International Journal of Engineering & Technology, 7 (4) (2018) 5690-5698

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.22272
Research paper

COABRA: collaborative secured auditing with privacy preservation policy on group shared data in cloud based digital repositories

Arunkumar. S. [1]*, Anbarasi M.S.[2]

[1]*Research Scholar, Department of Computer Science and Engineering, Pondicherry Engineering College, Pondicherry University, Puducherry, India 605014*
[2]*Assistant Professor, Department of Information Technology, Pondicherry Engineering College, Pondicherry University, Puducherry, India 605014*
**Corresponding author E-mail:arunatr85@gmail.com*

Abstract

Cloud storage services like digital repositories, allow users to store and share data across multiple users within a group. However it is always a big time challenge to conserve the integrity of the data stored, due to the high probability of inevitable hardware and software failures. The Third Party Auditor (TPA) is allowed to verify the integrity of the data ie checking correctness of the data. There are many challenges during auditing process, mainly privacy issue. Another prevailing challenge is to achieve high degree of access control and traceability along with extensive abstraction to preserve privacy.This proposal provides solution for this privacy preserving issue by Holomorphic Circle Sign (HCS) method without revealing the identity of signer where the signer is one of the group members. Also provides Random Combination Block (RCB) method to audit random combination of encoded data rather than actual data. RCB method provides reduced data transmission and avoids decoding/recalculation process.The proposal also support public auditing on regenerated data which is used for providing reliability on shared data and prompt traceability mechanism that tracks.

Keywords:*Cloud Audits; Cloud Storage Services; Homomorphic Signature; Encode.*

1. Introduction

Due to the improved sophistication and user friendly ubiquitous access of stored data, cloud services have emerged as an ultimate destination for all type of users ranging from private to organizational, for storing and sharing their data [1]. There are various cloud hosting services like SkyDrive, rapid share, etc., which allow users to upload, share and access data pervasively. Considering all such inherent features of cloud, we could use it as a composite repository to store all our private data and access it ubiquitously. Especially when we consider the burden of submitting piles of hardcopies of documents at various Government and non-government workstations for verification purposes, we can realize that we could utilize cloud services to create a private repository to store and share all such documents with great ease. We title this idea as "Digital repository" in this paper and illustrate various related problems and corresponding solutions. The Digital repository is analogous to a typical physical private locker but would be a virtual private storage space given to a user to upload and manage all his personal data and officials for verification and validation purposes. A unique user ID would be required to access the repository.

To envision the digital repository as a group shared cloud, let's consider the repository is maintained and accessed by various users (u1, u2….un) ∈ G where n represents the number of users within the group G. if we reshape the digital repository for an organization to store their group shared data like organizational documents, bidding quotations, employee database, etc., the repository would be accessed by various departments within the group for various purposes. All these users have to be provided with appropriate levels of accesses. While this becomes a composite solution for storing all significant data; maintaining the integrity, privacy, confidentiality of the data stored needs keen attention [12], [14].

Integrity refers to assuring the intactness of the data stored, durably, until a user makes changes to them. There are various challenges in assuring the integrity due to unavoidable system and human errors. One of the prevalent solutions is to arrange a Third Party Auditor (TPA) or public verifier [2 - 4] to carry out audit delegations and report the status of integrity to the data owner or group manager in case shared data[5]. In casehowever, this solution provokes privacy issues caused by certain third party verifiers who try to access the information while scrutinizing the same.

Certain existing solutions allow the auditors to verify random blocks [6] of data instead of extracting the entire data during auditing. In this case, the blocks need to be protected by appropriate mechanisms of encryption and signature to maintain data confidentiality. We need to note that there is a risk of identity privacy and data privacy leakage [7] to the auditors who can use the same to manipulate various information regarding the audited data. Hence, a group signature solution [8], [9] was proposed where all the blocks are signed by aring signature, σ'm to preserve identity and data privacies. But verifier should download entire block for verification. It leads to increase in communication cost.

Another big challenge in ring signature is tracing the signer log record because any user can generate the ring signature without knowledge of the group users. Misfeasor may misuse shared data

by generating ring sign without getting the proper permission from group manager. So we need efficient tractability system to protect data from misfeasor.

To accomplish this, the hosting organization or group managers need an monitoring mechanism to let them beware of what is happening on their services. When traceability and accountability [10], [11] of users and their actions on a group shared data are considered, the group manager should be able to instantly announce the users and their activities whenever a dispute occurs on accountability of an action performed over the data. An appropriate logging solution needs to be incorporated for assuring the same [12].

The remainder of this paper is organized as follows:Section 2 overviews the related works. In Section 3, some preliminaries and cryptographic primitives are reviewed. In Section 4, we describe threat analysis, design goals and our system model. In Section 5, the proposed scheme is presented in detail. Sections 6 we evaluate the performance of the system. Finally, we conclude the paper in Section 7.

2. Related works

In PDP [6], the first public auditing solution, a public verifier would pick random block mi' from the uploaded block of data (m_1, $m_2 \ldots m_n$) and verify the integrity by set of challenge and responses between TPA and cloud as shown in Fig. 1. However, the system suffers from serious issues of user and data privacy leakage during auditing because block mi is signed by private key of any one of the user u. obviously verifier uses public key of user u for verification. So verifier may know about access frequencies of users on particular block. It may lead to identify the sensitive block among several block and main role player among group users.

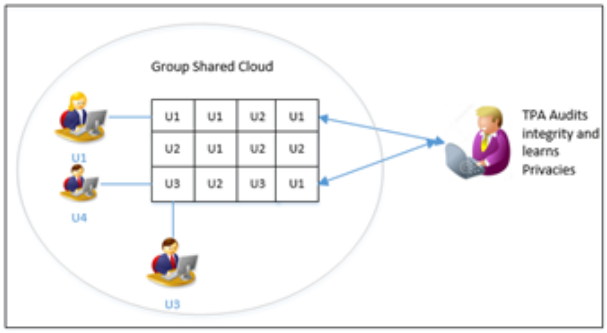

Fig. 1: Conventional Public Auditing Solution.

