Efficient and Secure Group based Collusion Resistant Public Auditing Scheme for Cloud Storage

Smita Chaudhari[1]*, Gandharba Swain[2]
Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation, Vaddeswaram-522502
Guntur, Andhra Pradesh, India

Abstract—**Tremendous changes have been seen in the arena of cloud computing from previous years. Many organizations share their data or files on cloud servers to avoid infrastructure and maintenance costs. Employees from different departments create their specific groups and share sensitive information among group members. Revoked users from the group may try to access this information by colluding with an untrusted cloud server. Many researchers have specified revocation procedures using re-signature, proxy-re-signature concept to deflect the collusion between the cloud server and a revoked user. But these techniques are costly in terms of communication overhead and verification cost if combined with auditing techniques to prove the integrity of outsourced data on the cloud server. To reduce this cost, a collusion resistant public auditing scheme with group member revocation is proposed in this paper. In this scheme, the data owner regularly updates the recent valid members list which is used by a third-party auditor to validate the signature so that collusion can be avoided. To verify the integrity of outsourced data, proposed scheme uses one of the modern cryptographic technique indistinguishability obfuscation combined with a one-way function which can reduce the verification time significantly. Experimental results show that the proposed scheme decreases the communication overhead and verification cost compared to existing schemes.**

Keywords—Public auditing; collusion attack; ring signature; message authentication code; indistinguishability obfuscation; dynamic data

I. Introduction

With the rapid growth of data, many organizations or even individuals has started outsourcing data at cloud storage. Outsourcing data at remote places reduces the burden of storage management at a local site as well as the infrastructure and maintenance costs of an organization. But this cloud paradigm has brought with it many new challenges related to security. Since data may be stored at different remote servers, cloud users are not having ownership of their own data. Data integrity at an untrusted Cloud Server (CS) is a major security concern [1]. Outsourced data may intentionally or accidentally be deleted at remote sites. CS may hide such incidents from users to maintain reputation. Periodic verification may consume resources and create a burden on the user side. To get rid of this, the cloud user delegates this verification responsibility to a Third-Party Auditor (TPA) who is a professional and having the capability to check the integrity of the outsourced data periodically on behalf of the user. Public auditing is a technique by which TPA can check the correctness of data without copying the entire data file at its end.

Sharing services such as Dropbox and Google Drive are widely used by cloud users to share the data with multiple members in the group. Group Manager creates a group with multiple users. The group manager or any group user uploads shared data or files which can be retrieved or edited by all the group members. Before uploading a file on CS, group users need to calculate the signature to maintain and confirm the integrity of outsourced data. Whenever any user wishes to modify the data, that user has to resign that modified blocks. When users left the group, the revocation must be done properly so that revoked users are no longer able to access information from the group. The signature of blocks computed by revoked users must be recomputed by existing users. Wang et al. [2] proposed an efficient public auditing scheme Homomorphic Authenticable Proxy-Re-Signature scheme (HAPS) with user revocation. But with this scheme information may be revealed to the revoked user because of collusion between revoked user and CS.

Security threats can be classified as *internal* and *external* threats. Many organizations concentrate only on external threats because of confidence that internal threats can be monitored by organization policies and access rights. Hence they concentrate on an unfamiliar outsider who can get unauthorized access to their information. Although it is not possible for one entity to get unauthorized access, dishonest internal and external participants may collaborate and launch a collusion attack to get sensitive data [3].

Public auditing for cloud storage system comprises Cloud users, CS, and TPA. For the efficient processing of the auditing system, many auditing schemes assume all these entities to be honest and fully trusted. But in practice, some of these entities may be dishonest and can collude with each other to generate a collusion attack. Guo et al. [4] proposed an Outsourced Dynamic Provable Data Possession (ODPDP) scheme where any one of the three participants may be dishonest or two entities may collude with each other. By using a log-audit mechanism, the scheme resists any dishonest participant or collusion. In most auditing techniques, TPA is assumed to be expert, reliable, and having capability to validate the outsourced data on behalf of cloud user. But in real life, certain TPAs are honest but curious. They may collude with CS to pass the verification of some corrupted events. Many researchers have given solution [5]-[7] to detect

*Corresponding Author

and prevent collusion from dishonest TPA using feedback method or game-theoretic analysis.

In certain situations during partial and total file loss, CS is one entity that is not trusted. CS may try to deceive users by manipulating verification tags and proving to possess correct files. CS may delete less frequently accessed user data to create space for new data. To manipulate this event, CS may collude with TPA to pass the verification and deceive the user. Many researchers [8]-[10] have given solution using pseudo-random string or pairing-based server aided verification scheme to detect and prevent collusion from malicious CS.

Cloud users many times create groups and share the contents with each other. Revoked users from group may collude with CS or TPA to get unauthorized access of sensitive information. To avoid collusion due to revoked user, it is necessary to re-sign the blocks signed by the revoked user previously or regularly update the valid user list to CS or TPA so that they can differentiate between valid and revoked user. Zhu and Jiang [11] proposed a secure anti-collusion data sharing as well as revocation scheme for a dynamic group. In this scheme, if a user is revoked, Group Manager generates a new random re-encryption key. Using this key encrypts the block and signs the message with latest timestamp. So, there is no need to re-compute and update the secret keys of other users.

Group signature is a cryptographic technique in which any group member can sign the data but the identity of signer is anonymous in generated signature. To create a confidential network among group members, Group Key Agreement (GKA) protocol is used. Rather than a common symmetric key among group members, Wu et. al[12] proposed Asymmetric Group Key Agreement (ASGKA) protocol in which public key can be used to validate signature as well as encrypt messages whereas any signature can be used to decipher the ciphertext under this public key. Many researchers [13-14] have proposed revocation techniques using ASGKA and verifier local verification to avoid collusion. However, these schemes create increased communication and computation overhead.

To overcome the overhead of computation, Hequn et.al. [15] utilized backup files for resigning after user has revoked. In this scheme, they store original as well as backup files on cloud during upload. When user is revoked, the existing user will resign on the backup file instead of original. So revoked users do not have to share their security credentials with cloud.

In previous schemes, after revocation, signature of a revoked user has to be re-calculated by the existing user. This may create computation and communication overhead on the existing user. Proxy re-signature scheme can be used in which semi-trusted proxy computes re-signature on behalf of group instead of existing user. Many revocation schemes are proposed [16-18] with proxies to convert signatures from revoked users which reduces overhead of CS and Group users. Yuan and Shucheng [29] proposed public auditing with data sharing between multiple users. They have proposed revocation technique in which constant size of integrity proof information is transmitted to verifier.

Ring Signature [19] is another variation of group signature in which user can sign messages using his own private key and the other's public key, without their consent or concern. Thokchom and Saikia [20] proposed collusion avoidance with integrity verification using ring signature approach.

