ISSN 2348 – 9928
IJAICT Volume 5, Issue 11, November 2018 Doi:01.0401/ijaict.2015.02.02 Published on 05 (11) 2018

ENHANCED PRIVACY PRESERVING METADATA VERIFICATION BY ACCOMPLISHING TRACEABILITY FOR SHARED DATA IN CLOUD

Mr. J. Moses Pushparaj
PG Scholar,
Computer Science and Engineering,
Jeppiaar Engineering College,
Chennai, Tamilnadu, India.

Ms. K. Rekha
Assistant Professor,
Computer Science and Engineering,
Jeppiaar Engineering College,
Chennai, Tamilnadu, India.

*Abstract— **With cloud computing and data storage services, the data not only stored in the cloud but it can be shared among multiple users in the cloud. However the integrity of the shared cloud data protection is formidable task due to the hardware, software failures and human errors. For secure purpose to introduce an effective TPA to perform public auditing for the shared data is very importance, so that the users can be worry free of the outsourced data. While auditing the integrity of the shared data will inevitably reveal confidential information, identity privacy to the public verifiers (TPA). In this paper we propose a secure cloud storage system to ensure that the shared data integrity can be checked publicly, group users in the shared data need to generate signatures on all the blocks in shared data performed by different users. Especially, we utilize ring signatures to calculate the authentication information needed to audit the integrity of shared data. By using Public Auditing, the identity of the signer on each block in shared data is reserved private from a third party auditor (TPA), who is still able to verify the integrity of shared data without retrieving the entire file. Using Hash Based Verification, we can also provide traceability about the shared data.***

*Keywords— **Public Auditing, Privacy Preserving, Data Integrity, Shared Data, Cloud Computing, HBV, Traceability***

I. INTRODUCTION

With Cloud Computing data storage and sharing services in the cloud, users can easily able to modify and share data as a group offered by cloud service providers at a lower marginal cost. Data sharing becomes a standard feature in most of the cloud storage services including Google Drive, iCloud and Dropbox [1]. It is common for the users to think whether their data remain protected over long period of time due to the software, hardware failures and the human errors in an untrusted cloud server, the integrity of the shared data is still a compromised one. So it is important to protect the integrity of the shared data in the cloud and offer peaceful mind to users, the best way is to introduce a Third Party Auditor (TPA) [8] to perform auditing tasks on behalf of the data owner.

One of the traditional approach for checking the data correctness is to retrieve the entire data from the cloud server, and verify the integrity of the shared data by verifying the correctness of the signatures [e.g., RSA] [13]. or Hash values [e.g., MD5] [14]. This kind of approach can be able to successfully check the integrity of the shared data but the efficiency on cloud data is in doubt. The common reason for that approach is the size of the cloud data is generally large. To verify the shared data integrity the entire data has to be downloaded will cost users amounts of computation and communication resources.

PDP (provable Data Possession) [5],[6] that allows a public verifier to publicly auditing the integrity of the shared data without retrieving the entire data that are stored in the untrusted cloud environment. Most of the work focused on how a dynamic data, identity privacy and data privacy can be supported during the public auditing process. Moreover, most of the earlier work focusing only an auditing the integrity of the personal shared data in the cloud. Wang et al. [7], [8] recently designed a privacy-preserving auditing mechanisms for shared data in the cloud that the identity of the signer on each block in shared data is reserved private from a third party auditor (TPA) during the public auditing task. By preserving the identity privacy the Third Party Auditor cannot be able to find out which user in the group or which block in shared data having higher valuable target than others in the group.

Recent mechanisms information used for verification are computed with ring signatures finally leads to take much time to audit when the size of the information verification and the number of users are linearly increasing in the group. So while adding the new users in the group all the existing verification information need to be recomputed [12], [15]. Moreover the identities of signers on each block is unconditionally protected by using ring signatures which prevents the data owner to trace the identity of the signer on each block when any of the user in the group is misbehaved.

In this paper, we propose a new privacy preserving mechanism to publicly audit the integrity of the shared data that are stored in the untrusted cloud server and shared across multiple users in the group. For that we take an advantage of ring signatures to construct homomorphic authenticators, so that the Third party Auditor can be able to publicly audit the integrity of the shared data without retrieving the entire data from the cloud server and the public verifier cannot reveal the identities of signers on each block for the whole blocks of the data. Moreover, the size of the information verification and also the time taken to audit the data are not affected even the number of users are linearly increasing in the group.