Many [7][14] Homomorphic authentication based techniques are used to check the correctness of data without retrieving entire-block. Homomorphic authentication also [15] is used to assure erasure codes-based data distributed on multiple servers and its correctness. This precisely provides a strategy for dynamic data. Certain proposals introduced a mechanism for auditing the correctness of data under the multi-server scenario, by network coding rather than erasure codes [21]. Few solutions proposed alternatives to avoid high decoding computation cost and save computation resource for online data owners [22].but all the above mentioned homomorphic authentication concentrated on private data rather than public or Group shared data(ie. multi owner data).

A privacy-preserving Storage and Retrieval [17] was proposed to store and retrieve the data from / to cloud with privacy preservation policy. It protected the identity of user while accessing the data by other user. But it is inefficient for public auditing. The consecutive solutions of PDP[6] to overcome the above privacy issues [2] [3], [9], [13] [18],proposed group signatures and global encryption of all data blocks with such as global signature group signature. However, It is single point of failure and some of members in group Some of group member would not like to trust others for signing on their data. And the solution failed to provide a proper user revocation. and demanded, re-computation and re-distribution of the group signature and encryption keys, to all the users in the group resulting in overhead on the system and risk to the system leaving it accessible to the revoked user until their access is revoked and their data is re-encrypted.

Panda proposed a precise solution for user revocation by extended proxy re-signature strategy [19]. However, it suffered from identity privacy and data privacy issues if used as a public auditing solution since the data blocks had the original signatures of users as such.

Traceability of users' activities were not given much concern in the prevailing audit solutions, while it is imperative to accomplish proper traceability and accountability checks to trace users' activities whenever an intra-group conflict occurs upon anomalous activities on the shared data. NPP [20] was discussed about tractability issue and solution through tree based data structure and version control. But it required large storage and costly infrastructure. The work [12] explored much on this aspect and suggested certainsolutions of implementing logging mechanisms that could be employed to trace the activities of users and data owners.

In a gist following are the existing problems and our corresponding contributions to resolve them, in this paper:

1) The data stored in cloud services are vulnerable to loss of integrity during inevitable system failures and human errors. Existing systems provided solutions of periodical auditing of data but suffered from data and identity privacy issues during auditing, which would be handled in the proposed solution diplomatically with advanced signatures and strategic encryptions, ensuring secured auditing.
2) Few existing auditing solutions [5], proposed mechanisms to preserve integrity and identity of data and users, but lagged in providing a proper user revocation mechanism. So need a solution for handling user revocations promptly by an advanced revocation strategy.
3) On the other hand, actions performed on the data by users and have to be tracked keenly to ensure traceability and accountability. The proposed solution provides a tenacious logging mechanism to ensure the same. This log is not visible to TPA and the log can be maintained and audited only by Group Manager or internal user under the control of Group Manager.

3. Problem statement

In this section we would discuss about various existing threats to cloud storage services, our design goals and the system model provides an overview of the proposed solution.

3.1. Threat analysis

a) Integrity Threats

Integrity of data might be affected in various ways by a third party or a user within the group, or due to unavoidable hardware failures and software corruptions.

Sadly, the hosting organizations do not let the data owner know about the integrity loss, to maintain their reputation.

b) Privacy Threats

While a TPA verifies the integrity of data stored by extracting random blocks of data, we need to note that the TPA would be able to learn the identity of the users and data by the signatures being used on the blocks.

For example, let us consider that the digital repository is used to store the bidding quotations of a manufacturing company. During an audit if the verifier is not a loyal person and learns the signature a pattern analysis if data could be carried out as follows:

1) First the TPA would learn the most significant user among the group who signs on all the files like a final. For example, a CEO of a manufacturing company verifying and signing the tender quotations.
2) 2. The verifier could now narrow down his search to the frequently signed records by the above significant user

(e.g.) might be a bidding rate being updated in the quotation documents.

In above way, both identity as well as data privacy is threatened if the system is vulnerable to such leakage when a public verifier scrutinizes the data.

c) Non-traceability and non-accountability threats

Suppose, if a plain group signature strategy is applied to all the data blocks, it assures to preserve the privacy of user, however, it invokes the risk of non-traceability and non-accountability, where the users perform certain anomalous activity on the shared data but doesn't takes up accountability of the activity. In this case the owner should be able to trace the activities of users and point out the culprit user.

3.2. Design objectives

COABRA, is designed to meet following objectives:

Privacy Preservation: A Verifier should not able to discover the identity of the user.

Less the overhead of auditing process: a verifier should able to verify the data block without downloading the entire block.

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

Traceability: A Group manager should be able to trace the activities of users and point out the culprit user.

3.3. System design

The overall proposed system model consists of three components:

1) Key and Signature Generation Component (KSGC)
2) Data Audit Component (DAC)
3) Traceability and Access control Component (TAC)

KSGC consists of the KeyGen algorithm which generates the users' private key Ski and public keys Pki during the user Ui registration. DAC consists of two algorithms. CircleSign algorithm generate Circle signature using the public key of the entire user from the group. The verifier cannot able to identify the user who signed the block, but he knows the signer is one of member in the group .DataAudit algorithm generates the audit data that is the Circle signed super blocks using homomorphic authentication property. It also verifies the integrity of data by the challenge response mechanism. The Figure 2 shows the challenge response mechanism.

Fig. 2: Audit Challenge Request & Response Mechanism.

User sends audit challenge to CSP and TPA.CSP generates super block by applying homomorpic technique. Super block is linear combination of n Circle signed blocks.

TAC is specific for tracing the activities of the users and comprises of an Access Tracker component that deals with access control strategies like capturing activities, inserting it's details into the table and trigger a mail to a data owner when any anomalous activity is being performed by a user on a data. The second component of the TAC is the TraceView that takes care of providing the tracked details from the table to the data owner or auditor with respect to the query sent to the component.

4. Preliminaries

This sections walks through certain cryptographic primitives and their properties to get a clear picture of the proposed solution.

4.1. Bilinear maps

a) Bilinear map μ is a map μ: $G_1 \times G_2 \rightarrow G_d$.