To check the correctness of outsourced data or auditing, using traditional cryptographic techniques for example homomorphic authenticators, Elliptical curve cryptography, or identity-based cryptography, proof generation and verification time is a major challenging issue. Since most of these techniques are based on bilinear pairing, computation overhead leads to greater verification time. To reduce this, modern cryptography technique known as Indistinguishability Obfuscation (IO) [21] was used by many researchers [22]-[24] for public auditing. These researchers have shown that IO combined with one-way function (OWF) greatly reduces the verification time during auditing process. Sun et.al.[30] proposed auditing using IO with symmetric key. Rabaninejad et.al.[31] proposed lightweight auditing using ID-based cryptography.

The main contribution of our work is as follows:

- Zhang et.al [22] proposed public verification scheme using IO. In this scheme, group based verification is not considered. In this paper , Zhang et. al.[22] scheme is extended by forming group of members where different users can share the files.
- Zhang et.al.[22] scheme identified collusion attack between malicious auditor and cloud server but has not given any solution for this. Thokchom and Saikia [20] proposed collusion handling between revoked user and cloud server with integrity verification. This scheme uses vector commitment scheme for integrity verification which increases computation time because of bilinear pairing. Proposed scheme extends Zhang et.al[22] scheme with collusion handling of Thokchom and Saikia[20] between cloud server and revoked user.
- Comparison between existing scheme and proposed scheme is performed and shown that verification time is greatly reduced because of IO.

The remaining part of this paper is organized as follows: Section-II briefed related work. Section III elaborates brief idea about proposed work. Preliminaries for proposed scheme is in Section–IV. Section-V describes proposed work. Sections VI, VII and VIII discusses about security, performance analysis and implementation respectively.

II. Related Work

Many individuals or organizations are using different cloud services for storage as well as sharing information. Drop-box and Google Drive provides sharing services to cloud users. People communicate with each other by creating group and disclosing data among each other. To verify shared data integrity, users in group generate signature on blocks. Different blocks are signed by multiple users during modifications. To maintain security, revoked users must be treated properly. The blocks signed by revoked users must be resigned by existing user in the group to preserve integrity and security. This may create unnecessary burden with respect to computation and communication cost. Yuan et al. [29] proposed integrity auditing scheme by multiuser modification using polynomial-based authentication tags as well as efficient user revocation. This scheme delegates user revocation process to CS to reduce the burden of user but it may create another security issue. Revoked user can collude with malicious CS to retrieve and update the data unnoticed.

Wang et al. [2] proposed a scheme named Panda that proposes public auditing of shared data with user revocation. This scheme uses homomorphic authenticators for integrity verification combined with proxy re-signature scheme. In this scheme, When a user is revoked, to reduce the resigning overhead of existing user, semi-trusted proxy is used to resign the blocks of revoked user. The idea of semi-trusted proxy avoids the collusion between CS and revoked user. But still collusion between revoked user and proxy can leak information. Again this makes system insecure since proxy can get the security credentials of revoked user during revocation.

So to create a collusion resistant public auditing system for shared dynamic cloud data, Jiang et al. [13] proposed another scheme using vector commitment and verifier-local verification group signature. The scheme is efficient but provides only partial data dynamics. Data insertion and data deletion operations are not supported by this scheme. Scheme also shows improvement during verification compared to Panda [2] Scheme.

One of the important functionality of public auditing system is lightweight. There has to be minimum communication as well as computation overhead on TPA and cloud user during verification. Zhang et al. [22] proposed public verification scheme using modern cryptography technique IO. Since IO alone is one of the weaker primitive, if combined with OWF, can implement different cryptographic primitives [21]. Zhang et al. [22] scheme has proposed lightweight public auditing scheme using IO and MAC to check the exactness of outsourced data on cloud storage. This scheme also provides dynamic data updation using Merkle Hash Tree (MHT) as well as Batch Updation. But this scheme faces collusion since user has to share MAC key to TPA. CS may collude with TPA to pass the verification of some malicious updations. Again this scheme doesn't support groups where user can share data and work in collaboration. If this scheme supports group, revoked user may collaborate with CS and TPA to generate collusion attack.

Thokcham [20] proposed collusion resistant public auditing scheme for shared data within a group. This scheme uses vector commitment for integrity verification and ring signature for group operations. Addition and revocation of group members is managed by data owner. During revocation, data owner refreshes valid member list to TPA. Collusion between revoked user and CS is not possible since during verification, TPA uses only this valid member list. Scheme also supports dynamic data updates such as insert, modify and delete. But the problem with this scheme is that as the data size is increased, computation cost during insertion operation is also increased as compared to delete and modify operation. In conclusion, there is a need to construct lightweight collusion resistant public auditing scheme which support shared data with proper user revocation policy.

III. Overview of Proposed Scheme

Proposed scheme involves mainly three entities: Cloud Server (CS), Cloud Users, and Third party Auditor (TPA).Cloud user encompasses data owner or other users in a group. Data owner is any group user who share a file with group members by uploading on CS. Groups are analogous to departments in organizational structure. Every department creates groups of employees in that department to share documents and files. Some employees may be a member of multiple groups.

Proposed scheme works mainly in five phases: Setup, Store, Audit, Prove, and Verify. In setup phase, security parameters are generated. User group is formed by generating private and public key pair for each user. During store phase, any group member or Data owner uploads a file F on CS. Before uploading a file, it is divided into number of blocks. Using secret parameters, file owner creates $\tilde{F}$ which consist of file F comprising blocks n, a file tag τ, and signatures of all data blocks $\{\sigma_i\}_{i\in[1,n]}$. Using ring signature scheme, file owner can sign a data with his private key and public keys of remaining members and uploads a file on CS.

During audit phase, Data Owner generates a challenging message and an auditing circuit corresponding to auditing program, obfuscates it, and passes it to CS. In the meantime, key parameters of obfuscated program are passed to TPA. It reduces the burden of computation on TPA. During prove phase, CS generates a proof based on challenge message and obfuscated program and passes it to TPA. TPA uses proof information, public parameters, challenging message and key elements of obfuscated program to generate a result of verification during verify phase by validating the proof. TPA also validates the signature with verification process of ring signature.

Collusion handling involves revocation scheme which avoid collusion between revoked user and CS. A revoked user is a group member who withdraws the group either because of retirement or any other reason. Such member must not have granted access to group information after leaving organization. File owner handles the addition and deletion of members in group. To avoid the collusion, file owner give updated list of valid users in group signed by him with timestamp to CS and TPA on periodic basis. During verification, TPA considers this latest list received to detect and avoid the collusion.

