International Journal of
Advanced Technology and Innovative Research

www.ijatir.org

ISSN 2348–2370
Vol.08,Issue.10,
August-2016,
Pages:1939-1945

Protection of Shared Data using Auditing in Public Cloud

NALLAMOTHU GOPI[1], RACHA REVATHI[2], NITTALA SWAPNA SUHASINI[3]
[1]PG Scholar, Dept of CSE, Abdulkalam Institue of Technology and Science, Khammam, TS, India,
E-mail: nallamothugopil8@gmail.com.
[2]Assistant Professor, Dept of CSE, Abdulkalam Institue of Technology and Science, Khammam, TS, India,
E-mail: racharevathi540@gmail.com.
[3]Associate Professor & HOD, Dept of CSE, Abdulkalam Institue of Technology and Science, Khammam, TS, India,
E-mail: nittala_swapna@yahoo.com.

Abstract: In certain, we take advantage of ring signatures to compute the verification understanding wanted to audit the integrity of shared competencies. With our mechanism, the identification of the signer on every block in shared potential is stored individual from a third occasion auditor (TPA), who stays to be competent to publicly affirm the integrity of shared understanding without retrieving the whole file. Our experimental results show the effectiveness and efficiency of our proposed mechanism when auditing shared advantage. Load balancing makes cloud computing additional efficient and improves patron pleasure. This text introduces a higher load balance mannequin for the general public cloud centered on the cloud partitioning thought with a switch mechanism to opt for unusual approaches for exceptional situations. The algorithm applies the game thought to the burden balancing technique to support the efficiency within the public cloud environment.

Keywords: Load Balancing Model, Public Cloud, Cloud Partition, Game Theory.

I. INTRODUCTION

Cloud concept is nothing but the storage service, but it can also share across multiple users. we firstly prioritizes privacy preserving mechanism because while auditing data from cloud services it’s not a secured while that private information is publicly protected by cloud service. Specifically, the group signature scheme enables users to anonymously use the cloud resources, and the dynamic broadcast encryption technique allows data owners to securely share their data files with others including new joining users which protects the confidentiality from the revoked users in the dynamic broadcast encryption scheme. We propose that while any user is accessing the data from cloud it must be secured by unauthorized person or hacker. Cloud is un-trusted file storage, so we utilize encryption based access control for sharing document in the cloud storage service. User’s data is encrypted by using cryptographic technique because unauthorized person can hack the user’s private data. In this cryptographic technique we uses different algorithms like signature algorithm, key generation algorithm, ring verify algorithm, etc. these algorithms are used in the cryptographic technique. Users can enjoy high-quality services by migrating local data management systems into cloud servers. The main reason is that the size of cloud data is large in general. Downloading the entire cloud data to verify data integrity will cost or even waste users amounts of computation and communication resources, especially when data have been corrupted in the cloud. Besides, many uses of cloud data (e.g., data mining and machine learning) do not necessarily need users to download the entire cloud data to local devices [2]. It is because cloud providers, such as Amazon, can offer users computation services directly on large-scale data that already existed in the cloud.

II. HOMOMORPHIC AUTHENTICABLE RING SIGNATURES

In this part, we introduce a company new ring signature scheme, which is suitable for public auditing. Then, we will exhibit construct the privateness-preserving public auditing mechanism for shared competencies in the cloud centered on this new ring signature scheme in the subsequent section. As we offered in previous sections, we intend to make use of ring signatures to cover the identification of the signer on every block, in order that exclusive and touchy knowledge of the workforce simply is not disclosed to the TPA. However, common ring signatures cannot be straight used into public auditing mechanisms, because these ring signature schemes do not aid block much less verification. Without block less verification, the TPA has to down load the whole data file to confirm the correctness of shared knowledge, which consumes immoderate bandwidth and takes prolonged verification occasions. For that reason, we first bring together a brand new homo morphic authenticable ring signature (HARS) scheme, which is accelerated from a typical ring signature scheme, denoted as BGLS. The ring signatures generated via making use of HARS is competent now not most effective to maintain identification privateness however moreover to support block less verification. Using HARS and its properties we headquartered within the prior part, we now assemble Oruta, our privateness retaining public auditing mechanism for shared talents inside the cloud. With Oruta, the TPA can confirm the integrity of shared knowledge for a group of users without retrieving the entire information. In the meantime, the identification of the signer on each and every block in shared data is saved amazing from the TPA in the course of the auditing. To permit every individual inside the staff to conveniently adjust advantage within the cloud and share the brand new-day variant of advantage with the leisure of the crew, Oruta have got to additionally aid dynamic operations on shared knowledge. How to preserve the users Identity attributes from the TPA because the TPA is un trusted server If the TPA gets hacked by hacker then it may be leakage the users private information so we gave the protection to the server, while it get hack then it will give notification to the user ready to another new user. And again TPA will get ready to another new user.