Where G1, G_2 andG_d be three multiplicative cyclic groups of prime order p. $g_1$and g_2 be a generator of $G_1$and G2 respectively. Bilinear map μ is a mapμ: $G_1 \times G_2 \rightarrow G_d$ with the following properties:

1) Computability: there exists an efficient algorithm for Computing map μ.
2) Bilinearity: for all $u \in G_1$ and $v \in G_2$ and a, b $\in Z_p$, $\mu(u^a, v^b) = \mu(u, v)^{ab}$
3) Non-degeneracy could be expressed as: $\mu(g_1, g_2) \neq 1$.

4.2. Security assumptions

The following security assumptions are considered: Computational Diffie-Hellman (CDH) Problem:

Let a, b $\in Z_p$', given g, g^a, $g^b \in G_1$ and$g_{ab} \in G_1$.

Definition 1:*Discrete Logarithm Problem.*

For a $\in$Zp, given g, h = $g^{ab} \in$ G1, output a. The Discrete Logarithm assumption holds in G1 if no polynomial time algorithm has advantage at least in solving the Discrete Logarithm problem in G1, which means it is computational infeasible to solve the Discrete Logarithm problem in G1.

Definition 2:*Computational Co-Diffie-Hellman Problem.*

For a $\in$Zp, given g_2, $g^a{}_2 \in$ G2 and h $\in$ G1, compute $h^a \in$ G1. The co-CDH assumption holds in G_1 and G_2 if no polynomialtime algorithm has advantage at least in solving the co-CDH problem in G_1 and G_2.

4.3. Ring Signature

This is a variant of Ring signatures which are generated by using public key of all the users in the group. But any TPA who verifies the signature will not be able to uncover the identity of the actual signer and hence identity privacy of the users is conserved. The ring signature [23] is constructed on bilinear maps. We will employ the concept to sign the auditing proof that would be verified by TPA.

4.4. homomorpic authenticators

Homomorphic authenticators [6], (homomorphic verifiable tags), used in public auditing mechanisms allow public verifiers to verify the integrity of data without downloading the entire data. They exhibit the following properties,

- Unforgeability: Only users in group with valid private keys can only generate a valid signature.

Let (p_k, s_k) denote the signer's public/private key pair, σ_1 denote the signature on block m1 $\in Z_p$, and σ_2 denote the signature on block $m_2 \in Z_p$.

- Blockless verifiability: Blockless verifiability refers to verifying the integrity of data without downloading the entire set of blocks but just a linear-combination of all the blocks which we refer here as super block.

Given σ_1 and σ_2, two random values v_1, v_2 in Z_p and a block m' $=v_1m_1 + v_2m_2 \in Z_p$, a verifier is able to check the correctness of block m'without knowing m1 and m2.

- Non-malleability: Non-malleability refers to the feature by which without private key, it is not possible for anyone to generate the signature on the super block by combining all the existing signatures.

Given m_1 and m_2, σ_1 and σ_2, two random values v_1, v_2 in Z_p and a audit block m'$= v_1m_1 + v_2m_2 \in Z_p$. A user, who does not have private key s_{ki}, is not able to generate a valid signature σ' on the super block m by combining σ_1 and σ_2.

Block less integrity checking allows the auditor to verify super block which is linear or random combination of n blocks. If correctness of the of super block is verified, the Verifier believes all n blocks are correct. It also reduces the communication cost because TPA no needs to download the all blocks.

5. COABRA

5.1. Overview

COABRA is COllABoRative secured Access to group shared cloud data. It is a novel system that collaborates various security strategies to accomplish a composite tenacious security mechanism to the data stored in a public cloud that are shared across a group of users. It resolves various short comings in previous works done on the same problem area.

5.2. Pre-requisites and related operations

There are few prerequisites that we need to make to the existing structures before implementing COABRA. A typical cloud hash table index [5] would contain:

m_i= message block,σ_i= signature,id_i = block identifier and t_i= signer identifier i.e. user identifier

We would leverage the hash tables to support dynamic operations efficiently without changing the block identifier of the existing blocks within the same hash table [5].

During audits the TPA would not be provided with the signer identifier and actual data blocks in order to preserve the identity privacy of the users and the respective data privacy.

For implementing the traceability mechanism, a table has to be created within the system database. The table could be created with all the necessary fields that would be required to trace the actions of the user. In our experiment we have created a table (Tracker_Table) with fields like block identifier, record identifier, file name, file owner, accessed user, type of actions performed, time of action and location of accessed system which is accessible only by Group Manager and not by TPA.

5.3. Construction of COABRA

The Fig. 3. shows the system architecture of COABRA depicting various components that forms the system. Each component is discussed as follows:

a) Key Generation Component (KGC)

The KGC component consists of two algorithms:

KeyGen algorithm performs the action of generating the encryption keys (public key and private key) for users. The keys are generated right at the time of user registration process and public key information are stored in the Key directory. The generated keys are used by COABRA to provide confidentiality and integrity services to shared data across cloud and the client system. The public key of group users is used to generate the Circle signature which is used to sign the data block. Circle sign algorithm explains how public key of all other user are used to generate the Circle signature for protecting the privacy of user.

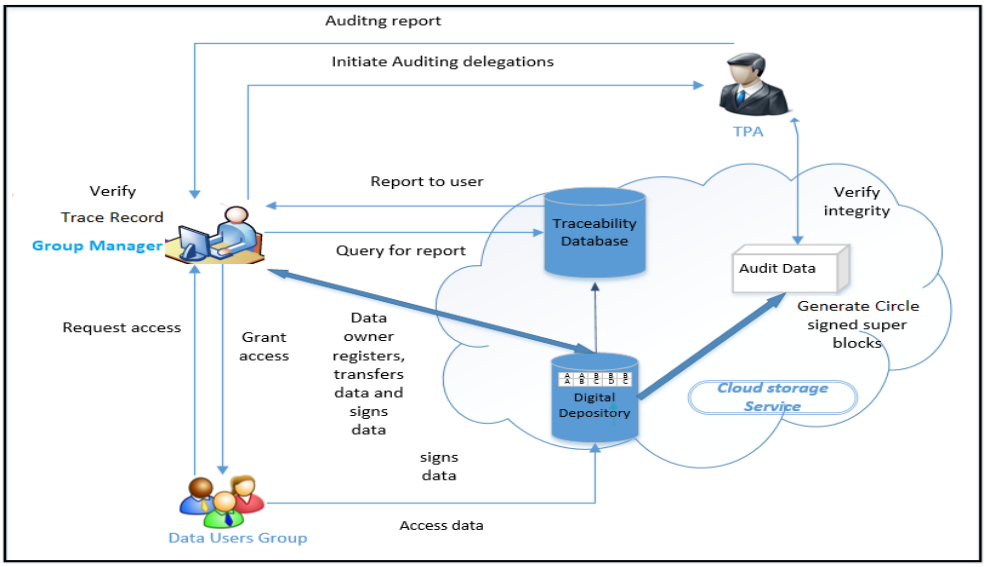

Fig. 3: COABRA System Architecture'0.

b) KeyGen Algorithm

When a user wants to register in the cloud, the required details are collected from the user and posted to the cloud. The moment the registration request reaches the cloud, KeyGen algorithm is invoked.