IV. Preliminaries

This section introduces Indistinguishability Obfuscation, Ring Signature and Merkle Hash Tree, which are basic building blocks of proposed scheme.

A. *Industinguishability Obfuscation*

Concept of program obfuscation was first introduced by Barak et al. [25]. According to this work, program obfuscation is a method which create computer programs "unintelligibile" but still preserve their functionality. They proposed two methods for IO:Virtual Black-Box Obfuscation(VBO) and

Indistinguishability Obfuscation(IO). VBO is nothing but a black box instantiating the program. This technique has several applications in cryptography but authors indicated that VBO is not possible to accomplish. So they have proposed another concept Indistinguishability Obfuscation(IO) which obfuscates any two different (same size) functions that implement unique functionalities, they are still computationally differentiable with respect to each other. This paper follows indistinguishability obfuscation defined by [26].

Amit and Brent [21] have shown how to create basic cryptographic primitives from IO. Psuedo-Random generator (PRG) approach produces public-key encryption where public keys are obfuscated programs and ciphertext are short. This scheme mainly contains three procedures: Setup, Encrypt and Decrypt.

Assume PRG, a pseudo random generator that maps $\{0,1\}^{\lambda}$ to $\{0,1\}^{2\lambda}$. Let F is a puncturable PRF that contains input of 2λ bits and generates a single bit output.

- Setup: This algorithm chooses puncturable PRF key k for F and generates an obfuscated program for PKE Encrypt function as below.

PKE Encrypt
Input: Punctured PRF key k, message m, random value r Є $\{0,1\}^{\lambda}$.

1. Calculate t=PRG(r)
2. Produce c=(c_1=t, c_2=F(k,t) ⊕ m).

The obfuscated program PKE Encrypt* is as below.

- The public key PK is the obfuscated program and secret key SK is k.

PKE Encrypt*
Input: Punctured PRF key k(t^*), message m, random value r Є $\{0,1\}^{\lambda}$.

1. Calculate t=PRG(r)
2. Produce c=(c_1=t, c_2=F(k,t) ⊕ m).

- Encrypt(PK, m): This algorithm selects an arbitrary value r and executes the obfuscated program of PK on input m,r .
- Decrypt(SK, c=($\boldsymbol{c_1}$, $\boldsymbol{c_2}$)): The output of decryption algorithm m'= F(K,$\boldsymbol{c_1}$) ⊕ $\boldsymbol{c_2}$.

B. Ring Signature

Rivest et al. [19] formalized a notion of ring signature scheme for group. This scheme comprises only users. There is no centralized entity such as manager or data owner. This scheme is mainly suitable when the group members do not wish to participate or collaborate in generating signature. With this scheme, there are no prearranged groups, no procedures for altering, deleting groups, no means to issue specific keys among members and no way to revoke anonymity of genuine signer until signer's wish. These features make this scheme very useful for generating proofs in auditing system where groups are involved.

Proposed scheme utilizes CDH based Ring Signature Scheme [28] that is unforgeable and anonymous under CDH noton. Scheme uses multigenerator programmable hash function by Hofheinz and Kiltz [32]. This scheme is demarcated by two algorithms: Ring_sign and Ring-verify.

Ring-sign: This procedure takes as input given message m. Each group member selects a secret key Sk = $\boldsymbol{x_i}$ that belongs to $\boldsymbol{Z_p}$ and public key Pk= $\boldsymbol{g^{x_i}}$.

- Signer t uses global parameter h, $\boldsymbol{u_0}$,$\boldsymbol{u_1}$ …. . $\boldsymbol{u_l}$ Є $\boldsymbol{G_1}$ of l random elements.
- Signer t will select random $\boldsymbol{r_i}$ Є $\boldsymbol{Z_p}$ for entire members of the group and calculate $\boldsymbol{s_i}$ = $\boldsymbol{g^{r_i}}$. Signer again computes.

$$\boldsymbol{s_t} = (\mathbf{h.} \prod_{i=1,i\neq t}^{n} \boldsymbol{Pk_i^{r_i}} \cdot (\boldsymbol{u_0} \cdot \prod_{j=1}^{l} \boldsymbol{u_j^{f_j}})^{-r_{n+1}})^{1/x_t}$$

The ultimate signature is σ = ($\boldsymbol{s_1}$, $\boldsymbol{s_2}$ …. . $\boldsymbol{s_{n+1}}$).

- Ring-verify: Using signature, message F, and public keys of all members, verifier checks the following equation.

$$\prod_{i=1}^{n} \boldsymbol{e(s_i , Pk_i)} \ \mathbf{.}\ \mathbf{e}(\boldsymbol{s_{n+1}}, \boldsymbol{u_0} \prod_{j=1}^{l} \boldsymbol{u_j}^{\boldsymbol{F_j}}) \overset{?}{=} \mathbf{e(g,h)}$$

C. Merkle Hash Tree

In cloud storage system, data holders may modify data dynamically at any time. During auditing process, the homomorphic authenticator scheme utilizes the index data that is to be used in tag computation. But modification in data results in the recomputation of all corresponding authenticators.

The Merkle Hash Tree (MHT) [27] is used to attain data dynamics through auditing. As shown in Fig. 1, the leaves of the MHT are assumed as the blocks *f_i* of file. A publicly qualified root value R and the Auxiliary Authentication Information (AAI) of individual leaf is utilized to check the block *f*. For the path that joins from the leaf to the root, AAI comprises whole siblings of the nodes. Zhang et al. [22] also used the MHT to support dynamic data during auditing. Proposed scheme uses this auditing scheme to support groups in cloud data storage.

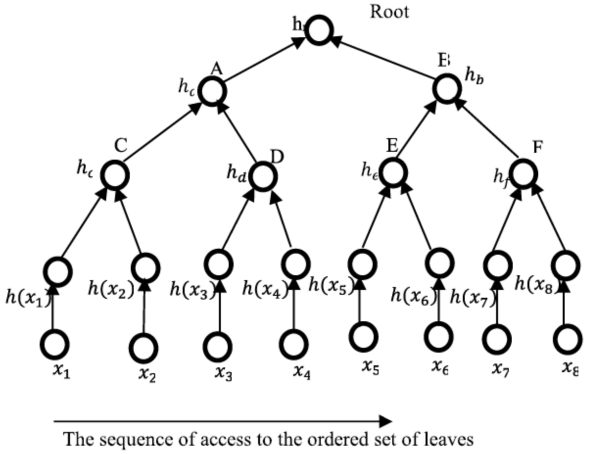

Fig. 1. MHT Authentication Tree.