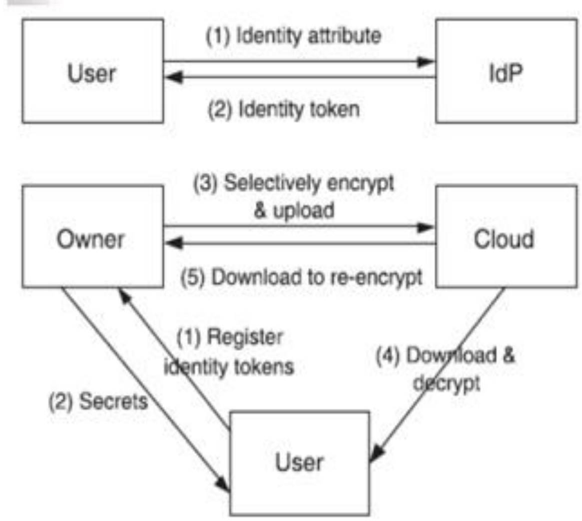


Fig.1. Overall System Architecture

An dynamic operation entails an insert, delete or replace operation on a single block. Nonetheless, when you consider that the computation of a ring signature entails an identifier of a block (as supplied in HARS), typical approaches, which first-rate use the index of a block as its identifier, are typically no longer suitable for helping dynamic operations on shared knowledge. The intent is that, when a consumer modifies a single block in shared advantage by way of performing an insert or delete operation, the indices of blocks that after the modified block are all transformed and the differences of those indices require customers to re-compute the signatures of those blocks, despite the fact that the content material of those blocks will not be modified. Previous than the normal consumer outsources shared knowledge to the cloud, she decides all the team contributors, and computes all of the preliminary ring signatures of the complete blocks in shared knowledge with her private key and all of the crew members' public keys.

III. SYSTEM MODEL

As illustrated in Fig2, the system model in this paper involves three parties: the cloud server, a group of users and a public verifier. There are two types of users in a group: the original user and a number of group users. The original ser initially creates shared data in the cloud, and shares it with group users. Both the original user and group users re members of the group. Every member of the group is allowed to access and modify shared data. Shared data and its verification metadata (i.e., signatures) are both stored in the cloud server. A public verifier, such as a third party auditor providing expert data auditing services or a data user outside the group intending to utilize shared data, is able to publicly verify the integrity of shared data stored in the cloud server. When a public verifier wishes to check the integrity of shared data, it first sends an auditing challenge to the cloud server. After receiving the auditing challenge, the cloud server responds to the public verifier with an auditing proof of the possession of shared data. Then, this public verifier checks the correctness of the entire data by verifying the correctness of the auditing proof. Essentially, the process of public auditing is a challenge and- response protocol between a public verifier and the cloud server.

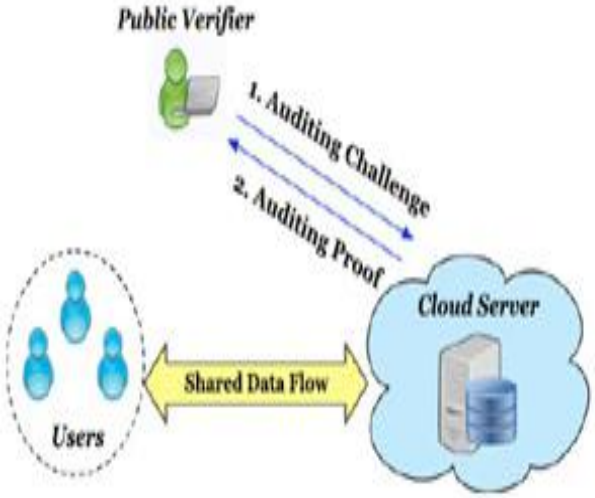


Fig.2. Our System Model Includes the Cloud Server, a Group of Users and a Public Verifier.

A. Threat Model

Integrity threats. Two kinds of threats related to the integrity of shared data are possible. First, an adversary may try to corrupt the integrity of shared data. Second, the cloud service provider may inadvertently corrupt (or even remove) data in its storage due to hardware failures and human errors. Making matters worse, the cloud service provider is economically motivated, which means it may be reluctant to inform users about such corruption of data in order to save its reputation and avoid losing profits of its services. Privacy threats. The identity of the signer on each block in shared data is private and confidential to the group. During the process of auditing, a public verifier, who is only allowed to verify the correctness of shared data integrity, may try to reveal the identity of the signer on each block in shared data based on verification metadata. Once the public verifier reveals the identity of the signer on each block, it can easily distinguish a high-value target (a particular user in the group or a special block in shared data) from others.