Let G_1, G2 and G_d be multiplicative cyclic groups of order p. g_1 and g_2 be generators of group G_1 and G_2 respectively.

- Consider the bilinear map $\mu: G_1 * G_2 \rightarrow G_d$ and a computable isomorphism $\dot{I}: G_2 \rightarrow G_1$.(ie : $\dot{I}(G_2) = G_1$)

Let us consider hash functions:

H : {0, 1}*→G1,

- The total number of users in the group be n. The global parameters (μ, İ, p, G1, G2, Gd, g1, g2, H, n).

Foruser u_i, randomly picks $s_i \in Z_p$ and computes $w_i = g_2^{si}$. $w_i$is belongs to G_2. User u_i's public key is $pk_i = w_i$, and corresponding private key is $sk_i = s_i$.

After generating the key, every user must store public key information to key directory which is used to provide public key information to all users in group.

c) Data audit component (DAC)

The DAC component consists of two algorithms one is CircleSig and DataAudit.

CircleSign that deals with generating the Circle signature on the audit block. The audit block is generated by generating a hash on a

linear set of blocks that form a complete data. Then the generated Circle signature is signed on data block by user and the same would be stored in the cloud. Fig no 4 shows Circle signature concept.

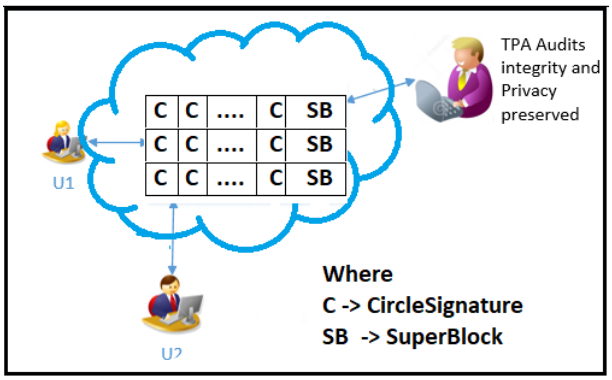

Fig. 4: Circle Signature and Super Block Creation Concepts.

U_1 and U_2 are two users in group. After accessing the data, users generates circle signature C instead of their own signature. The CircleSign explains how circle signature will be generated. This action is done at the time of every access on data block by the any group user. The audit data is generated as a metadata and stored into the database against the original data or block identifier so that the same could be provided to the TPA during audit.
The CircleSign algorithm functions as follows:

CircleSign Algorithm

Given n group member's public keys (pk_1,..,pk_n) = (w_1,..,w_n), an audit block m ∈ Z_p , the identifier of that block id .Circle signatureis generatedas follows:

1) User u_t(called as signer) where 1<= t <= n , picks random no r_i from Z_p for all remaining users (ie $i \neq t$), where value of i is from 1 to n
2) Consider $\sigma_i = g_1^{r_i} \in G_1$
3) Signercomputes

$$\varphi = H(id, pk_1, pk_2, ...pk_n) g_1^m \in G1 \quad (2)$$

4) Then fixes,

$$\sigma_i = \left(\frac{\varphi}{H\left(\prod_{i=1}^{n} \omega_i\right)} \right)^{\omega_i} \in G1 \quad (2)$$

The Circle signature of an audit block mis:

σ = (σ_1, σ_2, .. σ_n)∈G_1

In Circle sigantureσ, σ_t is one of elements. So TPA can verify the signataure is signed by one of the authorised group user, butcannot learn the identity of the perticular user.
Next algorithm is DataAudit provides functions for the public verifier to carry out auditing.
DataAudit Algorithm
When TPA recieves a auditing challenge from group manager, TPA forwards same to cloud storge. Mean while cloud also recevices same copy of challange from group manager. The group manager can allow the TPA to verfify the data. Eventhough Blockless verication method is used in this proposal, both Block level andBlockless verification methods are disscussed here for comparision purpose.HCS is used in both verification methods.
Block level verfivication:
In this method ,TPA downloadse all the data blocks and meta data of the corsponding data block from cloud. Then checks the correctness of the data block one by one. Verification process of block level is follows:
Verifier recieves the following information from group manager for verification process.Block identifier of m id, public key of all users (pk_1... pkn) = (w_1... wn) and Circle signature

$$\phi = H(id_1, pk_1, pk_2, ... pk_n).g_1^m \in G_1$$

σ = (σ_1, σ_2, .. σ_n).

TPA calculates

$$\mu(\varphi, g_2) = \prod_{i=1}^{n} \mu(\sigma_i, \omega_i). \quad (3)$$

Then checks

If equation no 3 holds, TPA also belives that block m is signed by one of user from group. Else it is not.
BlockLess Verification:
In this scenario, Cloud Service Provideris prepraring super block which is linear combination of randomlyselected blocks from multi cloud storage server.
Consider two Blocks m_1 and m_2, Circle siganture σ1= ($\sigma_{1.1}$,$\sigma_{1.2}$ … $\sigma_{1.n}$) for m_1 and σ2= ($\sigma_{2.1}$,$\sigma_{2.2}$ … $\sigma_{2.n}$)for m_2, Identifiers of m_1,m_2 are respectivly id_1,id_2 and two random values v_1,v_2∈ Z_p.
Super block m' is combined by cloud using following formula