V. Proposed Scheme

A. System Framework

As shown in Fig. 2, proposed scheme comprises mainly three components: Cloud Server (CS), Cloud User, and Third-Party Auditor (TPA). Signer is any Cloud user form group who share file with group members by uploading it on CS. Before uploading, signer generates verification tags on file. Signer also creates an audit circuit (a program for auditing) which verifies the integrity of outsourced data. Signer obfuscates the audit circuit embedded with MAC key K. Signer shares MAC key K with TPA and obfuscated program with CS.

Based on the challenge message received from TPA, CS calculates the inputs for obfuscated program and runs the obfuscated program. Generated MAC tag is forwarded to TPA. TPA only desires to validate the MAC tag.

B. Group based Integrity Verification

The proposed scheme implements the Zhang et al. [22] framework that works in five phases: *Setup, Store, Audit, Prove, and Verify.* This scheme is modified to handle collusion attack because of group member revocation using ring signature [20].

Setup: Let G and G_T are two multiplicative groups produced by g with order p comprise bilinear map e: G x G → G$_T$. Data owner D selects a signing key pair (ssk, spk), α, v where α → Z_p and v = g^{α} Є G. D chooses s random elements.

$u_1, u_2 \ldots\ldots u_s$ and determines pseudorandom permutation and function key $\pi_{key}(\ )$ and $f_{key}(\ )$ respectively. The secret and public parameters are sk=(α, ssk) and pk=(v, spk, $u_1, u_2 \ldots\ldots u_s$**).** Using key generation of CDH based ring signature scheme, group members randomly selects private key as x_i ЄZ_p and $y_i = g^{x_i}$ Є G as public key.

Store: Data owner transforms the data file F into blocks n and each block is again split into s sectors F= { $f_{i,j}$ }$_{1\le i\le n,\ 1\le j\le s}$. D computes file tag as τ = name|| n|| $u_1, u_2 \ldots\ldots u_s$ || sig_{ssk} (name||n|| $u_1, u_2 \ldots\ldots u_s$ **)** based on randomly selected names. Also computes tag for each data block as:

$$\sigma_i = (\mathrm{H(i||name)}\ .\ \textstyle\prod_{j=1}^{s} u_j^{f_{ij}})^{\alpha}\ ,\ \mathrm{i}\ Є\ [1,\mathrm{n}] \qquad (1)$$

Where H() is any secure hash function. D has to outsource

F̃= { F= { $f_{i,j}$ } $_{i\,Є\,[1,n],\ j\,Є\,[1,s]}$, ϕ ={σ_i}$_{i\,Є\,[1,n]}$, τ } on cloud. Before uploading on cloud, D has to sign a block on behalf of group using CDH based ring signature scheme. D randomly chooses u_0, r_i Є z_p and compute.

$$W_i = g^{r_i} \text{ for i= \{1, 2,…,n+1\}/\{j\}} \qquad (2)$$

Where, n – total number of user members in a group

j - serial number of the user member in the signature

who is signing it

Then compute h= H(ϕ||T) where T is timestamp. D again computes

$$W_j = (\mathrm{h}\ .\ \textstyle\prod_{i=1,i\neq j}^{n} y_i r_i\ .\ (u_0 \prod_{j=1}^{l} u_j{}^{F_j}\ )^{-\mathrm{r}}{}_{\mathrm{n+1}}\ )^{\ 1/}x_j \qquad (3)$$

The signature at time T is

$\phi^T = (W_1, W_2\ \ldots\ldots W_{n+1}$**)**

D uploads F$^{\mathrm{T}}$ = { F= { $f_{i,j}$ } $_{i\,Є\,[1,n],\ j\,Є\,[1,s]}$, ϕ^T , τ } on CS.

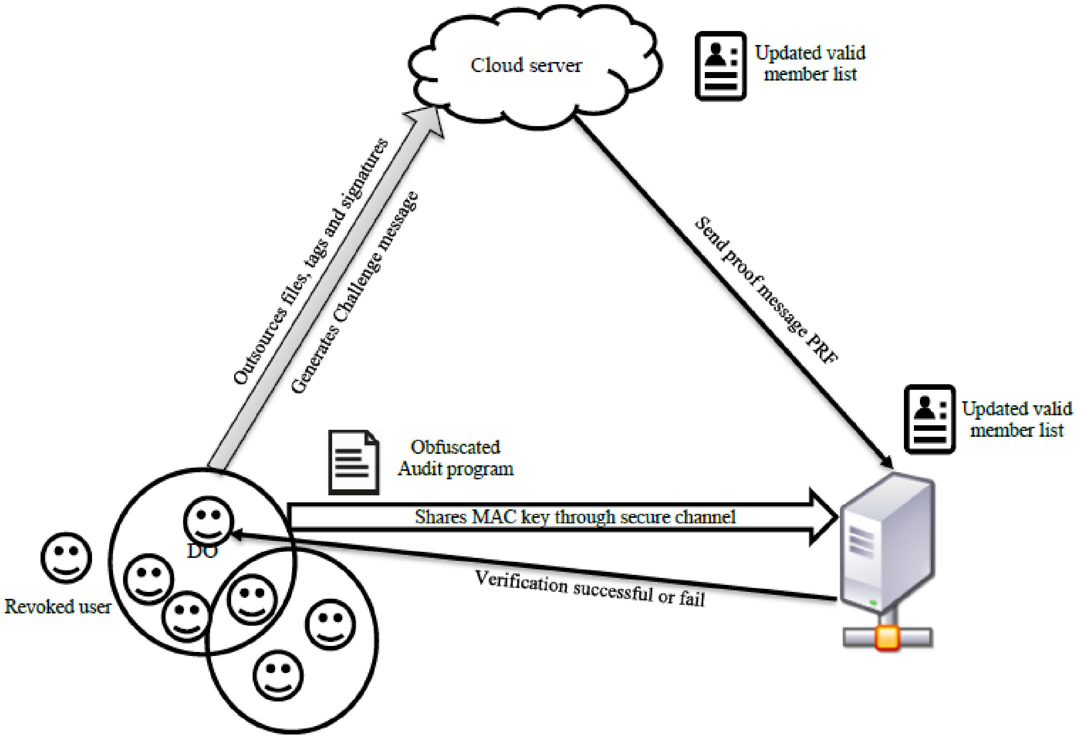
Cloud server
Updated valid member list
Outsources files, tags and signatures
Generates Challenge message
Send proof message PRF
Updated valid member list
Obfuscated Audit program
Shares MAC key through secure channel
Verification successful or fail
DO
Revoked user

Fig. 2. Architecture of Proposed System.

