AUDITING SECURE STORAGE AND PROVIDING EFFECTIVE KEY UPDATES

Mrs Anuradha Anumolu[1], Mr. Bulleswarao Bathula [2]

#1 Associate Professor , department of IT at DVR & Dr HS MIC College of Technology (Autonomous), Kanchikacherla, NTR District.

#2 MCA student in the Department of Computer Applications (DCA) at DVR & DR. HS MIC College of Technology, Kanchikacherla, NTR District

ABSTRACT_ It is run of the mill for information to be divided between a few clients as well as being put away on cloud servers because of cloud information administrations. Sadly, on the grounds that human mistake and equipment/programming breakdowns are genuine, cloud information honesty is raised doubt about. Without getting each of the information from the cloud server, various methodologies have been created to empower successful inspecting of cloud information honesty by both public verifiers and information proprietors.

However, utilizing these current systems for public audits of shared data integrity will eventually expose personal information—such as identity privacy—to public verifiers. In this study, we present a novel privacy-preserving strategy that makes it easier to conduct public audits of shared cloud-stored data. In particular, we calculate the verification metadata needed to confirm the accuracy of shared data using ring signatures. Our methodology permits public verifiers to productively confirm the honesty of shared information without recovering the whole record, while keeping up with the protection of the underwriter's personality on each block. Besides, as opposed to affirming each examining work each in turn, our procedure might deal with a few reviewing errands immediately. The results of our trials show how fruitful and productive our methodology is at guaranteeing the trustworthiness of shared information.

1.INTRODUCTION

It is run of the mill for information to be divided between a few clients as well as being put away on cloud servers because of cloud information administrations. Sadly, on the grounds that human mistake and equipment/programming breakdowns are genuine, cloud information honesty is raised doubt about. Without getting each of the information from the cloud server, various methodologies have been created to empower successful inspecting of cloud

information honesty by both public verifiers and information proprietors.

However, utilizing these current systems for public audits of shared data integrity will eventually expose personal information—such as identity privacy—to public verifiers. In this study, we present a novel privacy-preserving strategy that makes it easier to conduct public audits of shared cloud-stored data. In particular, we calculate the verification metadata needed to confirm the accuracy of shared data using ring signatures. Our methodology permits public verifiers to productively confirm the honesty of shared information without recovering the whole record, while keeping up with the protection of the underwriter's personality on each block. Besides, as opposed to affirming each examining work each in turn, our procedure might deal with a few reviewing errands immediately. The results of our trials show how fruitful and productive our methodology is at guaranteeing the trustworthiness of shared information.

2.LITERAURE SURVEY

1) Efficient dispersal of information for security, load balancing, and fault tolerance

AUTHORS: M. Rabin

An Information Dispersal Algorithm (IDA) is developed that breaks a file F of length L = ⏌ F⏋ into n pieces Fi, l ≤ i ≤ n, each of length ⏌Fi⏋ = L/m, so that every m pieces suffice for reconstructing F. Dispersal and reconstruction are computationally efficient. The sum of the lengths ⏌Fi⏋ is (n/m) · L. Since n/m can be chosen to be close to l, the IDA is space efficient. IDA has numerous applications to secure and reliable storage of information in computer networks and even on single disks, to fault-tolerant and efficient transmission of information in networks, and to communications between processors in parallel computers. For the latter problem provably time-efficient and highly fault-tolerant routing on the n-cube is achieved, using just constant size buffers.

2) Provable data possession at untrusted stores

AUTHORS: G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song

We introduce a model for *provable data possession* (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of

blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage system.

We present two provably-secure PDP schemes that are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. Experiments using our implementation verify the practicality of PDP and reveal that the performance of PDP is bounded by disk I/O and not by cryptographic computation.

3) Pors: Proofs of retrievability for large files

AUTHORS: A. Juels and B. S. Kaliski

In this paper, we define and explore *proofs of retrievability* (PORs). A POR scheme enables an archive or back-up service (prover) to produce a concise proof that a user (verifier) can retrieve a target file *F*, that is, that the archive retains and reliably transmits file data sufficient for the user to recover *F* in its entirety.