In addition the data owner (Group Manager) can trace the user's signatures on shared data and reveal the identities of signers on each block when it is necessary. The data owner who creates the data and shares the data in the cloud that can be divided into multiple blocks, for each block hash function has to be set for the traceability the user who are using the shared data doesn't aware of the hash function. The TPA while sending the auditing details also send the hash computation values and the part of the private key of the users on every block to the data owner. The data owner can match the part of the private key with the original private key pair to trace the users of the group using Hash Based Verification (HBV), so that he able to know how many times the files can be accessed by the users , who are all accessed the files. Now Data owner can be able to track the users activity. Since our mechanism can support both public auditing mechanism and traceability still able to preserve the identity of signers on each block.

Table 1. Comparison among different Mechanisms and Hash Based Verification (HBV)

	PDP [18]	WWRL [19]	Oruta	HBV
Public Auditing	✓	✓	✓	✓
Data Privacy	×	✓	✓	✓
Identity Privacy	×	×	✓	✓
Traceability	×	×	×	✓

II. RELATED WORK

Ateniese et al. [5], [6] are the first to proposed public auditability in their Provable data possession (PDP) model to ensure a client to verify the integrity of shared data stored in an untrusted server storages without retrieving the entire file. For auditing the shared data they utilize the RSA homomorphic linear authenticators and suggested the random sampling of few blocks of the shared file. It can support static data lack in efficiency of verification. PDP using the symmetric keys unfortunately this protocol does not support privacy preserving and it may lead to leakage of user data information to the Third Party Auditor.

Juels and kaliski et al. [6] describing the another model proof of retrievability (POR), which is also able to spot checking and error-correcting codes by ensuring both possession and retrievability in an untrusted server. Public auditability is not supported by this mechanism because the number of auditing tasks is fixed for the users. This approach works with encrypted data because they describing a merkle tree construction for PORs. Shacham and Waters et al. [4] describing POR that are the improved mechanisms which are built on BLS signatures and Pseudo random functions.

Hao et al. [3] designed a RSA based dynamic public auditing mechanism. Wang et al. also designed a public auditing mechanism with dynamic data based on Merkle Hash Tree. Zhu et al. [11] constructed a Index Hash Table to support the dynamic data by ensuring the correctness of users data stored in the server.
Wang et al. [1], [2], [7], [8] considered public auditing with data privacy in the cloud; the TPA is able to check the integrity of shared data in the cloud but cannot obtain a confidential information of the shared data. Recently, Oruta representing the privacy preserving public auditing mechanism in the shared public cloud the public verifier is able to verify the integrity of the shared data but cannot reveal the identity of the signer on each block is still preserving the identity privacy. Unfortunately it does not support the traceability where the signer on each block is unconditionally protected.

III. PROBLEM STATEMENT

3.1 System Model

In this paper, we consider for the cloud data storage and sharing services with three entities such as the Cloud Server, group of users who participate as users. The group of users which includes original user he who creates the group called Group Manager, and the number of group users. Initially the original user is the owner of the data and shares the data with other users in the shared group. Based on the access control policies other group users can be able to access the data and also they can able modify, download the shared data. The Cloud Service Provider provides cloud data storage and sharing services for users and has the large storage space. The third party auditor such as the public verifier is able to verify the shared data integrity by getting the requests from the group manager, without downloading the entire data.
When a group manager wishes to check the data integrity of the shared data he first sends an auditing request to the TPA. By

receiving the auditing request the TPA sends an auditing challenge to the Cloud Server. The Cloud Server by receiving the auditing challenge from the TPA after that sends an auditing proof for the required shared data claimed by the TPA. Eventually the TPA sends an auditing report to the group manager based on the metadata verification.

