Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud

With cloud data services, it is commonplace for data to be not only stored in the cloud, but also shared across multiple users. Unfortunately, the integrity of cloud data is subject to skepticism due to the existence of hardware/software failures and human errors. Several mechanisms have been designed to allow both data owners and public verifiers to efficiently audit cloud data integrity without retrieving the entire data from the cloud server. However, public auditing on the integrity of shared data with these existing mechanisms will inevitably reveal confidential information—identity privacy—to public verifiers. In this paper, we propose a novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud. In particular, we exploit ring signatures to compute verification metadata needed to audit the correctness of shared data. With our mechanism, the identity of the signer on each block in shared data is kept private from public verifiers, who are able to efficiently verify shared data integrity without retrieving the entire file. In addition, our mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one. Our experimental results demonstrate the effectiveness and efficiency of our mechanism when auditing shared data integrity.

EXISTING SYSTEM:

v Many mechanisms have been proposed to allow not only a data owner itself but also a public verifier to efficiently perform integrity checking without downloading the entire data from the cloud, which is referred to as public auditing . In these mechanisms, data is divided into many small blocks, where each block is independently 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 data user (e.g., researcher) who would like to utilize the owner's data via the cloud or a third-party auditor (TPA) who can provide expert integrity checking services.

v Moving a step forward, Wang et al. designed an advanced auditing mechanism .so that during public auditing on cloud data, the content of private data belonging to a personal user is not disclosed to any public verifiers. Unfortunately, current public auditing solutions mentioned above only focus on personal data in the cloud .We believe that sharing data among multiple users is perhaps one of the most engaging features that motivates cloud storage. Therefore, it is also necessary to ensure the integrity of shared data in the cloud is correct.

v Existing public auditing mechanisms can actually be extended to verify shared data integrity. However, a new significant privacy issue introduced in the case of shared data with the use of existing mechanisms is the leakage of identity privacy to public verifiers.

DISADVANTAGES OF EXISTING SYSTEM:

1. Failing to preserve identity privacy on shared data during public auditing will reveal significant confidential information to public verifiers.

2. Protect this confidential information is essential and critical to preserve identity privacy from public verifiers during public auditing.

PROPOSED SYSTEM:

] In this paper, to solve the above privacy issue on shared data, we propose Oruta, a novel privacy-preserving public auditing mechanism.

] More specifically, we utilize ring signatures to construct homomorphic authenticators in Oruta, 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 further extend our mechanism to support batch auditing, which can perform multiple auditing tasks simultaneously and improve the efficiency of verification for multiple auditing tasks.

] Meanwhile, Oruta is compatible with random masking, which has been utilized in WWRL and can preserve data privacy from public verifiers. Moreover, we also leverage index hash tables from a previous public auditing solution to support dynamic data. A high-level comparison among Oruta and existing mechanisms is presented.

ADVANTAGES OF PROPOSED SYSTEM:

1. A public verifier is able to correctly verify shared data integrity.
2. A public verifier cannot distinguish the identity of the signer on each block in shared data during the process of auditing.
3. The ring signatures generated for not only able to preserve identity privacy but also able to support blockless verifiability.

MODULES:

] Cloud server

] Group of users

] Public verifier

] Auditing Module

MODULES DESCRIPTION:

Cloud server

ü 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:

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

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

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

ü (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

ü 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 thirdparty 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.

ü 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.

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

ü 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.

ü 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

ü 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.

ü 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

ü 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.

ü 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.

ü 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 privacy.

SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

Ø System	:	Pentium IV 2.4 GHz.
Ø Hard Disk	:	40 GB.
Ø Floppy Drive	:	1.44 Mb.
Ø Monitor	:	15 VGA Colour.
Ø Mouse	:	Logitech.
Ø Ram	:	512 Mb.

SOFTWARE REQUIREMENTS:

- Ø Operating system : Windows XP/7.
- Ø Coding Language : JAVA/J2EE
- Ø IDE : Netbeans 7.4
- Ø Database : MYSQL