A POR may be viewed as a kind of cryptographic proof of knowledge (POK), but one specially designed to handle a *large* file (or bitstring) *F*. We explore POR protocols here in which the communication costs, number of memory accesses for the prover, and storage requirements of the user (verifier) are small parameters essentially independent of the length of *F*. In addition to proposing new, practical POR constructions, we explore implementation considerations and optimizations that bear on previously explored, related schemes.

In a POR, unlike a POK, neither the prover nor the verifier need actually have knowledge of *F*. PORs give rise to a new and unusual security definition whose formulation is another contribution of our work.

We view PORs as an important tool for semi-trusted online archives. Existing cryptographic techniques help users ensure the privacy and integrity of files they retrieve. It is also natural, however, for users to want to verify that archives do not delete or modify files prior to retrieval. The goal

of a POR is to accomplish these checks *without users having to download the files themselves*. A POR can also provide quality-of-service guarantees, i.e., show that a file is retrievable within a certain time bound.

3.PROPOSED SYSTEM

In this paper, to settle the above security issue on shared information, we propose Oruta, an original protection safeguarding public examining component.

All the more explicitly, we use ring marks to develop homomorphic authenticators in Oruta, so a public verifier can confirm the trustworthiness of shared information without recovering the whole information while the personality of the underwriter on each block in shared information is kept hidden from the public verifier.

Likewise, we further stretch out our instrument to help group examining, which can play out different evaluating errands all the while and work on the proficiency of check for numerous reviewing assignments. In the mean time, Oruta is viable with irregular covering, which has been used in WWRL and can protect information security from public verifiers. In addition, we support dynamic data by utilizing index hash tables from an earlier public auditing solution. An undeniable level examination among Oruta and it is introduced to exist instruments.

3.1 IMPLEMENTAION

Cloud server

In the first module, we design our system with Cloud Server, where the datas are stored globally. Our mechanism, Oruta, should be designed to achieve following properties:

(1) Public Auditing: A public verifier is able to publicly verify the integrity of shared data without retrieving the entire data from the cloud.

(2) Correctness: A public verifier is able to correctly verify shared data integrity.

(3) Unforgeability: Only a user in the group can generate valid verification metadata (i.e., signatures) on shared data.

(4) Identity Privacy: A public verifier cannot distinguish the identity of the signer on each block in shared data during the process of auditing.

Group of users

There are two types of users in a group: the original user and a number of group users. The original user initially creates shared data in the cloud, and shares it with group users. Both the original user and group users are 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.

Owner Registration: In this module an owner has to upload its files in a cloud server, he/she should register first. Then only he/she can be able to do it. For that he needs to fill the details in the registration form. These details are maintained in a database.

Owner Login: In this module, owners have to login, they should login by giving their email id and password.

User Registration: In this module if a user wants to access the data which is stored in a cloud, he/she should register their details first. These details are maintained in a Database.

User Login: If the user is an authorized user, he/she can download the file by using file id which has been stored by data owner when it was uploading.