IV. SYSTEM ARCHITECTURE

A. Modern Ring Signature Scheme

Overview: The main motto of ring signatures [2] [3] is to hide the identity of the signer on each block in order to keep private and sensitive information un-disclosed to public verifier. However, the traditional ring signatures does not support block less verifiability and so the verifier needs to download the entire data from the cloud to check the correctness of the shared data which in turn consumes more bandwidth and more time. Therefore, it designs a new homomorphic authenticable ring signature (HARS) scheme, which is extended from classic ring signature scheme. HARS generated ring signatures are not only able to preserve identity privacy but are also able to support block less verifiability.

1. Construction of HARS

The HARS contains three algorithms: KeyGen, RingSign and RingVerify. In KeyGen algorithm each user in the group generates his/her public key and private key. In RingSign algorithm a user in the group is able to generate a signature on a block and its block identifier with his/her private key and all the group members' public keys. A block identifier is a string; it distinguishes the corresponding block from others. A verifier can check whether a given block is signed by a group member in RingVerify.

B. Public Auditing Mechanism

Overview: Using HARS and its properties, a privacy-preserving public auditing mechanism for shared data on cloud is constructed. In this scheme, the public verifier can verify the integrity of shared data without retrieving the entire data.kept private from the public verifier during the auditing.

1. Reduce Signature Storage

Another important issue need to consider in the construction of this scheme is the size of storage used for ring signatures. By the taxonomy of the ring signatures in HARS, a block m is an element of Zp and its ring signature contains d elements of G1, where G1 is a cyclic group with order p. It means a |p|-bit block requires a d * |p| -bit ring signature, which forces users to spend a huge amount of space on storing ring signatures. It will be very frustrating for users, because cloud service providers such as Amazon, will charge users based on the storage space they use. To reduce the storage of ring signatures on shared data and still allow the public verifier to audit shared data efficiently, we exploit an aggregated approach to expand the size of each block in shared data into k *|p| bits. With the aggregation of a block, the length of a ring signature. is only d/k of the length of a block. Generally, to obtain a smaller size of a ring signature than the size of a block, it choose k > d. As a trade-off, the communication cost of an auditing task will be increasing with an increase of k.

2. Support Dynamic Operations

To enable each user in the group to easily modify data in the cloud, there is a need to support dynamic operations on shared data. Dynamic operation such as insert, delete or update operation are performed on a single block. Since the computation of a ring signature includes an identifier of a block, traditional methods which only use the index of a block as its identifier are not suitable for supporting dynamic operations on shared data efficiently. When a user modifies a single block in shared data by performing an insert or delete operation, the indices of blocks are changed after the block modification and the changes of these indices require users, who are sharing the data, to re-compute the signatures of these blocks, even though the content of these blocks are not modified. This mechanism can allow a user to efficiently perform a dynamic operation on a single block, and avoid the re-computation of indices on other blocks.

3. Batch Auditing

Sometimes, a public verifier may need to verify the correctness of multiple auditing tasks in a very short time. Directly verifying these multiple auditing tasks separately would be inefficient. By leveraging the properties of bilinear maps, the concept of batch auditing can be supported, which can verify the correctness of multiple auditing tasks simultaneously and improve the efficiency of public auditing.