Audit: During this phase, D selects a MAC key k and shares to TPA using a secure channel. D also generates a circuit.

```
$Audit_K$
Input: τ, {(i, $v_i$), $μ_j$} $_{i ∈ [1,n], j ∈ [1,s]}$,σ, {v, g, spk}
Constant: MAC Key K

Validae τ
 If no Valid
 Output ⊥
 Else
 Deconstruct τ to retrieve name, $u_1$,$u_2$ …. . $u_s$
 If Ver ({$(i, v_i)$, $μ_j$ , $u_j$}$_{i∈I,j∈[1,s]}$ , σ, name)=1
 Generate MAC$_k$ (name||{$(i, v_i)_{i∈I}$})
 Else
 Output ⊥
```

$Audit_K$ as described above. Uniform PPT algorithm iO takes securty parameters, audit circuit $Audit_k$ and computes public parameter P as P=iO($Audit_K$). TPA produces a challenge message using data blocks to be audited. Generates $\{k_1, k_2\}$ which are keys for pseudorandom permutation and function respectively. TPA sends these keys to CS.

Prove: Using $\{k_1, k_2\}$, CS computes i=πk_1(ξ) and v_i= fk_2(ξ) where ξ Є [1,c] and c is size of I (Input blocks to be audited). Based on public parameters and corresponding τ, $f_{i,j}$, σ_i,c CS computes σ= $\prod_{i∈I} {\sigma_i}^{v_i}$, $μ_j$= $\sum_{i∈I} v_i f_{ij}$ and

Prf= P(τ, {(i, v_i) , $μ_j$}$_{i∈I}$, σ, {v, spk})

CS send this PRF to TPA for verification phase.

Verify: Using CDH based ring signature process, TPA verifies the group signature based on input signature ϕ^T and public keys (y_1,y_2 …. . y_n**)** of all members in group, F^T, public parameter u_0. TPA first calculate h=H(ϕ||T). Then verifies

$$\prod_{i=1}^{n} e(W_{ij}, y_i)\ e(W_{n+1}, u_0 \prod_{j=1}^{l} {u_j}^{f_j}) = e(h,g)$$

To verify the correctness of data, TPA computes πk_1(ξ) and v_i= fk_2(ξ) and verify Prf $\overset{?}{=}$ MAC_k(name||{(i, v_i) $_{i∈I}$ }).

C. User Revocation

Data owner D can revoke any user because of some reason. When data owner revokes user, the signatures calculated on file blocks by revoked user have to be re-calculated again by existing user. The re-signature process is as follows:

Initially, D selects any random user to take responsibility for the blocks earlier signed by revoked user. D selects randomly an element u_0 and send it to existing user. Upon receiving this parameter u_0, existing user selects random $r_i \leftarrow z_p$. Then Computes W_i using (2) for all members except himself.

The existing user also computes hash using timestamp T and then generates signature with his own secret key using (3). Re-calculated signature φ^T = (W_1,W_2 …. . W_{n+1}**)** is outsourced on CS with tag calculated using (1) and original file blocks. The existing user also prepares a valid group member list, sign the list and share it with CS and TPA. While verifying the signature, TPA uses this updated member list which helps to detect and avoid collusion attack.

D. Dynamic Data Updation

The users can modify, insert and delete information in the files outsourced by D on CS. To enable this, Zhang et. al scheme [22] used MHT to avoid recalculation for block indexes of the file. In proposed scheme, during *Store* phase, D initially produces a tag for each data block and generates a tree with root ω_{MHT} and sends it to TPA. During *Audit* phase, D also generates an audit circuit for dynamic support.

During *prove* phase, CS sends Prf and Auxiliary Authentication Information (AAI) which comprises path list of node siblings to reach from the leaves to the root. During *Verify*, TPA validates ω_{MHT} and AAI. The dynamic updation scheme of Zhang et al. [22] is used as it is because even though any member of group has updated information in file, the changes are reflected in MHT during *Store* and *Audit* phase. TPA can verify the changes using ω_{MHT} and AAI.

VI. Security Ananlysis

This section describes security proof related to proposed system.

A. Authenticity

Theorem 1: For the cloud, it is impracticable to deceive the TPA and user in case of forgery.

Proof: The contents of outsourced files on CS may be corrupted or deleted intentionally or unintentionally (Hardware Failures). With proposed scheme, CS can't hide these changes from TPA and user. We prove this by a smiple game sequence as follows: Assume $\widetilde{CS}$ as malicious who try to pass the verification for corrupted data blocks.

1) The challenger selects (ssk, spk), α, K, u_1,u_2 …. . u_s as described in section V.

2) The challenger produces the circuit $Audit_k$ and calculates v=g^{α}, iO ($Audit_k$) and set it as public parameter.

3) $\widetilde{CS}$ generates the proof $\widetilde{prf}$ and send it to TPA.

4) Challenger verifies the proof by computing *prf*.

5) $\widetilde{CS}$ wins, if $\widetilde{prf}$ differs from *prf* while challenger does not reject.

In above game, it is not possible for $\widetilde{CS}$ to cheat challenger because of secure HMAC scheme in the system. Even though $\widetilde{CS}$ try to pass it's corrupted data blocks, there is a difference between the proof generated, which results in verification failure.

B. Revocation

Theorem 2: In proposed scheme, the group together with the CS is capable of converting the signature from one revoked user into signature of an existing user after revocation and revoked user not able to access group information with his security parameters.

Proof: To prove this, we use an arrangement of game as follows:

1) User U_a is revoked from a group because of some reason. D randomly chooses an existing user U_b , who is responsible for computing the re-signature on the blocks signed by U_a.

2) D randomly selects u_0 and send to U_b. User U_b make u_0 public and select random $r_i \leftarrow z_p$.

3) User U_b computes $W_i = g^{r_i}$ for all members of group except himself.

4) Downloads the blocks signed by U_a. Computes the hash and calculate the re-signature with his own private key x_j using (3) as in section V.

After revocation, in above game, there is no need for D or any group member to manually delete the security parameters of user U_a. User U_a even though trying to access the group information, not able to do that since user U_b has already uploaded re-signed data blocks and valid user list on CS.

C. Collusion Resistant

Theorem 3: It is impracticable for the CS to fabricate valid proofs to clear the verification test, even though a revoked user colludes with the CS.

Proof: Considering the above same game, assume that user U_a is revoked. He colludes with CS and modified some blocks of files. When TPA gives audit challenge for this file block, CS generate the fabricated proof.

Prf'= P(τ, {(i, v_i) , μ_j}$_{i\in I}$, σ, {v, spk})

User U_a wins, if TPA does not reject and pass the verification. U_a become successful in generating collusion attack. But collusion is not possible in proposed system since whenever member is revoked, D updates the current valid signed members list to CS and TPA. While validating the signature of users, the auditor utilizes only these valid members list signed by D.