Public verifier

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

Auditing Module

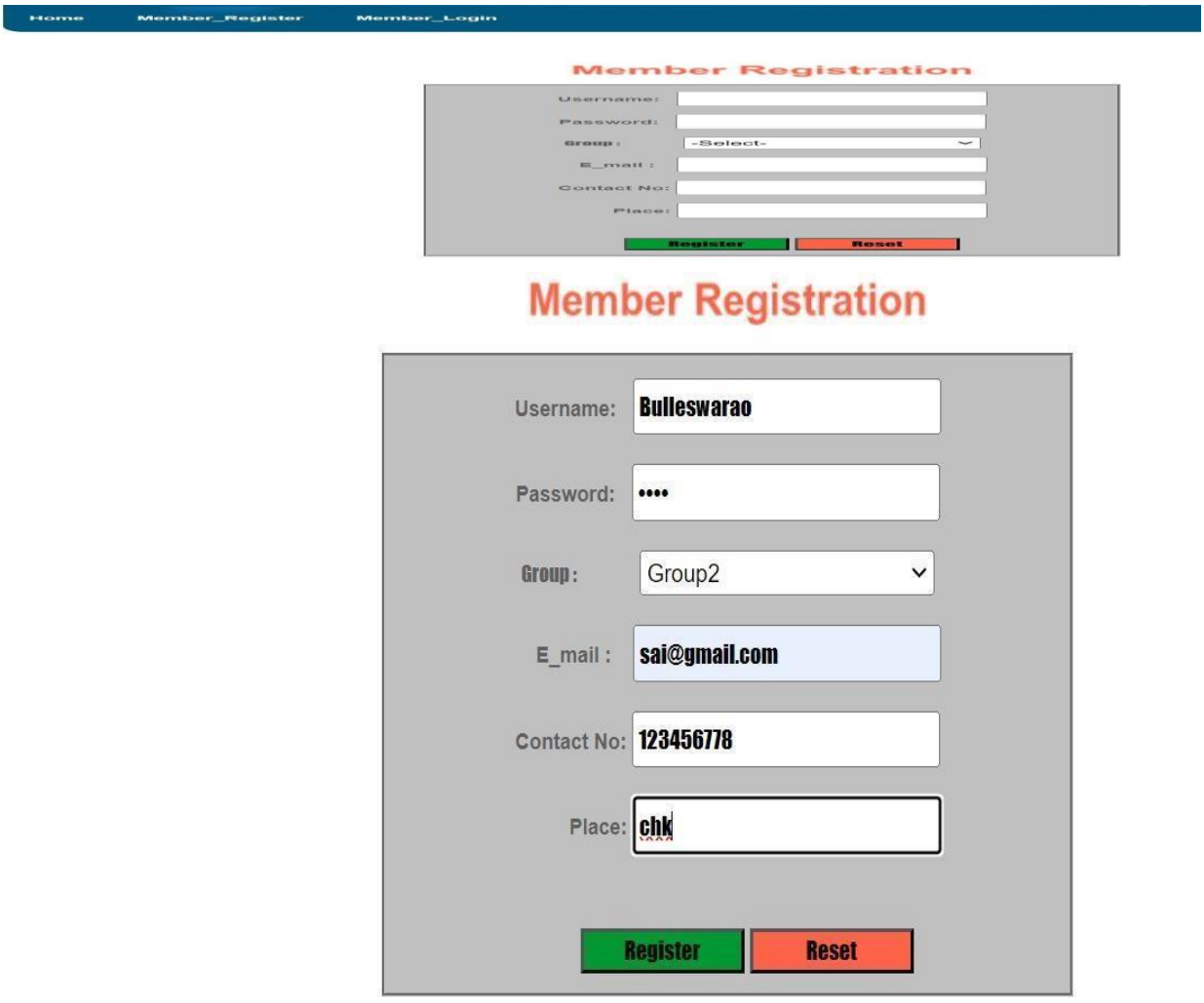
In this module, if a third party auditor TPA (maintainer of clouds) should register first. This system allows only cloud service providers. After third party auditor gets logged in, He/ She can see how many data owners have uploaded their files into the cloud. Here we are providing TPA for maintaining clouds.

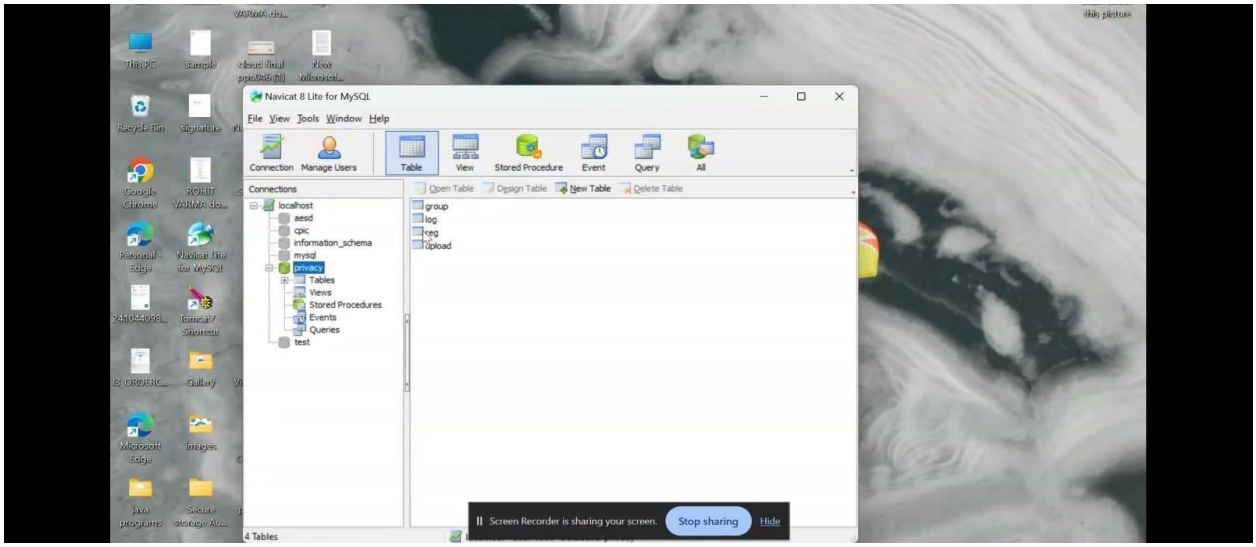
We only consider how to audit the integrity of shared data in the cloud with *static groups*. It means the group is pre-defined before shared data is created in the cloud and the

