to Hash

6) **Set Theory:**

Let s (be a main set of) = {U, F, A, H, C, P, E, S, TPADB, SSDB} Where,
U is an infinite set of all users who can upload files to the server database and retrieves from the server. And (u1, u 2, u3..., un) U.
F is a set of all files which are uploaded to the server database and retrieved from the server.
And (f1, f2, f3, ..,fn) *s* F
A is a set of all access lists for particular files which are uploaded to the server database and retrieved from the server. And (a1, a2, a3, ...an) *s* F
H is a set of hashes for respective partitions of particular files which are partitioned on the server and retrieved from the server. And (h1, h2, h3, ...,hn) *s* H
C is a set of challenges sent and verified by TPA for respective partitions of particular files which are partitioned on the server and retrieved from the server. And (c1, c2, c3, ...,cn) *s* C
P is a set of partitions of particular files which are partitioned by user on the server before encryption and retrieved from the server.
And (p1, p2, p3, ..., pn) *s* P
E is an encryption algorithms applied on the input data to get encrypted results.
S is the storage server of the system. The server is responsible for registering, authenticating and providing associations to the end user.
TPADB is the copy of the server database. This database is responsible for storing partition information and hashes. This is basically database on TPA server.
SSDB is a set of local database on storage server. It consists of partitions of files and hashes computed by storage server.

7) **Functionalities :**

U = RegisterUser (uid, password, full name, address, country name, contact number, email address);
password = SHA1(input-password);
UA = AuthenticateUser (uid, password, U);
F = Select file (Memory);
P = Partition file (F, PSize);
Ep = Encrypt Partitions of file (P, K)..AES; H = Generate Hash for Partitions of file (Ep); A = Add users to Accesslist (F, u1, .., un);
TPADB = AddHashes(H);
SSDB = AddPartitions(Ep);
UPLOAD(F);
H = Apply Encryption(Ep);
Results = Verify(Download(H));

8) **System Flow:**

following figure 2 gives the flow of system

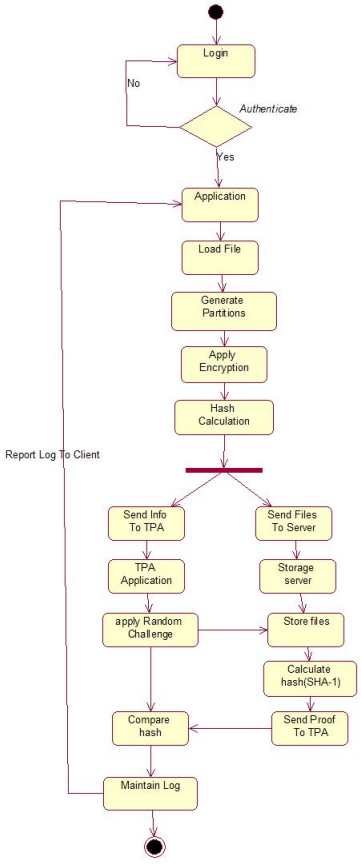

Fig. 2. Proposed System Flow.

9) **Algorithms used:**

1. AES Algorithm:

We used this algorithm to encrypt the partitions of file. AES is a new cryptographic algorithm that can be used to protect electronic data. AES is symmetric-key block cipher that uses keys of 128, 192, and 256 bits, and encrypts as well as decrypts data in blocks of 128 bits. Unlike public-key ciphers, which use keys, symmetric-key ciphers make use of the same key for encryption and decryption of data. Encrypted data which is returned by block ciphers should contain the same number of bits that the input data had. Iterative ciphers make use of a loop structure that repeatedly performs permutations as well as substitutions of the input data. Figure shows AES in action encrypting and then decrypting a 16-byte block of data using a 192-bit key.

2. SHA1 Algorithm:

We used this algorithm to have hashes of encrypted partitions. The SHA1 encryption algorithm specifies a Secure Hash Algorithm (SHA1), which can be used to generate a representation of a message known as a message digest. The SHA1 is used with the Digital Signature Algorithm (DSA) as noted in the Digital Signature Standard (DSS) and whenever a secure hash algorithm is needed. Both the transmitter and receiver of a message in computing and checking a digital signature use the SHA1.

C. Input,Output and Result:

- **Input:**
  Following are the inputs given to system.
  1. Files are given as input by users.
  2. These files are undergoing to partitions and these are encrypted afterwards.

- **Output:**
  Following are the outputs given by the system.
  1. Hash technique is used on encrypted data.
  2. Verification of data is done using hash created by TPA.

- **Result:** Following are the results observed by considering inputs given to the system and outputs given by the system
  1. To display the details through the logs and can be given result as successful in security checking or unsuccessful.