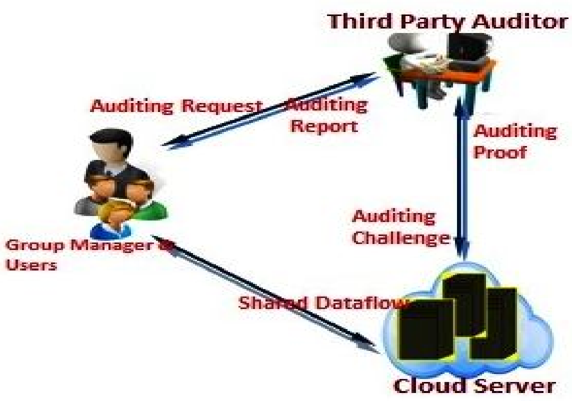


Fig 1: The System Architecture includes, Cloud, TPA, Group Manager and Users

3.2 Threat Model

Integrity Threats: Generally two kinds of threats are possible related to data integrity in the shared data. First, the external opponent may try to corrupt or attack the integrity of the shared data. Second, Due to the hardware and software failures even human errors the Cloud Service Provider may hide or even remove the data in its storage in order to avoid losing profits for their services, because they are economically motivated and they don't want lose their reputation.

Privacy Threats: The signer's identity on each block in the shared data is reserved private and confidential which group users identities do not disclosed to others. During the auditing process only the partially trusted third party auditor is allowed to verify the integrity of the shared data, might be trying to disclose the identity of signer on each block based on the verification metadata.

3.3 Design Goals

Our mechanism, HBV is designed to achieve following properties:

1) Public Auditing: The third party auditor is publicly verifying the data integrity without downloading the entire data from the untrusted cloud server.

2) Correctness: The Third party Auditor is correctly verify the shared data integrity.

3**) Unforgeability:** Only group user can be able to generate a valid verification metadata on shared data.

4) Identity Privacy: When an auditing task takes place the TPA cannot distinguish the signer's identity on each block in shared data.

5) Traceability: The group manager is able track the users activity from accessing the data from the cloud.

IV. PRELIMINARIES

4.1 Ring Signatures

The concept of ring signatures was first proposed by Rivest et al. in 2001 [16]. Using ring signatures, the third party auditor is convinced that the metadata (i.e., signature) is generated using any one of the group users private keys, but the public verifier cannot determine the users identity. Moreover, given a ring signature and group users d, the third party auditor cannot distinguish the signers identity with a probability of more than 1/d. This scheme is introduced by the Boneh et al. constructed on bilinear maps.

4.2 Simple Object Access Protocol (SOAP)

SOAP was designed by Don Box, Dave Winer, Mohsen Al-Ghosein and Bob Atkinson as an object-access protocol in 1998 for Microsoft. SOAP is an XML-based messaging protocol for accessing web services. It is very important for application development programs to allow Internet communication between programs. Today's most of the applications communicate using Remote Procedure Calls (RPC) between objects DCOM and CORBA, but HTTP was not designed for this kind of applications. RPC representing a compatibility and security problems, firewalls and proxy servers usually block this kind of traffic. One of the better way to communicate between applications over HTTP is supported by almost all the internet browsers as well as the servers, in which it is been created to complete this task. SOAP provides a communication between applications running on different types of operating systems, different kind of technologies and variety of programming languages. SOAP message is an ordinary XML document containing the following elements.

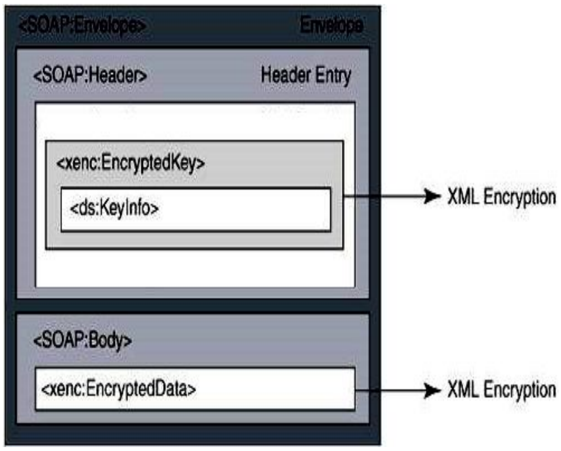


Fig 2 : SOAP Message Format

Envelop: The required SOAP Envelope element is the root element of a SOAP message. This element defining the XML document contains a SOAP message.

Header: SOAP Header element contains application-specific information (e.g., authentication, transaction, etc) about the SOAP message. When the Header element is available, then it should be the first child element of the Envelope.

Body: The required SOAP Body element contains the actual SOAP message intended for the ultimate endpoint of the message.