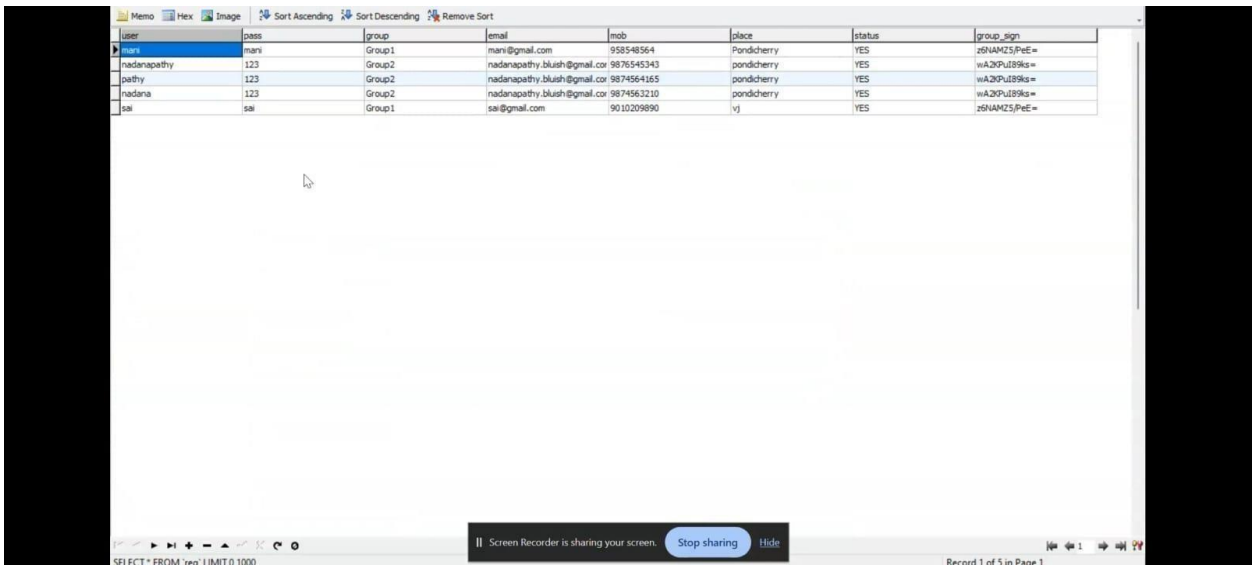
membership of users in the group is not changed during data sharing.