$$m' = v_1 m_1 + v_2 m_2 \in Z_p$$

TPA can verify m' without downloading the m1 amd m2 by verifying

$$\mu(\phi', g_2) = \prod_{i=1}^{n} \mu(\sigma_i', \omega_i). \quad (4)$$

Where

$$\varphi' = H(id_1, pk_1, ...pk_n)^{v_1} ..H(id_2, pk_1, ...pk_n)^{v_2} .g_1^{m'}$$

$$\sigma_i' = \left(\sigma_{1.i}^{v_1} .\sigma_{2.i}^{v_2}\right)$$

If super block m' is correct, then TPA trusts thatm1 and m2 are correct. Figure.4 shows super block creating process.This process is not only protect the privacyalso reduces the communication cost while auditing process
In the above way without revealing any single useful detail to the TPA the integrity of the data is verified correctly.
Traceability and Access control Component (TAC)
This component takes care of capturing the events that are performed on the group data by users. The captured data are stored in the database tables [11 - 12] and appropriate access control are imposed to control the access to users when they try to access a group shared data.
As per discussion in introduction chapter, tracing the user action is another big challenge in Circle signature mechanism. TPA unable to identify the signer, which can preserve privacy from auditor. But it does not supporting tractability. Group manager must know who all are made modification on shared data. Else it leads misfeasor's attack, which means legitimate insider doing unauthorized access on shared without reveal his identity. So Group manager need efficient token based log system to maintain log of every modification action.
The procedure of Token based log system is follows
When any user wants to modify the data block, he / she requests token from Group Manager U_{Gm}. After modification done, user ui must sign on block m_j using Circle signature. The Access controller AC checks the sign value of current version and previous version (see: Access Tracker Algorithm) .If both are same; there is no modification on the data block. Else it asks token from user ui.

Then token and user identity $u_i$id will be verified by group manager. If it is valid token, then user allow uploading the modification data block. RDBMS based tracing AccessTracker subroutine called automatically for inserting log record. Transitions among the Ui,U_{Gm} and AC are following

$$1) U_i \overset{U_i id, m_j id}{\Rightarrow} U_{Gm}$$

$$2) U_{Gm} \overset{Token}{\Rightarrow} U_i$$

$$3) AC \overset{Token, U_i id, m_j id}{\Rightarrow} U_{Gm}$$

$$4) U_{Gm} \overset{Grants}{\Rightarrow} AC$$

Where

$$Token = (Tk, Ts, Lt, U_i id, m_j id)$$

Tk: TokenNo, Ts:TimeStamp, Lt:LifeTime, U_i id :Identifier of i^{th} user, and $m_j$id: Identifier of j^{th} Data Block

Once AC gets grant signal from Group manager, AC begins two functions,

Uploading the modified block and Meta information whichis used for Privacy preservation external auditing by TPAon to cloud.

Creating Log Record and inserting record into logDatadaseTable_Tracker

The algorithm requires a table to log and maintain the activities that are carried out by users on the data shared within a group. The algorithm uses the table to manage the access requests from a user to access a block for which the data's owner is yet to grant permission to the requesting user.

The overall module is split into two components:

- AccessTracker which takes care of inserting a record about modification action that was performed on a data stored in the cloud. For example, when a user tries to edit a data that is shared among his group, the activity would be logged into the table as shown in the algorithm. On the other hand, if a user tries to edit a document for which he doesn't have the permissions from the group manager, an entry is made into the table with the appropriate request status and triggers a mail to the data owner. The user cannot perform any action on the file until the group manager grants the access. The access request status is always checked by the component whenever a user tries to edit a data block. If there is no existing pending request for a particular user on a given data, a new request in posted to the group manager.
- TraceView component handles a view report request from a user to check the activities performed on the user's data. The component takes the user name from the session and date range from user as inputs and displays the logged activities and violations. The report could be extracted by the user if required.

The algorithms are as follows,

AccessTracker Algorithm

```
AccessTracker(Hval_cur,HVal_Pre,Token)
{
Begin
Open connection to DB
If[ Signature value of previous version =!current version]
 If [ Is token valid]
{Allow_upload()
Insert into Tracker_table values (user_id, data_id,dataowner_id, time_of_access,
'data uploaded', loc_system)
}
 Else
{
Insert into Tracker_table values (user_id, data_id, dataowner_id, time_of_access,' invalid token', loc_system)
}
 End If;
End If:
```

For a TrackView Algorithm, the user inputs the preferred date range and the user's password. The user name is extracted from the session value. These inputs are fed into the algorithm and it outputs all the tracked detail from database for the specified date range with respect to the user id.

If the Group manager wishes to download the record, it could be done by an export function. Group user can audit the user activity through this record in case any insider attack. Note, this internal audit process supports only tracing user activity For checking integrity, We preferred external auditing which discussed in Circle sign and verification algorithm. By separating auditing process as internal and external auditing, COABRA design supports both privacy preservation and traceability.

Hence by using RDBMS, to store the tracked data we can optimize the storage space required to store the tracking information. At the same time by incorporating strong access control checks as part of the access tracking algorithm, we have tighten up the security of the data stored and accessed on the group shared cloud.

6. Security analysis of COABRA

6.1. Security analysis for audit

Statement 1: An auditor can correctly ascertain the integrity of stored data by COABRA.

Proof: In COABRA, the integrity of a given data is agreed to be intact only if equation (3) is satisfied. Hence, we need to prove the validity of equation (3)

Given the properties of bilinear maps,

The final audit proving equation of COABRA is below:

$$\mu(\varphi, g_2) = \prod_{i=1}^{n} \mu(\sigma_i, \omega_i) \quad (3)$$

Let us expand the LHS to prove the correct verifiability,

$$\prod_{i=1}^{n} \mu(\sigma_i, \omega_i) = \mu(\sigma_t, \omega_t) \cdot \prod_{i \neq t} \mu(\sigma_i, \omega_i)$$

$$= \mu\left(\left(\frac{\varphi}{\tilde{1}(\prod_{i=1}^{n,(i \neq t)} \omega_i^{r_i})}\right)^{1/r_t}, g_2^{r_t}\right) \cdot \prod_{i \neq t} \mu(g_1^{a_i}, g_2^{r_i})$$

$$= \mu\left(\left(\frac{\varphi}{\tilde{1}(\prod_{i=1}^{n,(i \neq t)} g_2^{r_i a_i})}\right), g_2\right) \cdot \prod_{i \neq t} \mu(g_1^{a_i r_i}, g_2)$$

$$= \mu\left(\left(\frac{\varphi}{(\prod_{i=1}^{n} g_1^{a_i r_i})}\right), g_2\right) \cdot \prod_{i \neq t} \mu\left(g_1^{a_i r_i}, g_2\right)$$

$$= \mu(\varphi, g_2)$$

RHS=LHS. Hence proved.

