Privacy Preserving Dynamic Provable Data Possession with Batch Update for Secure 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*—Cloud Server (CS) is an untrusted entity in cloud paradigm that may hide accidental data loss to maintain its reputation. Provable Data Possession (PDP) is a model that allows Third Party Auditor (TPA) to verify the integrity of outsourced data on behalf of cloud user without downloading the data files. But this public auditing model faces many security and performance issues such as: unnecessary computational burden on user as well as on TPA, to preserve identities of users from TPA during auditing, support for dynamic updates etc. Many PDP schemes creates computational burden either on TPA or Cloud User. To balance this overhead between TPA and User, this paper proposes Privacy-Preserving Dynamic Provable Data Possession (P^2DPDP) scheme, which is based on ODPDP scheme. In ODPDP scheme, user relieves the burden by signing a contract with TPA regarding verification of his outsourced data. But this scheme generates computation overhead on TPA. To reduce this computation overhead of TPA, our P^2DPDP scheme uses Indistinguishability Obfuscation (IO) with one-way function such as message authentication code to make a lightweight auditing process. P^2DPDP scheme uses Rank-based Merkle Tree (RBMT) to support dynamic updates in batch mode which greatly reduces computation overhead of TPA. ODPDP lacks privacy which is maintained in P^2DPDP using ring signature technique. Our experimental results demonstrate the reduced verification time and computation cost compared to existing schemes.**

Keywords—Public auditing; ring signature; Indistinguishability Obfuscation; Rank-based Merkle Tree (RBMT)

I. Introduction

With the rapid development of cloud computing, many individuals or small-scale organizations started outsourcing their data on untrusted CS. This paradigm even though, proved to be a boon, has brought many security challenges with it. The user is not having physical ownership of data since outsourced data may be stored at any server. This outsourced data may get damaged unintentionally because of disk crashes or natural disasters. Sometimes CS may delete infrequent data intentionally to create space for new users. These incidents or data loss are not reported to cloud users to maintain reputation [1]. Provable Data Possession (PDP) is a technique that allows any user to check the integrity of data blocks outsourced on CS without downloading the entire data file. Many researchers have proposed cryptographic techniques using homomorphic authenticators [2]-[12]. Cloud users can check the integrity of outsourced data on their own but this frequent task creates an additional burden on the cloud user in terms of time and computation cost. So, researchers have proposed solution in which user can delegate auditing work to TPA having expertise and skill. TPA generates a challenge message and CS has to produce the proof based on recent data. This proof is verified by TPA. This model has many security challenges since CS is one of the untrusted entities and TPA even if trusted, curious about auditing work.

Integrity checking during dynamic update operations of data is a major challenge in the PDP model. Authenticators have to be recalculated during insertion, deletion, and modification operations on data because most of the authenticators are calculated based on indices of files. Researchers have proposed dynamic auditing schemes using Merkle Hash Tree (MHT) [24]. MHT is a hash-based data structure used for data verification. Another data structure used for dynamic data updates are Index Hash Table (IHT) [5],[6] Dynamic Hash Table (DHT) [18]. These techniques need some additional information to store which may increase storage and computation cost. Guo et al. [21] proposed Multi-leaf-authenticated (MLA) scheme for dynamic data using Rank based Merkle Tree (RBMT). This scheme authenticates multiple leaf nodes at once without storing height and status value. With this scheme, multiple dynamic update operations can be performed in batch mode.

In auditing model, TPA is one of the trusted entity but still curious and may compromise data and user privacy. During auditing, TPA may infer user information who have signed the blocks. Researchers have given multiple approaches to address this issue. Some privacy-preserving protocols [39],[40] are proposed based on aggregate signature [14] or hash-based commitment [15] but auditing is not mentioned in these schemes. Huang et al. [18] proposed privacy- preserving scheme using group signature and blockchain. Tian et al. [20] also proposed a privacy-preserving scheme that addresses data privacy using random masking to blind the data proof. Identity privacy is preserved using a modification record table to record operation information. Different variations of signature algorithms, such as blind signature, random sampling, ID-based privacy, and attribute-based signature are used by multiple researchers during auditing [22]-[28]. Attribute-based signatures are based on attribute-based cryptography [16],[17]. Ring signature is another variation of group signature in which ring or group is formed with multiple users. User sign the blocks and share it among the group members. Verifier determines that block is signed by one of the group members

*Corresponding Author

but can't reveal who has signed the block. Certificateless authentication [29],[30] is one of the cryptographic techniques for authentication. Some authors proposed privacy-preserving integrity verification scheme [31],[32] using certificate-less ring signature which greatly reduces the computation cost during auditing. Li et al. [19] proposed integrity checking of group shared data using certificateless signature. Ni et al. [33] proposed lightweight ID-P^{3}DP scheme in which privacy is achieved through zero-knowledge proof.

Many of the above auditing schemes are based on homomorphic authenticators which incur high computation cost and time during auditing. There is a need to propose a lightweight auditing process which can create a little burden on TPA as well as on cloud user. Indistinguishability Obfuscation (IO) is a modern cryptographic technique that uses one-way function for implementation of different cryptography constructs [35]. Researchers [36]-[38] proposed an efficient public verification scheme using IO combined with one-way function MAC which makes this scheme lightweight by generating very little burden on TPA. Tu et al. [13] proposed user-focus auditing which try to reduce the overhead of user by pre-generating challenges for TPA before auditing. Guo et al. [21] also proposed outsourced auditing scheme in which after each verification, TPA generates an audit data log which is checked later by the user. These schemes reduce the burden of verification on user. There is no need for a user to be available during verification, as per his convenience he can check the audit log.