V. PROPOSED SYSTEM

By ensuring the data integrity of the shared data in the cloud and reduce the online burden of the cloud users, computation resources enabling public auditing mechanism for the cloud shared data is of very importance so that users might be worry free for their cloud data. Enabling Third Party Auditor to publicly audit the outsourced shared data in the cloud when needed by the group manager that who had best capabilities and expertise in the field that users in the cloud do not have the experiences of auditing tasks. The Third Party Auditor periodically checks the integrity of the shared data that are stored in the cloud on behalf of the group manager which provides the assurance for the data correctness in the users data. It also provides the identity privacy of the users in the group when public verifier auditing the data. Traceability can be done with public auditing by the group manager through the public verifier using Hash Based Verification. Moreover to say that enabling public auditing mechanisms is an important role in the cloud environment and users always assess risk and gain trust in the shared public cloud.
Our HBV scheme consists of seven algorithms.

KeyGen, Join, SigGen, ProofGen, ProofVerify, TraceOpen

- **KeyGen:** Key generation can be done by the group manager to set up the mechanism that he who creates his private key and group public key.
- **Join:** The group manager can be able to generate a private key for a new group user and add this new user to the group user list.
- **SigGen:** The group user can generate sign on shared data by using his private key and group members public key.
- **Modify:** By using user private key and group members public key the user can be able to modify and update the shared data.
- **ProofGen:** It can be performed by the Third Party Auditor and the cloud server to interactively generate a proof of possession of shared data.
- **ProofVerify:** The Third Party Auditor is able to verify the integrity of the data by using the aggregate group public key, but he cannot distinguish the identity of the signer on each block.
- **TraceOpen:** The group manager is able to reveal the identity of the signer on each block using hash based verification to track the users activity.

V. METADATA VERIFICATION SCHEME

A public auditing mechanism is important in cloud environments because huge volumes of data have been updated frequently must audited with the efficient public auditing schemes.

6.1 Security Model

The security model is designed to check the data integrity by using the public auditing mechanism. This kind of security scheme verifies the metadata instead of verifying the actual data. The security model is categorized into two blocks

1) Metadata Generation
2) Metadata Verification

6.2 Metadata Generation

Initially the group has been created by the group manager. He who creates the data and encrypt the data with his own private key and common group public key. Before uploading the data into cloud the file is divided into multiple blocks. Each block having the metadata (i.e., signatures) frequently signed by different users. One more thing that same prime number has been set for the whole data blocks for hash based verification to achieve traceability. After all above things this data blocks has been migrated into cloud.

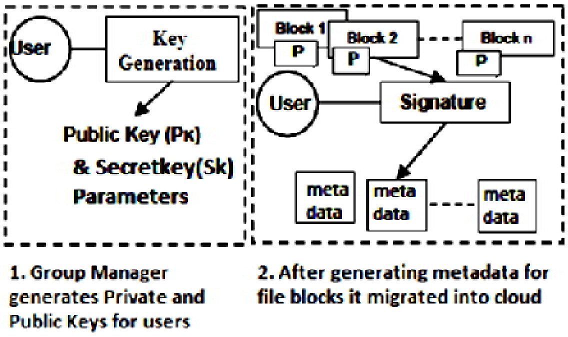

User
Key Generation
Public Key (Pκ)
& Secretkey(Sk)
Parameters
Block 1
P
Block 2
P
Block n
P
User
Signature
meta data
meta data
meta data
1. Group Manager generates Private and Public Keys for users
2. After generating metadata for file blocks it migrated into cloud

Fig : 3 Metadata Generation

6.3 Metadata Verification

Once the data along with signatures has been uploaded into the cloud the third party auditor can perform verification scheme at any requested time by the group manager. If the user access the block, automatically the sign key generated based on the private key order. Then the cloud will send the sign key with private key order to the TPA. While receiving the proof from the Cloud the TPA will verify the proof based on the sign key with public key order given by the group manager. If both are equal then the integrity of the data is fine. Otherwise TPA will send the fail report to the group manager.