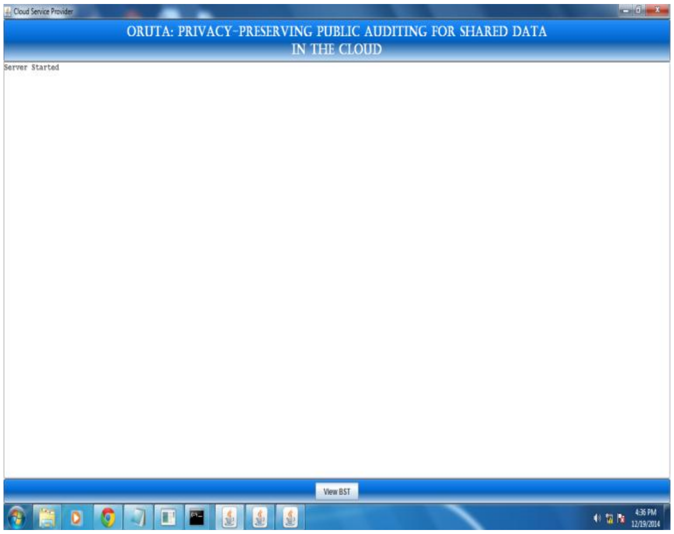
4. Ring Signatures

The concept of ring signatures was first proposed by Rivest et al. [8] in 2001. With ring signatures, a verifier is convinced that a signature is computed using one of group members' private keys, but the verifier is not able to determine which one. More concretely, given a ring signature and a group of d users, a verifier cannot distinguish the signer's identity with a probability more than 1=d. This property can be used to preserve the identity of the signer from a verifier. The ring signature scheme introduced by Boneh et al. [2] (referred to as BGLS in this paper) is constructed on bilinear maps. We will extend this ring signature scheme to construct our public auditing mechanism.

5. Construction of Oruta

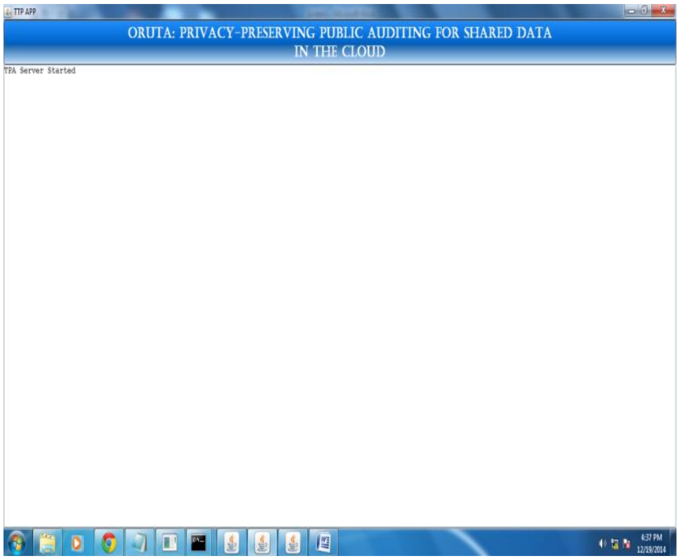
Now, we present the details of our public auditing mechanism. It includes five algorithms: KeyGen, SigGen, odify, ProofGen and ProofVerify. In KeyGen, users generate their own public/private key pairs. In SigGen, a user (either the original user or a group user) is able to compute ring signatures on blocks in shared data by using its own private key and all the group members' public keys. Each user in the group is able to perform an insert, delete or update operation on a block, and compute the new ring signature on this new block in Modify. Proof Gen is operated by a public verifier and the cloud server together to interactively generate a proof of possession of shared data. In ProofVerify, the public verifier audits the integrity of shared data by verifying the proof. Note that for the ease of understanding, we first assume the group is static, which means the group is pre-defined before shared data is created in the cloud and the membership of the group is not changed during data sharing. Specifically, before the original user outsources shared data to the cloud, he/she decides all the group members.

