HARS mechanism for Privacy-preserving Public Auditing on shared data in cloud (Oruta)

Ms. Sonam Mahadev Kamble¹, Prof. Archana Lomte²
JSPM’s Bhivarabai Sawant Institute of Technology and Research, Wagholi, Pune.
Maharashtra, India.
sonamkamble@ymail.com¹, archanalomte@gmail.com²

Abstract

The cloud data services provide a common place for storing data as well as share data across multiple users. Still there are some issues of integrity because of hardware or software failures and human errors. Many systems allow both data owners (group users) and public verifiers (Third Party Auditor) 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. The proposed system provides the privacy-preserving mechanism that supports public auditing on shared data stored in the cloud. Here they exploit ring signatures to compute verification signature needed to audit the correctness of shared data. Using this 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 downloading the entire file from the cloud and also able to perform multiple auditing tasks simultaneously instead of verifying them one by one.

Index Terms: Public Auditing, Privacy preserving, shared Data, Cloud Computing.

1. Introduction

Using cloud computing and data storage, users are able to access and share resources provided by cloud service providers at a low and marginal cost. It is routine for users who depends on to access cloud storage services to share data with others in a group, as data sharing becomes standard feature in most cloud storage offerings, including Google Drive and Dropbox. The integrity of data in cloud storage, however, is subject to skepticism and search, as data stored in the cloud can easily be lost or corrupted due to the unexpected hardware or software failures and human error.

The traditional approach for maintaining data correctness is to retrieve the entire data from the cloud, and then verify data integrity by checking the correctness of signatures (e.g., RSA) or hash values (e.g., MD5) of the entire data [2]. Using this conventional approach successfully check the correctness of cloud data. However, the efficiency of using this traditional approach on cloud data is not better.

In the proposed system, data is divided into many small blocks, where each block is independently signed by the data 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 owners’ data via the cloud or a third-party auditor (TPA) who can provide expert integrity checking services. By Ring Signature system is rule on all users so it called Oruta stands for One Ring to Rule Them All [11].

Existing public auditing mechanisms can actually be extended to verify shared data integrity and data freshness. 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. To protect the confidential information, it is essential and critical to preserve identity privacy from public verifiers during public auditing [3].

However, it also poses a significant risk to the confidentiality of those stored files. To preserve data privacy, a basic solution is to encrypt data files, and then upload the encrypted data into the cloud. Unfortunately, designing an efficient and secure data sharing scheme for groups in the cloud is not an easy task due to the following challenging issues. First, identity Second, that any member in a group should be able to data storing and sharing
services provided by the cloud, which is defined as the multiple-owner manner. Third, member revocation and signed receipt e.g., new member participation and current member revocation in a group.

2. Literature Review

A cryptographic algorithm categorised as a symmetric and asymmetric algorithms. Symmetric algorithm uses a single key known as secret key and used for both the process encryption and decryption whereas asymmetric algorithm uses two keys; one is the public key made available publically and the other one is the private key, which is kept secret used to decrypt the data. Breaking the private key is not easily possible even if the corresponding public key is known well in advance. Examples of symmetric algorithm comprise of Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA), Advanced Encryption Standard (AES) on the other hand asymmetric key algorithm include RSA algorithm [3].

PDP (Provable Data Possession) [2], allows a verifier to check the correctness of a client’s data stored at an untrusted server. By utilizing RSA-based homomorphic authenticators and sampling strategies, the verifier is able to publicly audit the integrity of data without retrieving the entire data, which is referred to as public auditing. Unfortunately, their mechanism is only suitable for auditing the integrity of personal data.

A. Juels and B.S. Kaliski [7] defined another similar model called Proofs of Retrievability (POR), which is also able to check the correctness of data on an untrusted server. The original file is added with a set of randomly-valued check blocks called sentinels. The verifier challenges the untrusted server by specifying the positions of a collection of sentinels and asking the untrusted server to return the associated sentinel values.

To support dynamic data, Ateniese et al. [9] presented an efficient PDP mechanism based on symmetric keys. This mechanism can support update and delete operations on data; however, insert operations are not available in this mechanism. Because it exploits symmetric keys to verify the integrity of data, it is not public verifiable and only provides a user with a limited number of verification requests. Erway et al. [10] introduced dynamic provable data possession (DPDP) by using authenticated dictionaries, which are based on rank information.

3. Proposed System

System Model

As illustrated in Figure.1, the system model consist three parties: the cloud server, a group of users and a public verifier (Admin). There are two types of users in a group: the original user means data owner and a number of users in group. 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 or Admin, such as a third party auditor providing efficient 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 wants 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 done between a public verifier and the cloud server or the data storage server.