D. Hardware and Software Used

Hardware Configuration

- System : Pentium IV 2.4 GHz.
- Hard Disk: 40 GB.
- Monitor : 15 inch VGA Color
- Mouse : Logitech Mouse.
- RAM : 512 MB
- Keyboard : Standard Keyboard

Software Configuration

- Operating System : Windows XP.
- Coding Language : Java
- Database : My SQL

E. Modules

1) **Owner (User A):**
As mentioned, cloud data storage not only provides dynamic as well as scalable storage services, but also allows easy file sharing as demanded. A difficult problem is support for services with legacy users, who can access and can modify the owners data in the cloud. Under this multi-writer model, acquiring the same data dynamics support for public analyzing services while maintaining file consistency is another future challenge. Owners have many activities it does authentication i.e. successful login by providing user id and password. It also determines access policies and it can give access to other users.

2) **Other Users (Users N):**
As it has provided access by owner users it can upload download and update files.

3) **Storage Server:**
Using Hashing techniques helps achieve a constant communication overhead for public analyzing ability. However, the approach to support data dynamically may have security and efficiency problems. It stores files on database and it computes hash whenever TPA asks and send proofs to TPA.

4) **TPA:**
To fully ensure data security and save data resources, we propose to make it publicly analyzable cloud storage services, where data owners transfer this work to an external third party auditor (TPA) to check the outsourced data when needed. Third party analyzing provides a transparent and cost-effective method for establishing trust between data owner and server. Actually, based on result from a TPA, the sent audit report would help owners to evaluate the risk of their cloud data services, as well as be benefited for the cloud service provider to enhance their cloud based service platform. TPA stores hash and partition info and randomly sends challenges to storage server. When storage server sends proof it verifies proof and generates log.

IV. CONCLUSION & FUTURE WORK

We can conclude that by doing randomly partitions on file we are making secure and encryption adds security to it we encrypt the partitions of file so it should not be visible to everyone making it less private. We have used AES algorithm for encryption techniques. We used hashing technique by applying SHA1 algorithm to make it less in space and time to check random challenges so that integrity of data can be checked with particular time period. In this hashing it gives us small size hashes which are not easily recognized by all the users so though publically analyzing we are making it privacy

retaining throughout. In some special case unlike HARS we use SHA1 so owner can be identified though privacy preserved. We are publically analyzing more static as well as dynamic users and access of files are given to non-owner users by owners& process of dynamic data is possible by all users. Traceability is main key feature of our project which is possible though protecting private data.

ACKNOWLEDGMENT

The proposed system is based on IEEE Transaction paper under the title ”Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud” IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 2, NO. 1, JANUARY-MARCH 2014

REFERENCES

[1] B. Wang, B. Li, and H. Li, Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud, Proc. IEEE Fifth Intl Conf. Cloud Computing, pp. 295-302, 2012.

[2] 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, A View of Cloud Computing, Comm. ACM, vol. 53, no. 4, pp. 50-58, Apr. 2010.

[3] K. Ren, C. Wang, and Q. Wang, Security Challenges for the Public Cloud, IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.

[4] D. Song, E. Shi, I. Fischer, and U. Shankar, Cloud Data Protection for the Masses, Computer, vol. 45, no. 1, pp. 39-45, 2012.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing, Proc. IEEE INFOCOM, pp. 525-533, 2010.

[6] B. Wang, M. Li, S.S. Chow, and H. Li, Computing Encrypted Cloud Data Efficiently under Multiple Keys, Proc. IEEE Conf. Comm. and Network Security (CNS 13), pp. 90-99, 2013.

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

[8] The MD5 Message-Digest Algorithm (RFC1321). https://tools. ietf.org/html/rfc1321, 2014.

[9] 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. 598-610, 2007.

[10] H. Shacham and B. Waters, Compact Proofs of Retrievability, Proc. 14th Intl Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT 08), pp. 90- 107, 2008

[11] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, Dynamic Provable Data Possession, Proc. 16th ACM Conf. Computer and Comm. Security (CCS09), pp. 213-222, 2009.

[12] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, Enabling Public Verifia-bility and Data Dynamic for Storage Security in Cloud Computing, Proc. 14th European Conf. Research in Computer Security (ESORICS09), pp. 355-370, 2009.

[13] C. Wang, Q. Wang, K. Ren, and W. Lou, Ensuring Data Storage Security in Cloud Computing, Proc. 17th Intl Workshop Quality of Service (IWQoS09), pp. 1-9, 2009.

[14] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, Remote Data Checking for Network Coding-Based Distributed Storage Systems, Proc. ACM Workshop Cloud Computing Security Workshop (CCSW10), pp. 31-42, 2010.

[15] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S Yau, Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds, Proc. ACM Symp. Applied Computing (SAC11), pp. 1550-1557, 2011..