V. RESULTS



```
Server Started
```

Fig.2. TPA Screen: Start the TPA Server



```
TPA Server Started
```

Fig.3. Client Screen: Start the Client Application

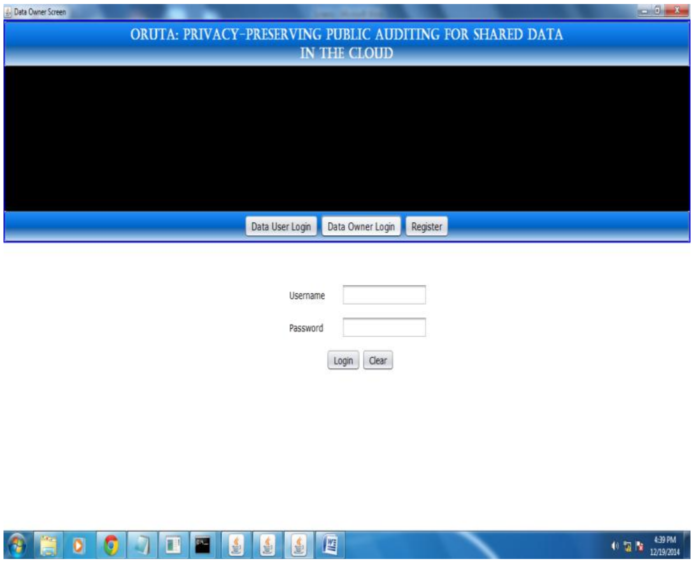
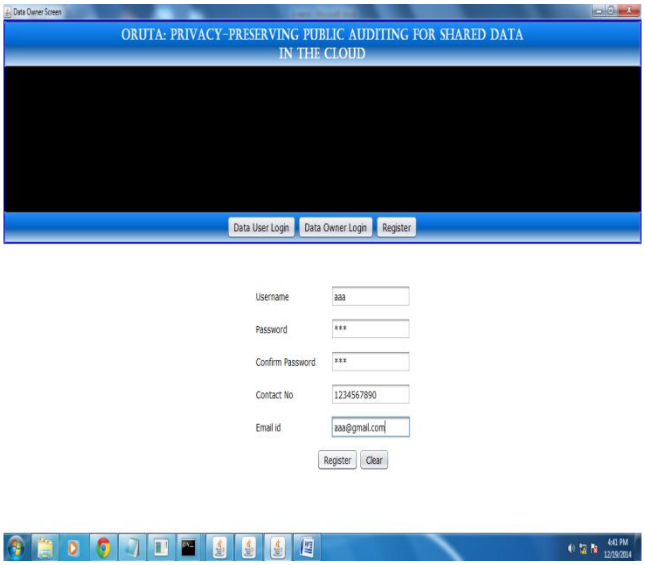
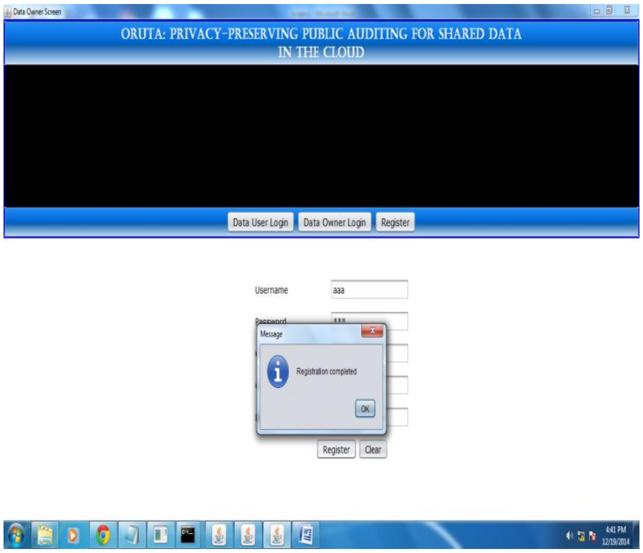


Fig.4. Registration Screen.



Username aaa
Password ***
Confirm Password ***
Contact No 1234567890
Email id aaa@gmail.com

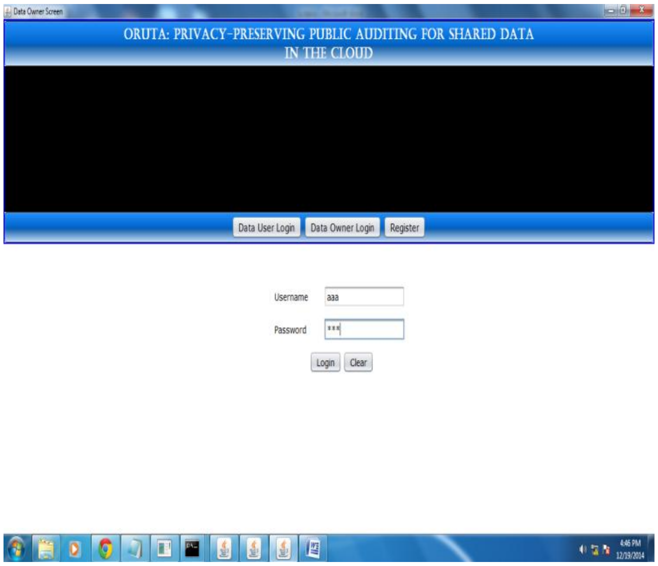
Fig.5.



Username aaa
Password ***

Registration completed

Fig.6. After Registration Successful.