From the above we could prove that the auditor can correctly ascertain the integrity of data.

Statement 2: The cloud re-signature scheme is Homomorphic authenticable.

Proof: To prove this we need to show that the cloud re-signatures are blockless verifiable i.e without knowing the blocks anyone can verify its integrity and non-malleable i.e no one can forge the signature by combining the available signatures.

Blockless Verifiable: Given a user U_i's public key P_K^A, two random values v_1 and $v_2 \in Z_p$, the block id's id_1 and id_2, and their corresponding signatures σ_1 and σ_2, for a block $m'=m_1v_1+m_2v_2$ anyone can ascertain the integrity of the block by simply computing:

$$\mu(\varphi', g_2) = \prod_{i=1}^{n} \mu(\sigma'_i, \omega_i) \quad (4)$$

Hence, we need to prove the validity of equation (4)
Considering the properties of bilinear map, from the above equation we can say that under COABRA, blockless verifiability is possible.
Let us expand the LHS of equation no 4

$$= \mu(H(id_1, pk_1, ...pk_n)^{v_1}.H(id_2, pk_1, ...pk_n)^{v_2}.g_1^{m'}, g_2) ...(5)$$

Let us expand the RHS of equation no 4

$$= \prod \mu(\sigma_{1,i}^{v_1}.\sigma_{2,i}^{v_2}, \omega_i) \quad (6)$$

Hence, we need to prove the equation (4) through (5) & (6)
Let us take equation no 5

$$= \mu(H(id_1, pk_1, ...pk_n)^{v_1}.H(id_2, pk_1, ...pk_n)^{v_2}.g_1^{m'}, g_2)$$

$$= \mu(H(id_1, pk_1, ...pk_n)^{v_1}.g_1^{v_1 m_1}, g_2).\mu(H(id_2, pk_1, ...pk_n)^{v_2}.g_1^{v_2 m_2}, g_2)$$

$$= \mu(\varphi_1, g_2)^{v_1}.\mu(\varphi_2, g_2)^{v_2}.$$

$$= \prod_{i=1}^{n} \mu(\sigma_{1,i}, \omega_i)^{v_1}.\prod_{i=1}^{n} \mu(\sigma_{2,i}, \omega_i)^{v_2}.$$

$$= \prod_{i=1}^{n} \mu(\sigma_{1,i}^{v_1}, \sigma_{2,i}^{v_2} \omega_i)$$

Hence proved. Circle Signature supports block less verification
Non-Malleability: Given the signatures σ_1 and σ_2 and the blocks m1 and m2, no one could generate the cloud re-signature until and unless he/she knows the user's private key Sk. The strength of this is mainly due to the one way hash function such that with m1 and m2 given to a hash function the result m' could be generated. However, from the given hash value, a user may not be able to determine the value. Hence, proving that the cloud signature and the data are non-malleable in nature. We also proved that cloud re-signature scheme is homomorphic authenticable.

6.2. Security analysis for traceability component

1) Man-in-the-Middle Attack

The data stored within the database are accessible only from the user's valid session. Periodic session expiration are enabled to make sure that the user views the data about the access only when his session is active. Moreover the data transferred across, are encrypted by strong encryption algorithms with the user's private key and hence the data could be decrypted only if the attacker is aware of the private key which is infeasible.

2) Authenticity and Confidentiality Issues

The access to the tracking system is authenticated and authorized only if the user has a valid login credentials and token. Moreover the user name and password are obtained from the user along with the input time ranges and hence unauthorized access is not possible. In addition to this if necessary system level login and monitoring are enabled by the network security manager of the cloud system, any unauthorized access could be immediately blocked.
The reports that are extracted remain confidential as only a valid user or group manager whose credentials are valid can access the tracking system. Moreover, the extracted report are encrypted with user's private key and password and the same is required to decrypt the data. Hence, the data transferred across remains confidential.

7. Performance evaluation of COABRA

In this section we evaluate the performance of COARBA in various aspects like auditing efficiency, privacy parameter and the performance of the logging mechanism.
For experimental analysis the proposed system was designed in Visual Studio 2010 as a ASP.NET web application with C# as the programming language. The system used Windows 7 operating system with configuration of Intel Core I5, 2.5 GHz Processor and 4 GB RAM. The data used varied in size from 10KB to 2 MB. A total of 2 GB data were used to perform the analysis in various dimensions. The ASP.NET C# namespace System.Security.Cryptography was utilized to write the encryption, decryption and Hash generation. The classes under the namespace System. Diagnostics has been used to record and analyze the performance of the system.

7.1. Performance analysis of privacy parameter

Consider the shared data of Group which consists of n number of member. As per this proposal, any user can generate the homomorphic oriented Circle signature (HCS) .the probability that TPA succeed to discover the identity of signer is 1/n. So failure probability is 1- (1/n). The following table consists of some n value and its Privacy probability and relation between probability of privacy parameter and number of members in group is depicted in figure 5.