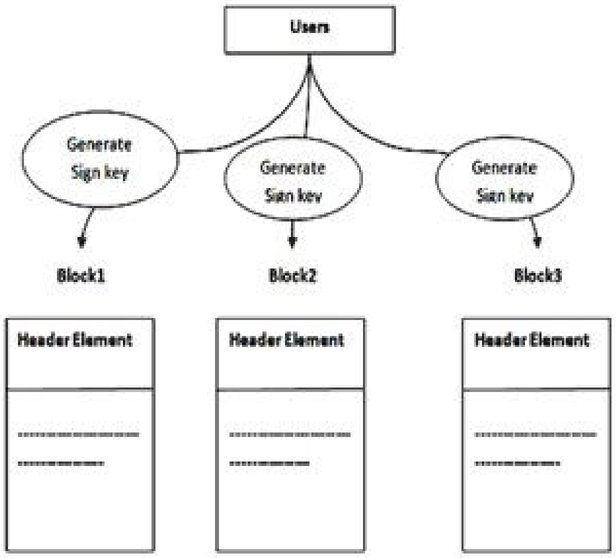


Fig 4 : Sign key Generated by the users by accessing the data blocks

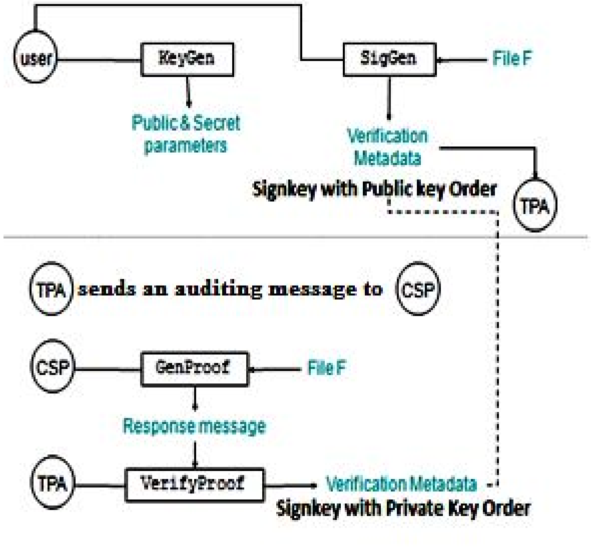


Fig 5. Metadata Verification

6.4 Hash Based Verification (HBV)

As mentioned earlier each and every block will be having the same prime number (P) for entire file. The group of users doesn't aware of the prime number. When users are accessing the data blocks HBV takes place.

Step 1 **: Randomly picks a prime number p.**
Step 2 **: Compute Hash value (H) for prime number p.**
Step 3 **: Store the hash value in public cloud.**
Step 4 **: If user access the shared data the hash value should perform either addition or multiplication operation.**
Step 5 **: Based on the user's access repeat step 4.**
Step 6 **: Finally, the operations are forwarded to the data owner.**
Step 7 **: Traceability computation involves reverse process of the hash value computation by using Subtraction or division operation.**
Step 8 **: Repeat step 7 until finds the generated hash value.**
Step 9 **: Based on the count value, traceability of the user should be predicted.**

Algorithm: Hash Based Verification

6.5 Assumptions and Steps

Let,

gP_k the group public key, **gS_k** group manager private key for encrypting the file blocks.

M_d be the signature generated for each block called as metadata.

F be the actual file to be verified by the TPA.

B be the single data block.

P be the prime Number for the hash based verification.

Step 1:

The File F can be divided into multiple data blocks is represented as,

$$\mathbf{F \rightarrow \sum (B_1+B_2+B_3+\ldots.+B_n)}$$

Step 2:

Prime Number has been randomly picked for the entire file and assigned the same to each and every blocks of the file for the HBV

$$\mathbf{F\rightarrow\sum(B_1\ P_1+\ B_2\ P_2\ +\ B_3\ P_3\ +\ldots.+B_n\ P_n)}$$

Step 3:

Signature (metadata) generated for each and every blocks of the file

$\mathbf{B_1\ P_1,\ B_2\ P_2\ ,\ B_3\ P_3,\ldots.B_n\ P_n}$

$\mathbf{\xrightarrow{gP_k\ gS_k}\ M_1,M_2,M_3,\ldots..,M_n}$

Step 4:

Data blocks are migrated into cloud along with metadata

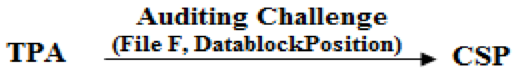
$\mathbf{B_1\ P_1,\ B_2\ P_2\ ,\ B_3\ P_3,\ldots.B_n\ P_n}$