Username aaa
Password ***

Fig.7. Data Owner Login

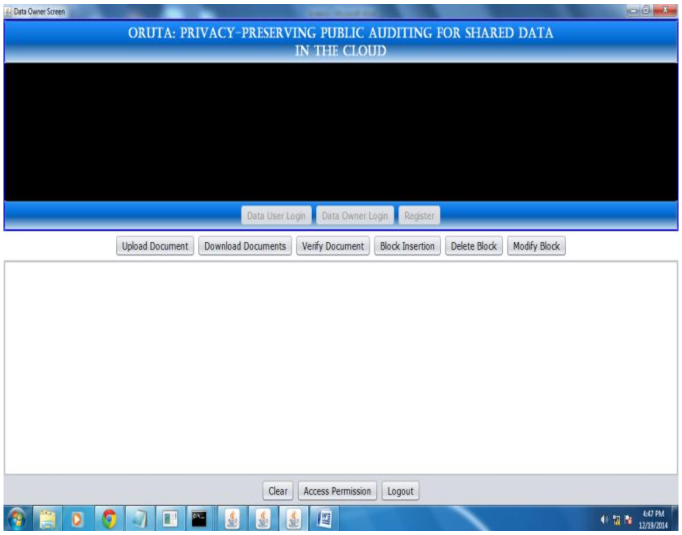
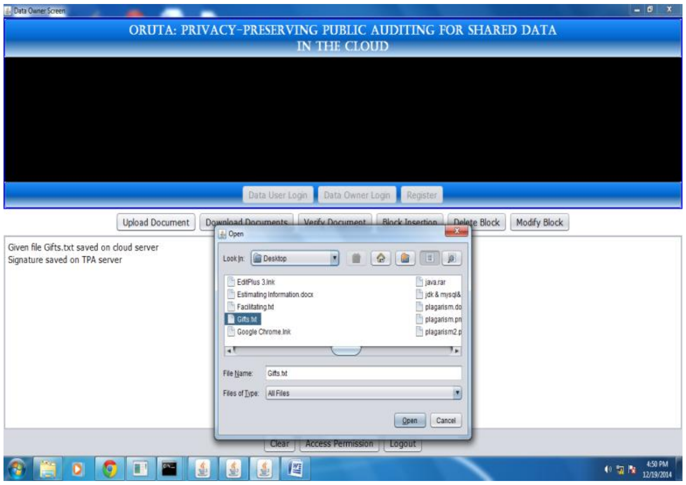


Fig.8. Owner Screen.



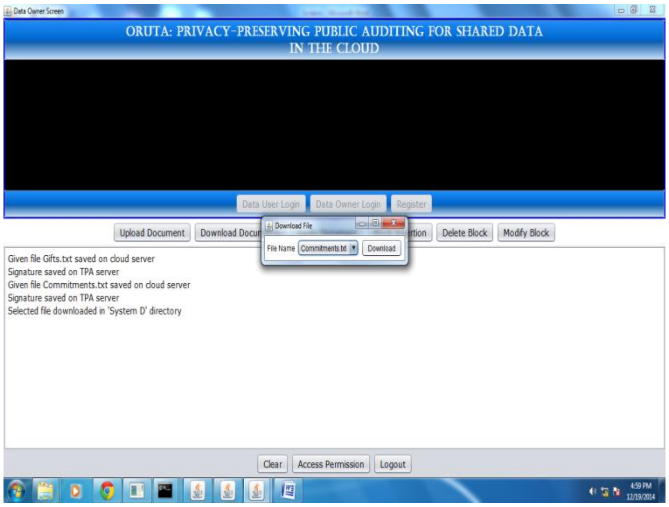
```
Given file Gifts.txt saved on cloud server
Signature saved on TPA server
```

File Name: Gifts.txt

Files of Type: All Files

Fig.9. Uploading Screen.

After uploading the document, it will be divided into blocks and stored into **"D:\Cloud"** folder as encrypted data. And, the signatures are stored into the database as well at the TPA server.

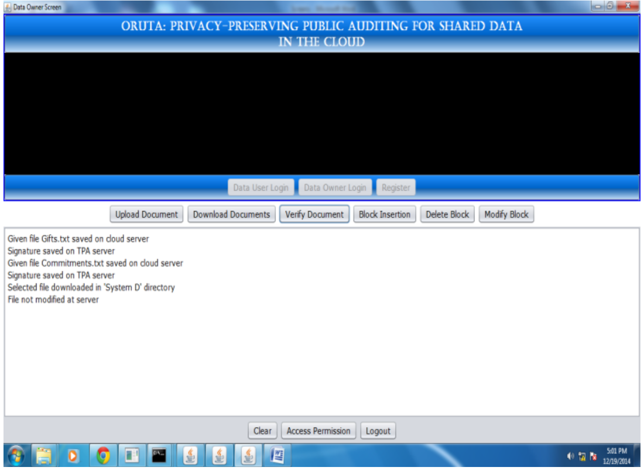


```
Given file Gifts.txt saved on cloud server
Signature saved on TPA server
Given file Commitments.txt saved on cloud server
Signature saved on TPA server
Selected file downloaded in 'System D' directory
```