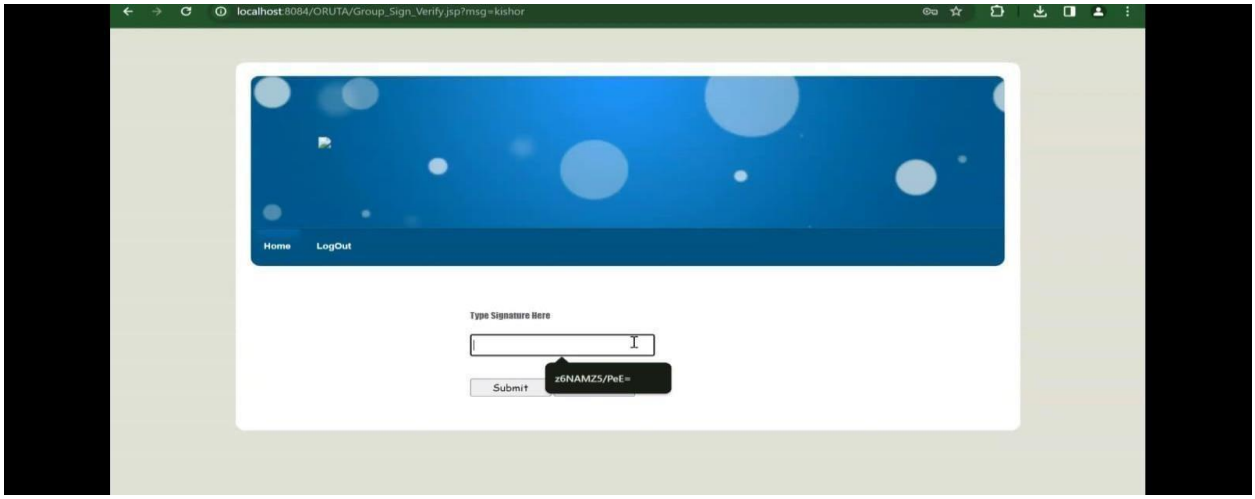
The original user is responsible for deciding who is able to share her data before outsourcing data to the cloud. Another interesting problem is how to audit the integrity of shared data in the cloud with *dynamic groups* — a new user can be added into the group and an existing group member can be revoked during data sharing — while still preserving identity priva