VII. Performance Analysis

To check the performance of public auditing system for cloud storage, different functionalities can be considered. These functionalities are: third party auditing, dynamic data operation, user revocation, immune to collusion attack, membership to several groups using the same key set. Multiple schemes proposed by different researchers explore some functionalities. Some schemes provide partial dynamic data updates such as only insertion and modification of data excluding deletion. Another functionality, lightweight auditing is the scheme in which TPA and data owner has to incur less burden as per communication and resource cost to complete the auditing task. A detailed comparison of these schemes is as below in Table I.

Initially we analyze the communication cost of proposed scheme and then evaluate it with different existing schemes. In auditing system, to analyze communication overhead consider communication between three entities: User, CS and TPA. Mostly communication overhead between user and CS is insignificant since user uploads the entire data to CS initially. So the communication overhead between CS and TPA is analyzed since these are the two entities involved in proof generation and verification process.

In proposed scheme, TPA challenges CS for specific blocks. So communication overhead between TPA and CS is $|\boldsymbol{K_1}|$+$|\boldsymbol{K_2}|$+HMAC where $\boldsymbol{K_1}$ and $\boldsymbol{K_2}$ are transformed keys of HMAC. Based on challenge, CS generates the proof by calculating HMAC through obfuscated program. This proof submitted to TPA for verification. During verification, TPA checks the correctness of outsourced data by confirming:

Prf $\overset{?}{=}$ $\boldsymbol{MAC_k}$(name||{(i, $\boldsymbol{v_i}$) $_{i\in I}$ })

So TPA has to only calculate the HMAC. Along with this TPA has to verify the signatures of users using verification method of CDH based ring signature. So communication overhead for TPA during verification is to calculate hash and verify each user using public key of each member. During user revocation, D revokes user and existing user has to resign the blocks signed by revoked user. The existing user has to download blocks signed by revoked user, calculates hash and recomputes the signature of all n group members and signs it with his own private key. The computation cost for the above three operations compared with existing methods is shown in Table II. The meaning of each notation is as follows: M for multiplication, P stands for pairing, H means for hashing, E for exponential, c is the number of challenged blocks, q is the total number of data blocks, s is the number of elements in a block and z is number of revoked user.

TABLE I. COMPARISON OF EXISTING SCHEME IN TERMS OF FUNCTIONALITY

Scheme	**Third-Party Auditor**	**Dynamic Data Operation**	**User Revocation**	**Immune to Collusion Attack**	**Membership to several groups using same key set**	**Lightweight**
Wang et.al.[2]	Yes	full	Yes	No	No	No
Jiang et. al.[13]	Yes	Partial	Yes	Yes	No	No
Thokcham et. al[20]	Full	Yes	Yes	Yes	Yes	No
Zhang et. al.[22]	Yes	Yes	No	No	No	Yes
Yuan and Shucheng[29]	Yes	Partial	Yes	No	No	No
Proposed System	Yes	Yes	Yes	Yes	Yes	Yes

TABLE II. COMMUNICATION OVERHEAD

Scheme	Proof Generation	Verification	User Revocation
Wang et.al.[2]	cM+cE	(c+n)E+(c+3n)M+(n+1)P+cH	2E+M+2P+H
Jiang et. al.[13]	(q-1)(M+E)	7P+M+9E+5H+z(M+2P)	Z(M+2P)
Thokcham et.al[20]	(q-c)(M+E)	(n+4)P+6E+7M+(c+1)M	(2n+2)E+nM
Yuan and Shucheng[29]	sE+(s+n)M+nP	6E+3M+3P	CE
Proposed Scheme	cH	(c+n)H	(c+n)E+H

VIII. IMPLEMENTATION AND EVALUATION

In this section, the implementation and evaluation of proposed scheme are discussed with experiments. All the experiments are carried out on a system having Windows 10 with AMD A12 processor, 2.70 GHz, 4.0 GB RAM. Considering security level as 128 bits, experiments are tested 5 times and average values are taken. All algorithms are implemented using Python language. For implementation, some blocks are kept constant so the size of each block may vary.

The performance of proposed system is evaluated in terms of cost of dynamic data operations, verification time with respect to group size and communication overhead in terms of KB. To evaluate the dynamic data operation cost, mainly three operations are performed: delete, modify and insert. Verification time (in Sec) of these operations for different data sizes varying from 2KB to 1000KB files is analyzed. Fig. 3 shows the performance of proposed scheme compared with Thokcham[20] scheme. The graph shows that insertion operation cost in Thokcham [20] increases as the data size is increased whereas verification time for all three operations in proposed scheme is constant.

Verification time with respect to group size is analyzed as depicted in Fig. 4. The verification time of proposed scheme is compared with a different existing scheme such as Panda [2], Yuan [29], Jiang [13] and Thokcham [20]. PS indicates proposed scheme. Graph shows increase in verification time for scheme Panda [2] and Thokcham [20]. Whereas Yuan [29], Jiang [13] and proposed scheme is constant even though the number of user members are increased. Evaluation of proposed scheme is performed by adding 100 users to the group. Graph proves that verification time in proposed scheme is not depending on the number of members in a group.

Table II shows the evaluation of communication overhead of proposed scheme where it is evaluated for proof generation, verification and user revocation. To calculate the complete communication overhead of auditing in terms of KB, the size of an auditing message is considered. The size of a message is calculated during integrity verification using IO technique and signature verification using CDH based ring signature scheme. The magnitude of an auditing message is 2|MAC|.(c+n) +n|MAC| bits where, |MAC| is the size of MAC generated by CS and TPA, c is the number of challenged blocks and n is total number of users. Zhang et al. [22] scheme has given the performance of communication overhead in terms of KB by considering challenged number of blocks. Since proposed scheme is based on Zhang's [22] scheme, for comparison, communication overhead is computed with respect to the number of users in a group. Fig. 5 shows the performance of communication overhead in KB. For 10 users, the communication cost of auditing is 10.24KB. The graph shows that as the number of users are increased, auditing message size also increased. We have not compared this computation cost with existing work because in our scheme, we have kept number of blocks as fixed during splitting the file while in most existing work, size of block is fixed. We have given the computation cost results for 200KB file, which is divided into 5 blocks of 40KB each.