File Name: Commitments.txt

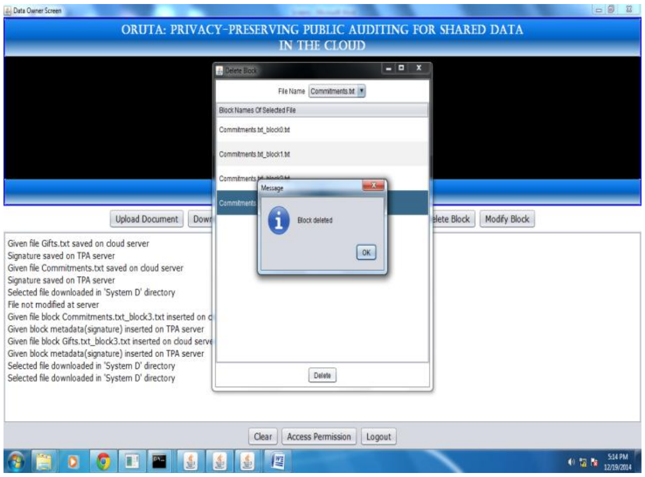
Fig.10. Download File Screen: It will be downloaded into "D-drive".

Verify Document Screen: To verify the document click on Verify Document and it will show message (Note: In this application for verifying the Signature we used the "ECKey" Algorithm which are given by Google API (my-wallet-bitcoinj-0.6.1.jar)) will be internally checking Original signature, data signature and public key.



```
Given file Gifts.txt saved on cloud server
Signature saved on TPA server
Given file Commitments.txt saved on cloud server
Signature saved on TPA server
Selected file downloaded in 'System D' directory
File not modified at server
```

Fig.11.



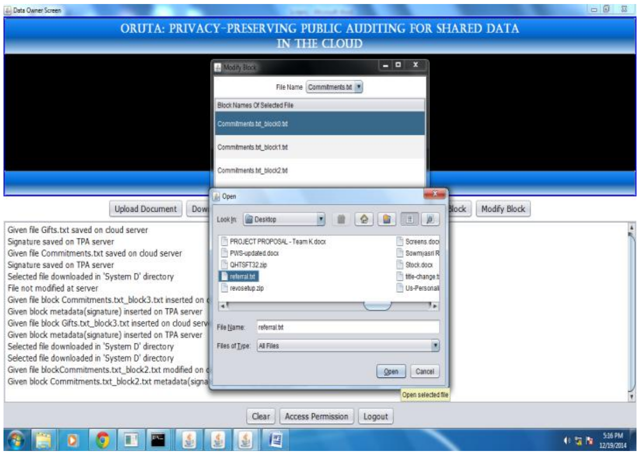
```
Given file Gifts.txt saved on cloud server
Signature saved on TPA server
Given file Commitments.txt saved on cloud server
Signature saved on TPA server
Selected file downloaded in 'System D' directory
File not modified at server
Given file block Commitments.txt_block3.txt inserted on c
Given block metadata(signature) inserted on TPA server
Given file block Gifts.txt_block3.txt inserted on cloud serve
Given block metadata(signature) inserted on TPA server
Selected file downloaded in 'System D' directory
Selected file downloaded in 'System D' directory
```

Block deleted

Data Owner Screen
ORUTA: PRIVACY-PRESERVING PUBLIC AUDITING FOR SHARED DATA IN THE CLOUD
Delete Block
File Name Commitments.txt
Block Names Of Selected File
Commitments.txt_block0.txt
Commitments.txt_block1.txt
Commitments.txt_block2.txt
Commitments
Message
OK
Delete
Upload Document
Dow
lete Block
Modify Block
Clear
Access Permission
Logout
5:14 PM
12/19/2014

Fig.12. Block Insertion Screen.

Modify Block: To modify the block select the block and insert new document to overwrite that block

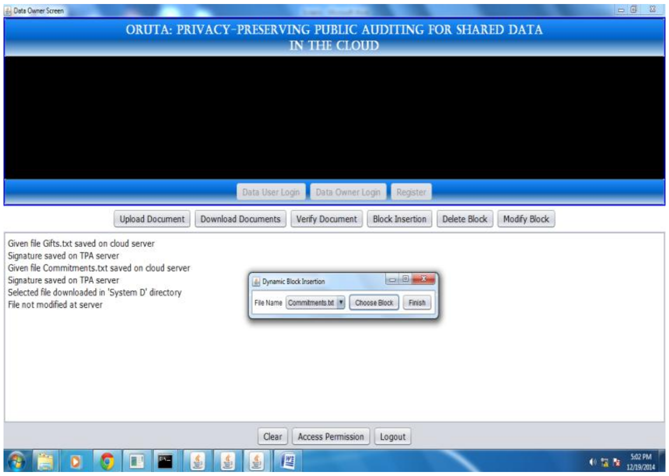


```
Given file Gifts.txt saved on cloud server
Signature saved on TPA server
Given file Commitments.txt saved on cloud server
Signature saved on TPA server
Selected file downloaded in 'System D' directory
File not modified at server
Given file block Commitments.txt_block3.txt inserted on c
Given block metadata(signature) inserted on TPA server
Given file block Gifts.txt_block3.txt inserted on cloud serv
Given block metadata(signature) inserted on TPA server
Selected file downloaded in 'System D' directory
Selected file downloaded in 'System D' directory
Given file blockCommitments.txt_block2.txt modified on c
Given block Commitments.txt_block2.txt metadata(signa
```