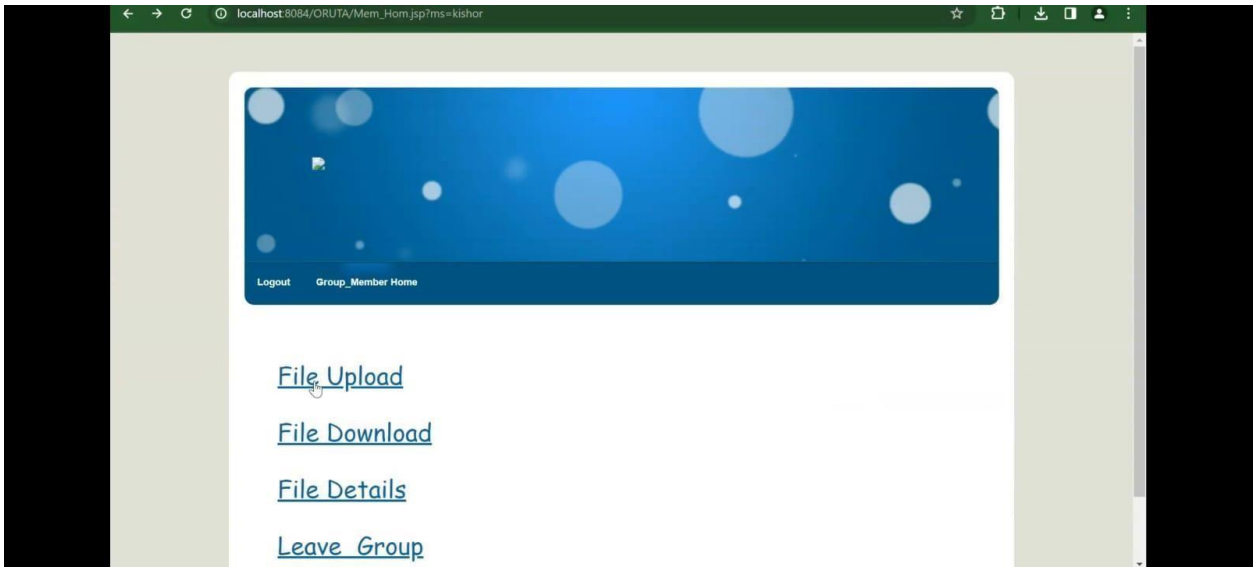
4.RESULTS AND DISCUSSION

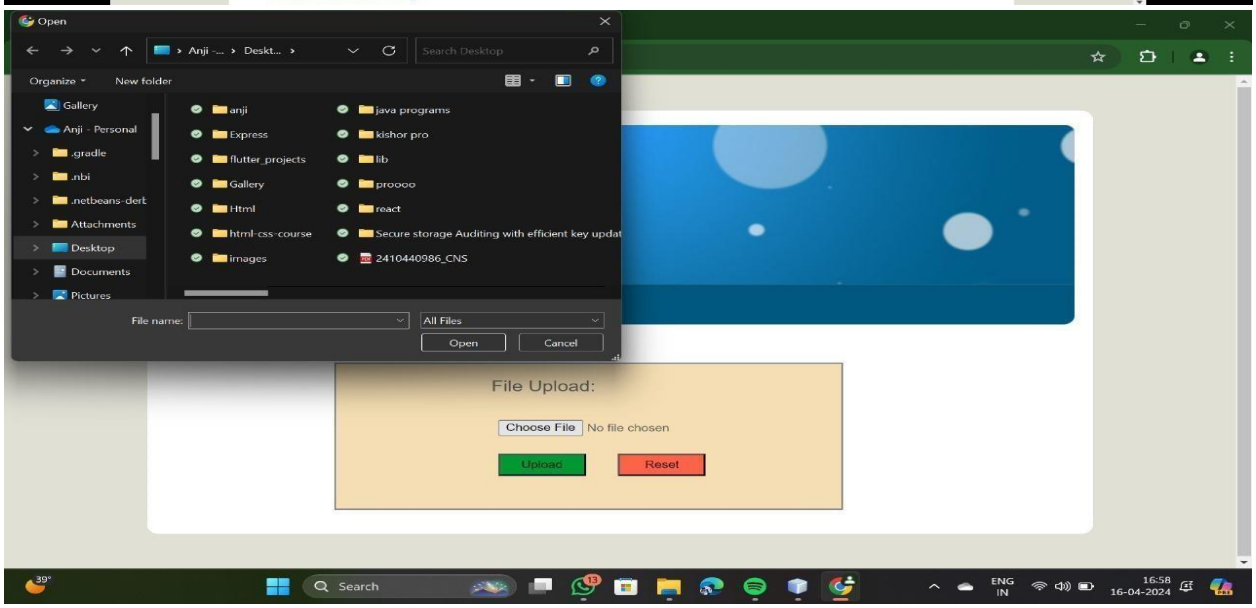


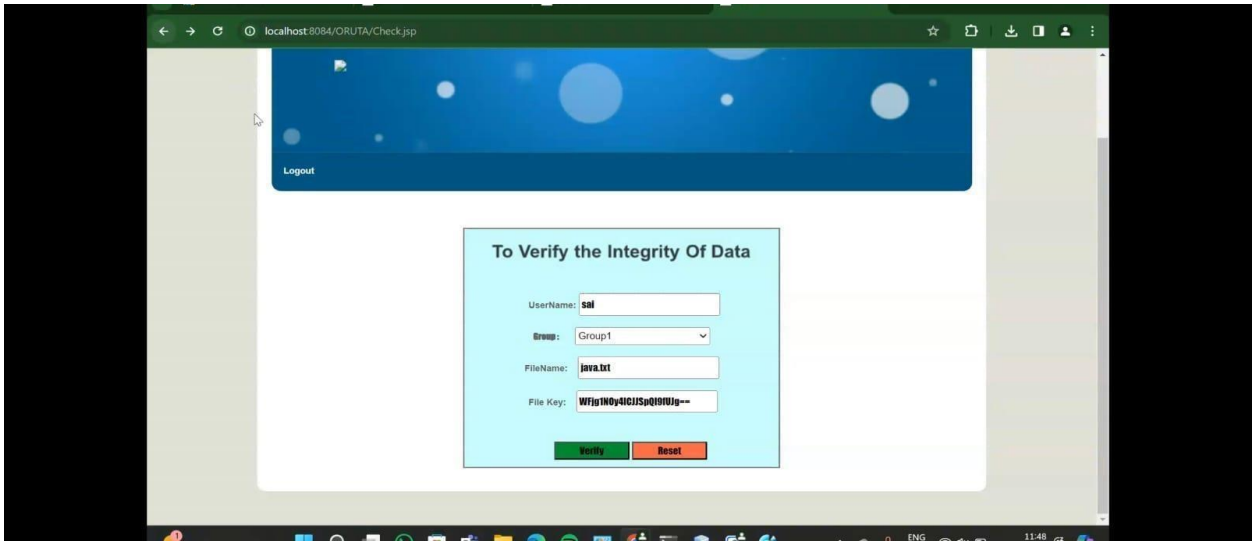

Navicat 8 Lite for MySQL
File View Tools Window Help
Connection Manage Users Table View Stored Procedure Event Query All
Connections
localhost
aesd
cpic
information_schema
mysql
privacy
Tables
Views
Stored Procedures
Events
Queries
test
group
log
reg
Upload
4 Tables
Screen Recorder is sharing your screen.
Stop sharing
Hide

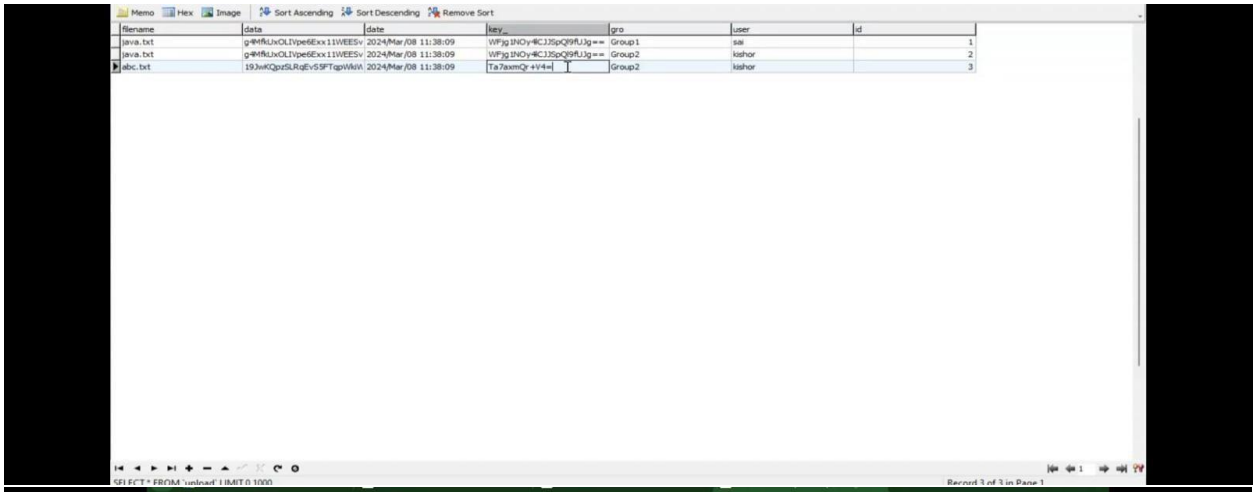

Memo Hex Image Sort Ascending Sort Descending Remove Sort
user pass group email mob place status group_sign
mani mani Group1 mani@gmail.com 958548564 Pondicherry YES z6NAMZ5/PeE=
nadanapathy 123 Group2 nadanapathy.bluish@gmail.cor 9876545343 pondicherry YES wA2KPuI89ks=
pathy 123 Group2 nadanapathy.bluish@gmail.cor 9874564165 pondicherry YES wA2KPuI89ks=
nadana 123 Group2 nadanapathy.bluish@gmail.cor 9874563210 pondicherry YES wA2KPuI89ks=
sai sai Group1 sai@gmail.com 9010209890 vj YES z6NAMZ5/PeE=
Screen Recorder is sharing your screen.
Stop sharing
Hide