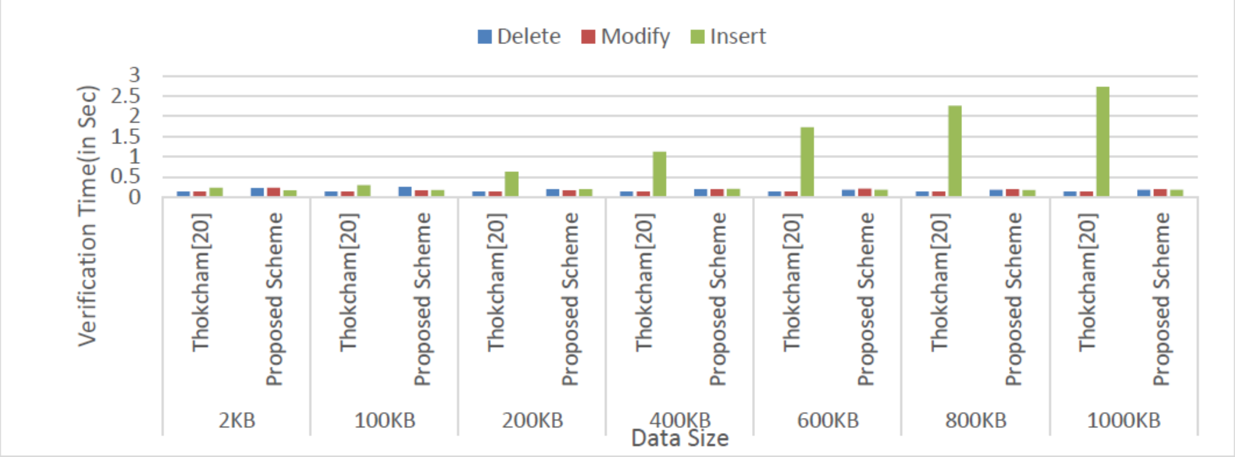

Fig. 3. Comparison of Cost for Dynamic Data Operation.

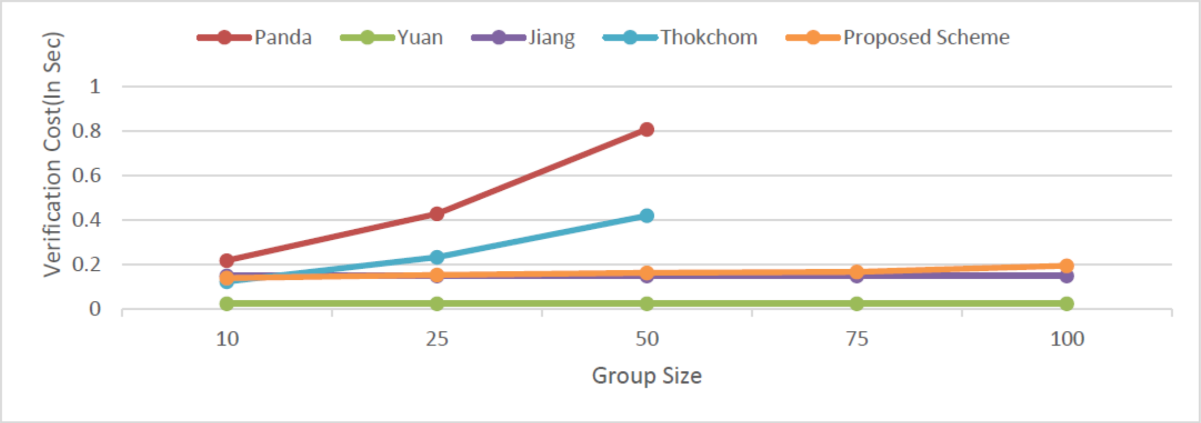

Fig. 4. Verification Time with respect to Group Size.

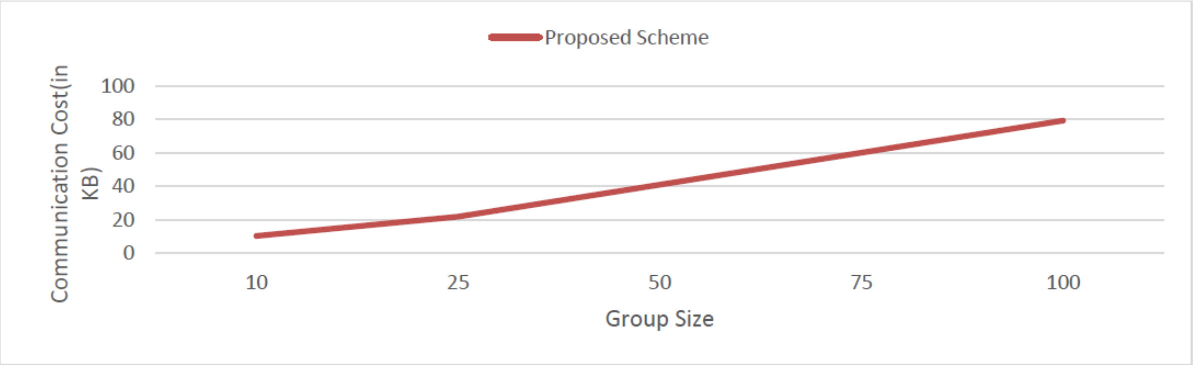

Fig. 5. Communication Cost with respect to Group Size.

IX. Conclusion

This article proposes an efficient and secure auditing scheme for cloud storage using modern cryptographic technique, Indistinguishability Obfusction. Proposed scheme allow cloud users to form a group and share information or files within group. Proposed scheme adopts ring signature scheme to sign the data files which are outsourced on cloud. During auditing, TPA check integrity of data by calculating MAC as well as signature of block using public keys of all users. Collusion may occur between revoked user and cloud if revocation not done properly. Our scheme proposed revocation policy as well as updates valid member list to CS and TPA which lead to avoid collusion in system.

The proposed scheme is efficient and lightweight since TPA only have to calculate the MAC tag for verification. The performance of proposed scheme is proved by comparing verification time during dynamic operations with existing schemes. Also analyzed the performance of communication overhead during auditing in terms of KB.

In regards to future work, we want to extend our scheme to include batch auditing in which TPA must have the competence to execute the auditing tasks concurrently. In our scheme, after revocation of any member data owner depute any existing user to re-computed the signatures of revoked user. This may create an additional burden on existing user. As a future work, we want to extend our revocation scheme to address this issue.

Acknowledgeent

This research work is an independent work and no financial assistance has been received for this work.

References

[1] S. Chaudhari, S. K.. Pathuri, “A Comprehensive Survey on Public Auditing for Secure Cloud Storage,” International Journal of Engineering and Technology, vol. 7, no. 2.7, pp. 564–569, 2018.

[2] B. Wang, C. Li and H. Li, “Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud” , IEEE Transactions on Services Computing, Vol. 8, No. 1, pp. 92-106, Feb. 2015.

[3] Q. Chan, C. Zhang and S. Zhang, “Detection Models of Collusion Attacks,” in Secure Transaction Protocol Analysis- (Lecture Notes in Computer Science).

[4] W.Guo , H. Zhang , S. Qin, F. Gao , Z. Jin , W. Li and Q.Wen , “Outsourced Dynamic Provable Data Possession with Batch Update for Secure Cloud Storage, Future Generation Computer Systems 95, pp.309-322, 2019.