File Name: referral.txt

Files of Type: All Files

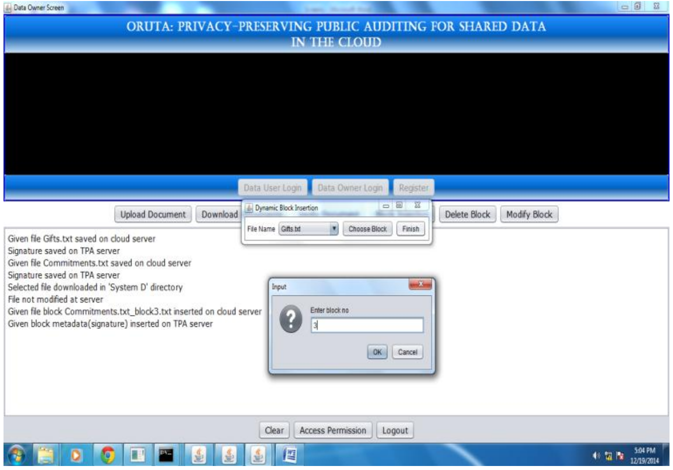
Fig.13.



```
Given file Gifts.txt saved on cloud server
Signature saved on TPA server
Given file Commitments.txt saved on cloud server
Signature saved on TPA server
Selected file downloaded in 'System D' directory
File not modified at server
```

Fig.14.

Click on Finish Button and it will ask the Block number (Note: The block number should not be the value of previous block number)



```
Given file Gifts.txt saved on cloud server
Signature saved on TPA server
Given file Commitments.txt saved on cloud server
Signature saved on TPA server
Selected file downloaded in 'System D' directory
File not modified at server
Given file block Commitments.txt_block3.txt inserted on cloud server
Given block metadata(signature) inserted on TPA server
```

Enter block no: 3

Fig.15.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose Oruta, a privacy-preserving public audit ing mechanism for shared data in the cloud. We utilize ring signatures to construct homomorphic authenticators, so that a public verifier is able to audit shared data integrity without retrieving the entire data, yet it cannot distinguish who is the signer on each block. To improve the efficiency of verifying multiple auditing tasks, we further extend our mechanism to support batch auditing. There are two interesting problems we will continue to study for our future work. One of them is traceability, which means the ability for the group manager (i.e., the original user) to reveal the identity of the signer based on verification metadata in some special situations. Since Oruta is based on ring signatures, where the identity of the signer is unconditionally protected [21], the current design of ours does not support traceability. To the best of our knowledge, designing an efficient public auditing mechanism with the capabilities of preserving identity privacy and supporting traceability is still open. Another problem for our future work is how to prove data freshness (prove the cloud possesses the latest version of shared data) while still preserving identity privacy.

VII. REFERENCES

[1]B. Wang, B. Li, and H. Li, “Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud,” Proc. IEEE Fifth Int’l 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 Int’l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT ’08), pp. 90- 107, 2008.

Author’s Profile:

Nallamothu Gopi hailed from Khammama (Dist.) born on 08th june 1992 .He Received B.Tech in Computer Science and Engineering from ADAM’S engineering college, JNTUH, Khammm, Dist, (TG). He Pursuing M.TECH in C.S.E from Abdul Kalam Institute of Technological Sciences, JNTUH, Vepalagadda, Kothagudem, Khammam Dist (TG).

B.Varsha has received her **B.Tech** degree in computer science engineering from JNTU, Hyderabad in 2011 and **M.Tech** degree in computer science engineering from JNTU Hyderabad in 2014. Presently working as Assistant Professor at Abdul Kalam Institute of Technological Sciences, kothagudem, Telegana.

N. Swapna Suhasini received MCA and M.Tech in Computer Science Engineering from Osmania University in 2002 & 2008 respectively. She is an author of 8 journal and conference papers. Presently she is working as Associate Professor and HOD of CSE Department in Abdulkalam Institute of Technological Sciences, Vepalagadda, Kothagudem, Affiliated to JNTU Hyderabad. She is Life Member of ISTE and also member of CSI. Her research and study interests include Data Mining, Big Data Analytics, and Cloud computing.