+ **CSP** ⟹

$\mathbf{M_1,\ M_2,\ M_3,\ldots..,\ M_n}$

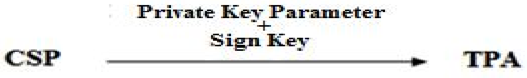
Step 5:

TPA sends an auditing challenge to the Cloud Server



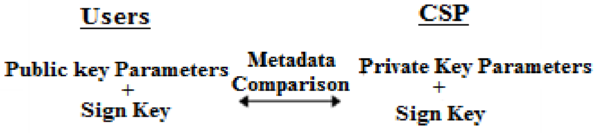
Step 6:

While receiving the auditing challenge from the TPA, the cloud server will send the auditing proof along with sign key and private key order to the TPA



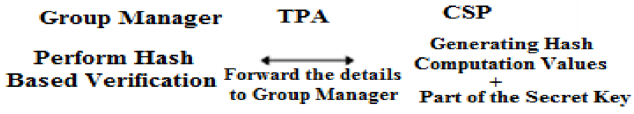
Step 7:

TPA receiving the Auditing proof from the CSP, the verify the data integrity.



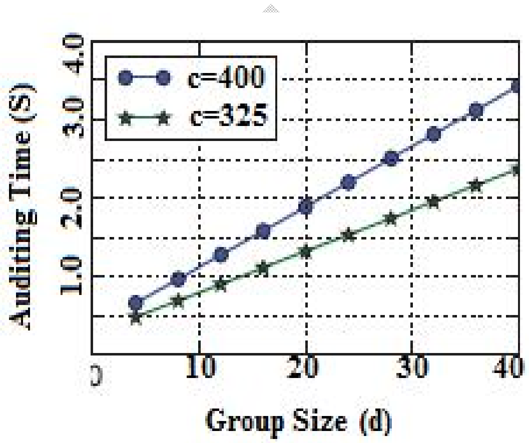
Step 8:

Group manager verifies the users activity for Traceability**.**

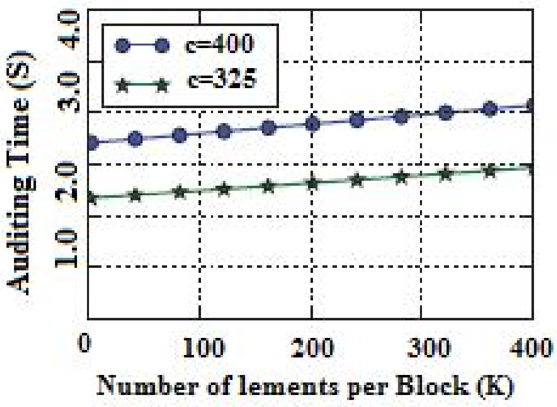


6.6 Experimental Results

Now we evaluate the performance of HBV with experimental results. We assume that the total number of blocks in the shared data. The auditing time is increasing linearly with the size of the group (d). So when C=400, when there are multiple users shares the data in the cloud, then the auditing time is about to take 0:5 seconds, at the same time when the group members number increases to 30, it will take around 3:0 seconds to complete the auditing task.



(a)Impact of Group Size (d) on Auditing Time (s)



(b) Impact of Auditing Time on k, when d=20

Fig 6 : Performance of Auditing

Comparing with the size of whole shared data, the computation cost that a Third Party Auditor consumes for an auditing task is small. For maintaining a higher detection probability, the Third party Auditor consumes greater computation and communication cost to complete the auditing task. When c=325, it taking 1:94 seconds for the third party auditor to audit the integrity of shared data, where the size of data is about 2 GB. When c=400, a third party auditor takes 2:34 seconds to check the integrity of the shared data.