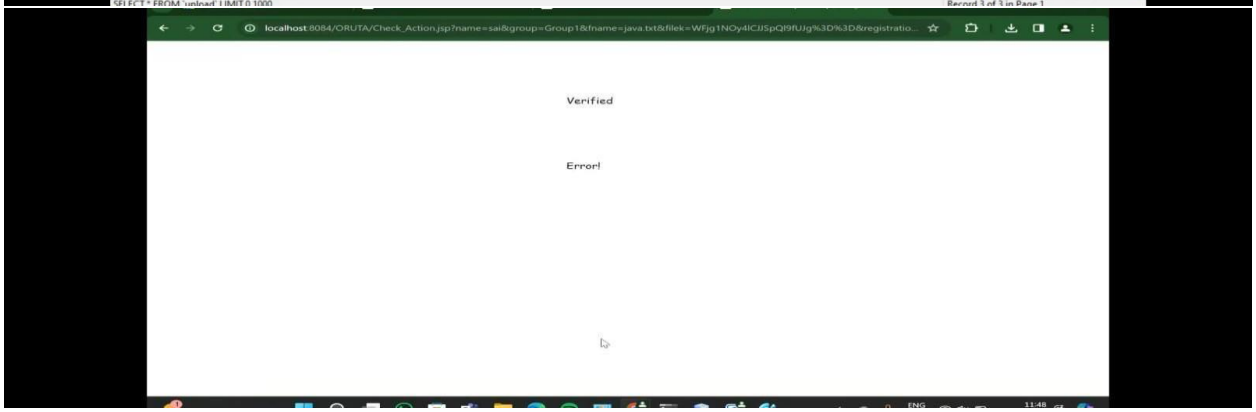

localhost:8084/ORUTA/Group_Sign_Verify.jsp?msg=kishor
Home
LogOut
Type Signature Here
Submit
z6NAMZ5/PeE=


localhost:8084/ORUTA/Mem_Hom.jsp?ms=kishor
Logout
Group_Member Home
File Upload
File Download
File Details
Leave Group


Open
Organize
New folder
Search Desktop
Gallery
Anji - Personal
.gradle
.nbi
.netbeans-dert
Attachments
Desktop
Documents
Pictures
anji
Express
flutter_projects
Gallery
Html
html-css-course
images
java programs
kishor pro
lib
proooo
react
Secure storage Auditing with efficient key updat
2410440986_CNS
File name:
All Files
Open
Cancel
File Upload:
Choose File
No file chosen
Upload
Reset
Search
ENG
IN
16:58
16-04-2024







4.CONCLUSION

In this paper, we propose Oruta, a privacy-preserving public auditing 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.

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.

[11] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic Provable Data

Possession," Proc. 16th ACM Conf. Computer and Comm. Security (CCS'09), pp. 213-222, 2009. [12] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing," Proc. 14th European Conf. Research in Computer Security (ESORICS'09), pp. 355-370, 2009.

[13] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS'09), pp. 1-

[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 (CCSW'10), 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 (SAC'11), pp. 1550-1557, 2011.

[16] N. Cao, S. Yu, Z. Yang, W. Lou, and Y.T. Hou, "LT Codes-Based Secure and Reliable Cloud

Storage Service," Proc. IEEE INFOCOM, 2012.

[17] B. Wang, B. Li, and H. Li, "Certificateless Public Auditing for Data Integrity in the Cloud," Proc. IEEE Conf. Comm. and Network Security (CNS'13), pp. 276-284, 2013.

[18] C. Wang, S.S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.

[19] B. Wang, B. Li, and H. Li, "Public Auditing for Shared Data with Efficient User Revocation in the

Cloud," Proc. IEEE INFOCOM, pp. 2904-2912, 2013.

[20] B. Wang, B. Li, and H. Li, "Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud," IEEE Trans. Services Computing, 20 Dec. 2013, DOI: 10.1109/TSC.2013.2295611. [21] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'l Conf. Theory and

AUTHOR PROFILE

Mrs AnuradhaAnumolu completed her M.Tech (CSE) from Acharya Nagarjuna University. She has published more than10 papers in indexing Journals, currently working as an Associate Professor and Head in the department of IT at DVR & Dr HS MIC College of Technology (Autonomous), Kanchikacherla, NTR District. Her areas of interest are Data Structures, Data Mining, Cloud Computing, Artificial Intelligence.

Mr. BULLESWARAO BATHULA, as MCA student in the department of DCA at DVR & DR. HS MIC COLLEGE OF TECHNOLOGY (Autonomous), **Kanchikacherla, NTR District**. He has completed B.Sc(MPC) in Vishwabharathi Degree College, Jaggayyapet from Krishna University . His areas of interests are Java, Python and web development.