[5] K. Huang , M. Xian , S.Fu and J. Liu, “Securing the Cloud Storage Audit Service: Defending Against Frame and Collude Attacks of Third Party Auditor “ , IET Communications , Vol 8 , No.12,pp.2106-2113,2014.

[6] Z.Wang , S. Cheung and Y. Luo ,”Information-Theoratic Secure Multi-party Computation with Collusion – Deterrence “ , IEEE Tranctions on Information Forensics and Security , Vol.12, No.4,2017.

[7] X.Wang, A.Hu and H.Fang , “Inproved Collusion-resistant Unidirectional Proxy Re-increption from Lattice , IET Information Security , Vol.14 , No.3 , pp.342-351,2020.

[8] Z. Sun , Y. Yang , Q. Shen , Z. Wu and X. Li , “MB-DDIVR: A Map-based Dynamic Data Integrity Verification and Recovery Scheme in Cloud Storage “ , In ICICS (Lecture Notes in Computer Science), 2016 , pp.335-345.

[9] S.S. Chow , M. H. Au and W. Susilo , “Server-Aided Signatures Verification Secure against Collusion Attack “, Information Security Technical Report 17 , 2013 , pp.46-57.

[10] H. Carter and P. Traynor, “ OPFE: Outsourcing Computation for Private Function Evaluation , IACR Cryptology Epint Arch . 2016.

[11] Z. Zhu and R. Jiang , “A Secure Anti-collution Data Sharing Scheme for Dynamic Groups in the Cloud “ , IEEE Tranctions on Parallel and Distributed Systems , Vol.27 , No.1, pp40-50 , Jan.2016.

[12] Q. Wu , Y. Mu , W. Sucilo, B. Qin and J. Domingo-Ferrer , “A Symetric Group Key Management “ in Proc.International Conferrence On the Theroy and Applications of Cryptographic Techniques , 2009,pp.153-170.

[13] T. Jiang , X. Chan and J. Ma , “ Public Integrity Auditing for Shared Dynamic Cloud Data with Group User Revocation “ , IEEE Tranctionns on Computers , Vol.65 , No.8 , Aug.2016.

[14] S, Kumar and L. Parthiban , “Cloud Data Integrity Auditing Over Dynamic Data for Multiple Users “ , International Joural of Intelligent Engineering and Systems , Vol.10 , No.5 , pp.239-246 , 2017.

[15] L. Hequn , W. Baocang , L. Ke , G. Ziyuan and Z. Yu , “ Public Auditing for Shared Data Utilizing Backups with User Revocation in the Cloud “ , Journal of Natural Sciences , Vol.23, No.2, pp.129-138, 2018.

[16] Y. Luo, M.Xu, K. Huang, D. Wang and S. Fu , “ Efficient Auditing for Shared Data in the Cloud With Secure User Revocation and Compuataions Outsourcing “ , Computers and Security , Vol.73 , pp.492-506, Mar.2018.

[17] M. Liu , Y. Wu , J. Chang , R. Xue and W. Guo , “ Verifiable Proxy Re-increption from Indistinguishability Obfuscation “ , in Proc.International Conference on Information and communication Security(Lecture Notes in Computer Science) , pp.363-378, 2016.

[18] L. Zhu , H. Wang , C. Xu , K. Sharif and R. Lu , “ Efficient Group Proof of Storage with Mallacious-member Distinction and Revocation “ , IEEE Access Vol.7 , pp.75476-75489, May.2019.

[19] R. Rivest , A. Shamir and Y. Tauman , “ How To Leak A Secret “, in Proc.Therory and Applications of Cryptology and Information Security-ASIA CRYPT , pp.552-565, 2001.

[20] S. Thokcham and D. Saikia , “ Privacy Preserving Integrity Checking of Shared Dynamic Cloud Data with User Revocotion” , Journal of Information Security and Applications , pp.2214-2126 , 2020.

[21] A. Sahai and B. Sahai “ How To Use Indistinguishability Obfuscation : Deniable Increption and More , in Proc. 46th Annual ACM Symposium on Theroy of Computing , pp.475-484 , May.2014.

[22] Y. Zhang , C. Xu , X. Linag , H. Li , Y. Mu and X. Zhang , “Efficient Public Verification of Data Integrity for Cloud Storage Systems from Indistinguishability Obfuscation “ , IEEE Tranctions on Information Forensics and Security , Vol.10 , No.3, pp.676-688, Mar.2017.

[23] S. Chaudhari , G. Swain and P. Mishra , “ Secure and Verifianble Multiparty Computation using Indistinguishability Obfuscation “ , International Journal of Intelligent Engineering and Systems , Vol.13 , No.5 , pp.277-285, Jul.2020.

[24] C. Guan , K. Ren , F. Zhang , F. Kerschbaum and J. Yu , “ Symetric-key Based Proofs of Retrivibility Supporting Public Verification “ , in Proc.ESORICS (Lecture Notes in Computer Science) , pp.203-223, 2015.

[25] B. Barak , O. Goldreich , R. Impagliazzo , S. Rudich , A. Sahai , S. Vadhan and K. Yang , On the (im) Possibility of Obfuscating Programs , in Proc.Advances In Crypotology – CRYPTO (Lecture Notes in Computer Science) , pp.1-18, Aug.2001.

[26] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai and B. Waters, “Candidate Indistinguishability Obfuscation and Functional Encryption for all circuits” , in Proceedings of FOCS, IEEE 2013,pp. 40-49.

[27] Q. Wang , C. Wang , K. Ren , W.Lou and J. Li , “ Enabling Public Auditablility and Data Dynamics for Storage security in Cloud Computing “ , IEEE Tranctions on Parellel and Distributed Systems , Vol.22, No.5, pp.847-859, May.2012.

[28] S. Schage, J. Schwenk, “A CDH-based Ring Signature Scheme with Short Signature and public keys”, in Proc. FC2020 (Lecture Notes in Computer Science 6052) , pp.129-142, 2010.

[29] J. Yuan, S. Yu, “Public Integrity auditing for Dynamic Data Sharing with Multiuser Modification”, IEEE Transactions on Information Forensics and Security, Vol. 10, No. 8, pp. 1717-1726, 2015.

[30] L. Sun, C.Xu, Y. Zhang and K. Chen,”Public Data Integrity Auditing without Homomorphic Authenticators from Indistinguishability Obfuscation”, International Journal of Information security,Vol.19,pp. 711-720, 2020.

[31] R. Rabaninejad, M. Attari, M. Asaar and M. Aref, “A Lightweight Identity-based Provable Data Possession Supporting User Identity Privacy and Traceability”, Journal of Information Security and Applications, Vol. 51, 2020.

[32] D. Hofheinz and E. Kiltz, “Programmable Hash Functions and Their Applications”, in Proc. Advances in Cryptology, LNCS 5157, pp. 21-38, 2008.