ISSN 2348 – 9928
Doi:01.0401/ijaict.2015.02.02 Published on 05 (11) 2018

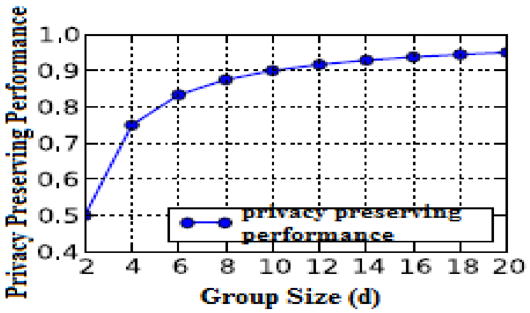


Fig 7 : Impact of d on privacy preserving Performance

The privacy preserving performance of our HBV mechanism is depending on the number of group members in the shared group. Consider a block in shared data, the most probability of a third party auditor fails to disclose the identities signer is 1 -1=d, where d>=2. More specifically, when the group member is greater, our HBV scheme has a much better privacy preserving performance.

Table 2. Privacy Preserving Performance of Auditing

System parameters	K=100, d=20	
Storage Usage	2GB + 250 MB (Data + Signatures)	
Selected Blocks	400	325
Auditing Time (s)	2.34	1.94

VII. CONCLUSION & FUTURE ENHANCEMENTS

In this paper, we propose a privacy preserving public auditing mechanism for shared data in the cloud using Hash Based Verification. For that we utilize ring signatures to compute verification meta data on shared meta data, so that the Third Party Auditor cannot reveal the identity of the signer on each block in the shared data. The group manager using his private key, new users can be added to group, and he can only disclose the identity of the signer on each block based on the Hash Based Verification on the shared data in the cloud. We will continue our work in future using complex polynomial construction in Hash Based Verification method we can improve the traceability of the shared data.

References

[1] CongWang ; Chow,S.S.M. ; QianWang ; KuiRen ;Wenjing Lou "Privacy_Preserving Public Auditing for Secure CloudStorage", IEEE Transactions on Computers Volume: 62 , Issue: 2 2013 ,PP no : 362 – 375

[2] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy- Preserving Public Auditing for Storage Security in CloudComputing," Proc. IEEE INFOCOM '10, Mar. 2010

[3] Hao, Z., Zhong, S., Yu, N.: A Privacy-Preserving Remote Data Integrity Check- ing Protocol with Data Dynamics and Public Verifiability. IEEE Transactions on Knowledge and Data Engineering 23(9), 1432–1437 (September 2011)

[4] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (Asiacrypt), vol. 5350, pp. 90-107, Dec. 2008.

[5] Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., Song,D.: Provable Data Possession at Untrusted Stores. In: Proc. ACM CCS. pp. 598–610 (2007)

[6] Ateniese, G., Pietro, R.D., Mancini, L.V., Tsudik, G.: Scalable and Efficient Prov-able Data Possession. In: Proc. ICST SecureComm. pp. 1–10 (2008)

[7] Wang, C., Wang, Q., Ren, K., Lou, W.: Ensuring Data Storage Security in Cloud Computing. In: Proc. IEEE IWQoS. pp. 1–9 (2009)

[8] Wang, C., Wang, Q., Ren, K., Lou, W.: Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing. In: Proc. IEEE INFOCOM. pp. 525– 533 (2010)

[9] 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.

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

[11] Zhu, Y., Hu, H., Ahn, G.J., Yau, S.S.: Efficient Audit Service Outsourcing for DataIntegrity in Clouds. Journal of System and Software 85(5), 1083–1095 (May 2012).

[12] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel and DistributedSystems, vol. 22, no. 5, pp. 847-859, May 2011.

[14] Chaum, D., van Heyst, E.: Group Signatures. In: Proc. EUROCRYPT. pp. 257–265. Springer-Verlag (1991)

[15] Ferrara, A.L., Green, M., Hohenberger, S., Pedersen, M.Ø.: Practical Short Signature Batch Verification. In: Proc. CT-RSA. pp. 309–324. Springer-Verlag (2009)

[16] C. Wang, Q. Wang, K. Ren, and W. Lou, "Towards Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Service Computing, vol. 5, no. 2, 220-232, Apr.-June 2012.

[17] R.L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'01), pp. 552-565, 2001.

[18] 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.

[19] 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.

[20] B. Wang, B. Li, and H. Li, "Panda: Public for Shared Data with Efficient User Revocation in the Cloud," IEEE Trans. Services Computing, 20 Dec. 2013, DOI:10.1109/TSC.2013.2295611

[21] B. Wang, B. Li, and H. Li, "Knox: Privacy- Preserving Auditing for Shared Data with Large Groups in the Cloud," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS'12),pp. 507-525, June 2012.