Figure.1. System Architecture

Threat Model

Integrity Threats, an opponent or attacker and sometimes cloud service provider may try to corrupt or violate the integrity of shared data and want to provide incorrect data to user rather correct information.

Privacy Threats, the identity of user on group is confidential, during the process of auditing, a semi-trusted TPA or public verifier, who audit the integrity of shared data, may try to reveal the identity of the signer on each block in
shared data based on verification information. Once the TPA recognizes the identity of the group user on each block, it can easily distinguish a high-value target (a particular user in the group or a special block in shared data).

### Design Objectives

The main design goals of the proposed scheme includes, (1) Public Auditing: A public verifier or Third party Auditor (TPA) 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: Group user can share data with secrete key unauthorised user do not have access to data. (4) Identity Privacy: A TPA cannot distinguish the identity of the signer or user who shares data on each block in shared data during the process of auditing.

### 4. Methodology

#### Ring signature

The concept of ring signatures was first proposed by Rivest et al. in 2001. Using 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 signers identity with a probability more than 1/d. This property can be used to preserve the identity of the signer from a verifier [11]. The ring signature scheme introduced by Boneh et al, is constructed on bilinear maps. Proposed system extends this ring signature scheme to construct public auditing mechanism.

#### Homomorphic Authenticators

Homomorphic authenticators, also called homomorphic verifiable tags, allow a public verifier to check the integrity of data stored in the cloud without downloading the entire data. They have been widely used as building blocks in the previous public auditing mechanisms. In a homomorphic authenticable signature scheme only a valid user can generate signature using private key, which denotes a homomorphic authenticator scheme based on signatures [11].

### 5. Algorithm

Homomorphic Authenticable Ring Signature (HARS) scheme is extended from a classic ring signature. The ring signatures generated by HARS are not only able to preserve identity privacy but also able to support blockless verifiability.

#### Construction of HARS

HARS contains three part Key Generation, Ring Signature and Ring Verification. In key generation, each user in the group generates his/her public key and private key. In Ring signature, 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 that can distinguish the corresponding block from others. A verifier is able to check whether a given block is signed by a group member in Ring Verification.

#### Construction of Oruta

Now, turns to details of public auditing mechanism. It includes Key generation where users generate their own public or private key pairs. Next part is Signature generation: 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 function Modify. Proof Generation is operated by a public verifier and the cloud server together to interactively generate a proof of possession of shared data. In Proof verification, the public verifier checks the integrity of shared data by verifying the proof. To construct this model need algorithm which provides encryption and decryption of data and give the security to data while sharing with the group users and also provides the ring signature for privacy preserving.

#### Mathematical Model

**Assumptions:**

- **U=** {Universal set of users who shared private information}. U= \{U_1, U_2, U_3, \ldots \ldots U_n\}
- **G=** {set of Groups}. G= \{G_1, G_2 \ldots \ldots G_n\}
- **X=** {Set of private information shared by user of group G}. X= \{X_1, X_2, X_3, \ldots \ldots X_n\}
- **K=** {Secrete key entered by user to access data}. Secrete key assigned to group to share information. K= \{K_1, K_2, \ldots \ldots K_n\}
- **Pv=** is public verifier.

**Description:**

i. Every usr must be register to the system.

ii. Public verifier Pv assign that user to specific Group G.

iii. After assigning group the secrete key is generated for that user of \{K_1, K_2, \ldots \ldots K_n\}. 

#### Description:
$K_1,...,K_n \}$ depends on to which group the user is assign.

iv. Using that secret key the private information on group is shared, data is not stored in local storage it is stored on cloud.

**Database:** For the database the cloud database is used that is Salesforce. Which provide data storage and which is stored in Encrypted form not in plain text. There are different database are created which in for of data Object. Salesforce use their own language to form database is SOQL (Structured Object Query Language). Here total five Tables or Objects created for the project.

i. **User:** Store user registration information

ii. **Group:** Group information

iii. **AdminMaster:** Admin information

iv. **Grouping:** Stored Shared information with group details and user of which group.

v. **GroupKey:** Store the group key information which are assign to group.

These algorithms used to encryption and decryptions of data are as follows:

a. **RC6:** In cryptography, RC6 (Rivest Cipher 6) is a symmetric key block cipher derived from RC5. It was designed by Ron Rivest, Matt Robshaw, Ray Sidney and Yiqun Lisa Yin to meet the requirements of the Advanced Encryption Standard (AES) competition [12].