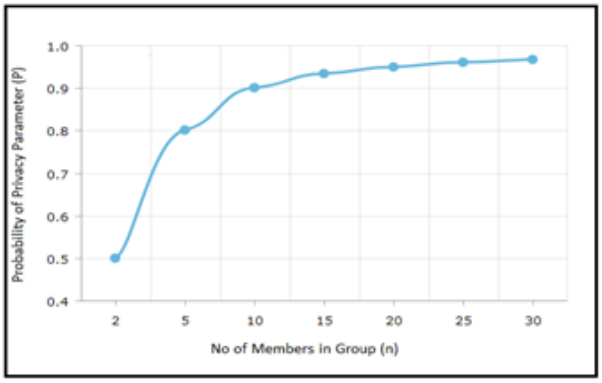

Fig. 5: Relation between probability of Privacy Parameter and number of Members in Group

Table 1: Probability of Privacy Parameter

S.no	No of user in the Group	Probability of Privacy parameter(1-1/n)
1	2	0.500
2	5	0.800
3	10	0.900
4	15	0.933
5	20	0.950
6	25	0.960
7	30	0.967

7.2. Performance analysis of verification time

7.3. Batch level vs. batch less verification

As per above mentioned configured system, verifier required 0.5 ms for checking the integrity of single block. File with various no of blocks and block level verification time are recorded in table 2. Let us consider that super block consists of 50 linearly combining blocks for block less verification. CSP must spent time for making super block. Time taken for same process on 50 blocks is 4 ms. By increasing no of blocks for making super block, we may reduce some quantity of verification time.Therefore Block less verification time isno of super blocks in file * time for single super block

creation + verification time for super blocks. Measured value is listed in following table. Figure .6 shows comparison results between block level and block less verification with privacy preservation policy.

Table 2: Block Level vs. Block less Verification Time

S.no	No of blocks in file	Verification Time(ms)	
		Block level	Block Less
1	100	50	8.5
2	200	100	16.5
3	300	150	24.5
4	400	200	32.5
5	500	250	40.5

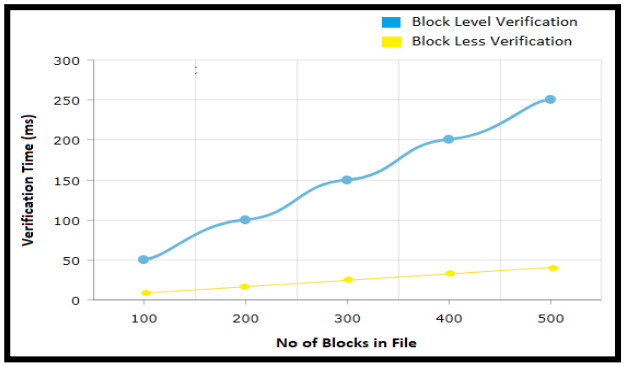

Fig. 6: Relation between Block Level Block less Verification.

7.4. Logging of view/upload/download actions

The average time taken to write a particular user action into the table is measured for evaluating the performance. The average time to insert a action that involved viewing or saving an edited data is about 8 ms(milliseconds) andthe average time taken for uploading/downloading a file is 10ms, irrespective of the file size or the group size. The time differences are due to the number of parameters inserted for different actions

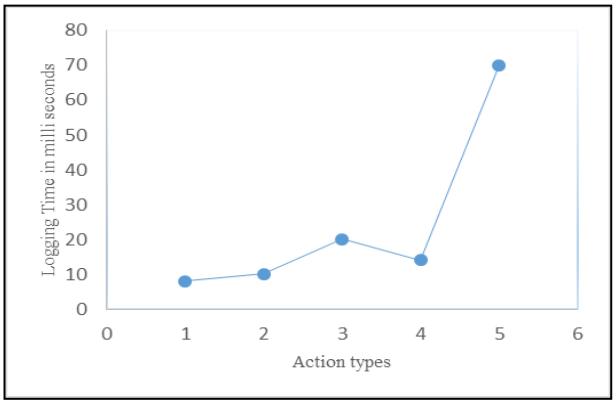

Fig. 7: Time for Logging Data into Database for Different Actions as in Table 3.

Table 3: Different Actions and Time Taken to Log

Action ID	Action	Time taken in ms
1	View/Edit	8
2	Upload/Download	10
3	Grant/Allow edit	20
4	Waiting/Deny	14
5	Fresh Request/mail	70

7.5. Logging of access evaluation and edit actions

When a user tries to edit a file, the time taken for evaluating the access permission of the user over the file, allowing or denying the user to edit the file and logging the action are measured. This has various scenarios and the time taken by the component differs for each action as follows:

1) If it is a fresh request from the user to edit the file:

The time taken to mail the owner and insert the action intothe table is about 70ms.

2) If the user's request is still pending or has been denied by the owner:

The time taken to evaluate the request status, report the status to the user and log the action takes 14 ms perrequest.

3) If the user's access is granted by the owner:

The time taken to evaluate the request status, invoke the edit module, change the status of the request and log the action into the table takes 20 ms per request.

The results are recorded in the Table 3 and depicted in Fig. 7.

Performance Evaluation of Auditing component

The performance of auditing data, is evaluated by carrying out a mock audit using COABRA with various sizes of data. From the analysis performed, it is observed that the time taken to generate the audit data and to perform the audit differs with the size of the data.

The fig.no.8 depicts how the audit performance differs with the size of the audit data. One could see that with the increase in size of data, the time for performing the audit also increases linearly. This is because the time for generating hash and, signing and encrypting the super block all takes some time. While performing audit all these functions are run again to determine the existing integrity of a given data and the value is compared to the value stored at the time of uploading or modification of the data.

Apart from the size of the file with the increase in size of the group, which apparently leads to more number of shared data, might increase the time to complete the audit. However, the time taken to complete the audit still depends on the size of the file as we have all other inputs ready at the time of audit. Also, the data transfer rate plays a key role in performance, which is common to any network based communication system. The experiment was carried out under a broad band network with a bandwidth of 2MB per second.

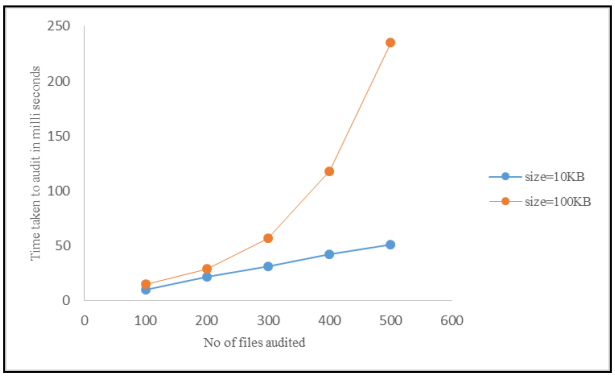

Fig. 8: Performance Evaluation of Auditing.

On the whole, from the experimental results, COABRA justifies to perform well as a composite secured system along with a precise and fast auditing solution.