RC6 proper has a block size of 128 bits and supports key sizes of 128, 192, and 256 bits, but, like RC5, it may be parameterized to support a wide variety of word-lengths, key sizes, and number of rounds. RC6 is very similar to RC5 in structure, using data-dependent rotations, modular addition, and XOR operations; in fact, RC6 could be viewed as interweaving two parallel RC5 encryption processes, however, RC6 does use an extra multiplication operation not present in RC5 in order to make the rotation dependent on every bit in a word and not just the least significant few bits, encryption or decryption with RC6 – w/ t/ b [12].

**Encryption**

**Input:** Plaintext stored in four w-bit, input registers M, N, O and P, r is the number of rounds w-bit round keys $S[0, ..., 2r + 3]$  
**Output:** Ciphertext stored in M, N, O, P  
**Procedure:**  
\[ N = N + S[0] \]  
\[ P = P + S[1] \]  
for \( i = 1 \) to \( r \) do  
\(
\begin{align*}
  t &= (N^*(2N + 1)) << < lg w \\
  u &= (P^*(2P + 1)) << < lg w \\
  M &= ((M \text{ Ex-OR } t) << < u) + S[2i] \\
  O &= ((O \text{ Ex-OR } u) << < t) + S[2i + 1] \\
  (M, N, O, P) &= (N, O, P, M) \\
  M &= M + S[2r + 2] \\
  O &= O + S[2r + 3]
\end{align*}
\)

**Decryption:**

**Input To Decryption:** (P, M, O, N)  
Ciphertext stored in four w-bit input registers M, N, O, P. Number \( r \) of rounds, w-bit round keys S [0, 2r + 3]  
**Output:** Plaintext stored in M, N, O, P  
**Procedure:**  
\[ O = O - S[2r + 3] \]  
\[ M = M - S[2r + 2] \]  
for \( i = r \) down to \( 1 \) do  
\(
\begin{align*}
  O &= O - S[2i + 1] \\
  M &= M - S[2i] \\
  M &= M + S[2r + 2] \\
  O &= O - S[2i + 1] \\
  (M, N, O, P) &= (P, M, N, O) \\
  t &= (N^*(2N + 1)) <<< lg w \\
  u &= (P^*(2P + 1)) <<< lg w \\
  O &= ((O - S[2i + 1]) >>> t) \text{ Ex-OR } u \\
  M &= ((M - S[2i]) >>> u) \text{ Ex-OR } t \\
  P &= P - S[1] \\
  N &= N - S[0]
\end{align*}
\)

To encrypt and decrypt data a key is needed and that key is generated by SPEKE algorithm.

b. **SPEKE:** SPEKE (Simple Password Exponential Key Exchange) is a cryptographic method for password authenticated key agreement. SPEKE is new field of password-authenticated key exchange. Using SPEKE secret key is generated and used to share data as well decrypt data.
6. Result and discussion

For the Privacy preserving and public auditing the RC6 algorithm used which provide Encryption and decryption. Encrypted data and key are stored separately on different storage media. Following are the steps:

Step1: First User must be register and Admin assign that User to specific group.

Step2: If any User wants to share data or files then must share data with the Secrete key which is produced after group assignment and present on email.

Step3: Before decrypting the data the users have to enter same secrete key which is sent on his mail and used to generate original data.

Step4: For accessing the data the user is restricted in read only mode and for insert, modify and delete the notification is sent to admin.

Step5: After encryption or decryption the original data is deleted from the Decryption services only encrypted data is not removed.

Step6: For securing the Account and Service Hijacking, we are eliminating the TPA. The work of TPA will be done by DATA AUDITING SERVICE and some by Admin, the user does not have the time, feasibility or resources to perform the storage correctness verification; he can optionally do this task using data auditor, making the cloud storage publicly verifiable and securely introduce an effective DATA AUDITING SERVICE, the auditing process should bring in no new vulnerabilities towards user data privacy.

6. Conclusion

The security aspect in cloud is major concern thus we have proposed novel system which can process the request in grouping or batch manner which can enhance performance and efficiency of data transfer in system. The algorithm clearly shows improvements to its predecessor in various fashions like security, transfer of data, scalability and Data Integrity. Also system proposed an innovative public auditing system for shared data with efficient user revocation in the cloud on misbehaving and existing users in the group can save a significant amount of computation and communication resources during user interaction.

ACKNOWLEDGMENT

I would like to sincerely thank my guide and ME coordinator Prof. A.C. Lomte (Professor, at Computer Department of BSIOTR, Wagholi, Pune), for her guidance and constant supervision as well as for providing necessary information regarding the project work and also for their support in completing the paper work. I also thank to my institute for giving permission for presenting my project work.

REFERENCES