8. Conclusion and future works

COABRA resolves various short comings of the prevailing solutions for group shared cloud storage services. COABRA provides a composite auditing solution to ensure the integrity of the data stored in the cloud, preserving the data and user privacy with minimal communication cost. On the other hand, it also provides traceability mechanism for providing utmost security in all the possible dimensions. Thus, COABRA proves to be a collaborative secured system that addresses solution for various security and privacy issues of a typical group shared cloud storage service with reduced communication cost.

Though, the proposed system proves to be a strong security mechanism from the application and system level, overhead to group manager for internal maintenance of log and auditing for traceability will be slightly high compared with existing systems. But this

can be compromised with reduced communication cost. Thus, this proposal is useful for (i) The organization/user who is ready to manage group activities internally with strong privacy to its end user (ii) The Cloud service provider to provide service to organization/user with reduced cost. Another important issue to be solved that is user revocation policy which is out of scope of CO-ABRA's proposal. There are several solution are discussed in various research paper. That solution could be extended to support dynamic group. Another interesting our future work is making the well-organized data structure for auditing whole organization in single shot which will be reduce huge amount of verification time comparatively other systems such as linearly combining blocks, hash tree ,etc.

References

[1] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A.,Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: A View of Cloud Computing. Communications of the ACM 53(4), 50–58 (2010).https://doi.org/10.1145/1721654.1721672.

[2] 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.

[3] Wang, B., Li, B., Li, H.: Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud. Tech. rep., University of Toronto (2012).

[4] Hao, Z., Zhong, S., Yu, N.: A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability. IEEE Transactions on Knowledge and Data Engineering 23(9), 1432–1437 (2011).https://doi.org/10.1109/TKDE.2011.62.

[5] X. Liu, Y. Zhang, B. Wang, and J. Yan, "Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud," IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 6, pp. 1182-1191, June 2013.https://doi.org/10.1109/TPDS.2012.331.

[6] 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.https://doi.org/10.1145/1315245.1315318.

[7] Wang, Q., Wang, C., Li, J., Ren, K., Lou, W.: Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing. In: Backes, M., Ning, P. (eds.) ESORICS 2009. LNCS, vol. 5789, pp. 355–370. Springer, Heidelberg (2009)https://doi.org/10.1007/978-3-642-04444-1_22.

[8] R. L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," in Proc. ASIACRYPT. Springer-Verlag, 2001, pp. 552–565.https://doi.org/10.1007/3-540-45682-1_32.

[9] D. Chaum and E. van Heyst, "Group Signatures," Proc. Int'l Conf.Theory and Applications of Cryptographic Techniques (EUROCRYPT), pp. 257-265, 1991.https://doi.org/10.1007/3-540-46416-6_22.

[10] R. Corin, S. Etalle, J.I. den Hartog, G. Lenzini, and I. Staicu, "ALogic for Auditing Accountability in Decentralized Systems,"Proc. IFIP TC1 WG1.7 Workshop Formal Aspects in Security and Trust,pp. 187-201, 2005.

[11] B. Crispo and G. Ruffo, "Reasoning about Accountability within Delegation," Proc. Third Int'l Conf. Information and Comm. Security (ICICS), pp. 251-260, 2001.https://doi.org/10.1007/3-540-45600-7_29.

[12] Smitha Sundareswaran, Anna C. Squicciarini, Member, IEEE, and Dan Lin"Ensuring Distributed Accountability for Data Sharing in the Cloud",Dependable and Secure Computing, IEEE TransactionsVolume:9,Issue:4, pp 556-56811 May 2012.

[13] ManjurKolhar, Mosleh M. Abu-Alhaj, Saied M. Abd El-atty," Cloud Data Auditing Techniques with a Focus on Privacy and Security" IEEE Security & Privacy, Volume: 15 Issue: 1 Pno.1-10,2017

[14] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.https://doi.org/10.1109/MIC.2012.14.

[15] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud Data Protection for the Masses," Computer, vol. 45, no. 1, pp. 39-45, 2012.https://doi.org/10.1109/MC.2012.1.

[16] 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. 1-9, 2009.https://doi.org/10.1109/IWQoS.2009.5201385.

[17] JingweiLi, Dan Lin, Anna CinziaSquicciarini , Jin Li , Chunfu Jia" Towards Privacy-Preserving Storage and Retrieval in Multiple Clouds" IEEE Transactions on Cloud Computing .Volume: 5 Issue: 3 . pg.no 499 –509, 2017.

[18] "Oruta, Privacy-Preserving Public Auditing for Shared Data in the Cloud" Boyang Wang, Baochun Li, Hui Li,IEEE cloud computing Volume:2,Issue:1, 43 - 56,13 January 2014.

[19] "Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud" Boyang Wang, Baochun Li, Member, IEEE, and Hui Li, Member IEEE, IEEE Transactions Volume:8, Issue:1, 43 - 56,03 Feb 2015.

[20] Anmin Fu, Member, IEEE, Shui Yu, Senior Member, IEEE, Yuqing Zhang, Huaqun Wang, and Chanying Huang "NPP: A New Privacy-Aware Public Auditing Scheme for Cloud Data Sharing with Group Users "DOI 10.1109/TBDATA.2017.2701347, IEEE Transactions on Big Data .

[21] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote Data Checking for Network Coding-Based Distributed Storage Systems," Proc. ACM Workshop.

[22] 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.

[23] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT'03), pp. 416-432, 2003.https://doi.org/10.1007/3-540-39200-9_26.

[24] M. Blaze, G. Bleumer, and M. Strauss, "Divertible Protocols and Atomic Proxy Cryptography," in the Proceedings of EUROCRYPT 98. Springer-Verlag, 1998, pp.127–144.https://doi.org/10.1007/BFb0054122.

[25] Details of System.Diagnostics Namespace: https://msdn.microsoft.com/en-us/library/gg145030(v=vs.110).aspx

[26] A. Juels and B.S. Kaliski, "PORs: Proofs of etrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS'07), pp. 584-597, 2007.https://doi.org/10.1145/1315245.1315317.

[27] D.J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigen- baum, J. Hendler, and G.J. Sussman, "Information accountability," Comm. ACM, vol. 51, no. 6, pp. 82-87, 2008.https://doi.org/10.1145/1349026.1349043.