Most of the auditing schemes create computational burden of tag generation and verification either on TPA or user. If we try to reduce the burden of TPA, it will increase the burden on user or vice-versa. Zhang et al. [36] scheme reduces the computation overhead of TPA but cloud user is actively involved in auditing process. Guo et al. [21] scheme proposes auditing scheme where user is not involved in auditing process but it creates computation burden on TPA because of Efficient Homomorphic Verifiable Tags (EHVT).

The remainder of this paper is described as follows: Section II describes related work, problem identification and contributions by authors. Section III elaborates basic building blocks of P^{2}DPDP scheme. Section IV explains the P^{2}DPDP scheme. Section V discusses the evaluation of P^{2}DPDP scheme in terms of security and performance.

II. Related Work, Problem Identification and Research Contribution

A. Related Work

A good number of solutions have been proposed by many researchers [1]-[10] for integrity verification of outsourced data. In most of these schemes, cloud user and TPA are actively involved during verification phase. This may create an additional burden on cloud users as well as on TPA. Guo et al. [21] proposed ODPDP scheme which relieves user's verification overhead by migrating frequent auditing tasks to TPA. In this scheme, a contract takes place between user and TPA regarding the frequency of verification task. TPA generates challenges based on this contract. After each verification, a log file is generated at TPA which contains an audit data log. User as per his convenience can check the log and make sure the integrity of his data as well as working of TPA. This scheme greatly reduces the overhead on user side. This scheme uses Multi-Leaf Authentication (MLA) solution with RBMT for dynamic data updates which greatly reduces storage cost as well as allows verification of multiple dynamic operations in batch mode.

To minimize the burden of TPA in terms of computations, it is necessary to develop a verification scheme which is lightweight in terms of computation. Zhang et al. [36] proposed lightweight auditing technique using IO and one-way function MAC. This greatly reduces the computation overhead of TPA during verification since TPA has to just calculate MAC each time and verify it with the received proof from CS. In this scheme, during outsourcing data at CS, a user has an additional overhead of generating circuit (audit program), obfuscating with MAC key, and send it to CS. But this is only a one-time cost to generate obfuscated program since it would not change along with the modification of public parameters and challenge message. This scheme uses Merkle Hash Tree (MHT) to support dynamic data updates on file.

In auditing model, generating privacy-preserving auditing technique is a major challenge because of the curious but trusted nature of TPA. Thokcham [34] proposed a privacy-preserving auditing technique using CDH based ring signature. This ring signature scheme is unforgeable and completely anonymous. Any ring member can sign a message using his private key and public keys of other members. So not every member needs to be present during the signing process. Using this scheme, anyone can check whether a signature is generated by a valid member of the group but at the same time not revealing the user's identity who has signed that message. Thus, preserving the identity privacy of user during verification from TPA.

ODPDP scheme reduces user overhead but increases burden on TPA in terms of computation. The computation cost on TPA in ODPDP during auditing is (l+s+1)Exp_G + 2Pair. Where l is number of challenged blocks, s denotes number of sectors per block, Exp_G is exponentiation operationon on group G and Pair is pairing operation. Compare to ODPDP scheme, Zhang et al. [36] scheme create less burden on TPA during auditing i.e., 2 $Hash_{Zp}$ where Hash is hashing operation on Z_p. It means TPA has to compute only 2 hash functions for verification. So proposed P^{2}DPDP scheme modifies the ODPDP by using IO and MAC proposed by Zhang et al. scheme instead of EHVT of ODPDP for integrity verification. ODPDP scheme not proposing any solution for identity privacy. We extend this scheme by using CDH based ring signature to achieve identity privacy proposed by Thokcham [34]. P^{2}DPDP also uses RBMT of ODPDP which allows verification of multiple dynamic operations in batch mode compared to MHT in Zhang et al. [36].

B. Problem Identification

To balance the computation overhead between user and TPA, this paper proposes Privacy-Preserving Dynamic Provable Data Possession (P^{2}DPDP) scheme for cloud storage. In this scheme, the main purpose of using IO is to reduce the computation burden of TPA while maintaining security. Since

TPA only needs to validate the commitments generated by CS, user's data will not be revealed to TPA which preserve data privacy during auditing process. To avoid the continuous involvement of user during auditing process, Cloud user and TPA sign a contract which includes starting address of the block, the frequency at which auditor launches a challenge, and number of challenged blocks. TPA verifies the outsourced data based on contract and generates a log file which can be verified by user as per his convenience. For the support of dynamic updates, RBMT is used that can perform multiple update operations in a batch way. P^{2}DPDP support user groups and preserve identity privacy by using CDH based ring signature scheme.

C. Research Contribution

Specifically, the contribution of our scheme P^{2}DPDP is as follows:

- Guo et al. [21] proposed ODPDP scheme which uses Effective Homomorphic Verifiable Tag (EHVT) for integrity verification and RBMT for dynamic data processing. To make the auditing process lightweight and to reduce computation overhead of TPA, we modify the integrity verification scheme of ODPDP by using indistinguishability obfuscation of Zhang et al. [36].
- Guo et al. [21] scheme not offering a solution for identity privacy during auditing. We extend this ODPDP scheme to achieve group user privacy using CDH based ring signature.
- We describe a concrete P^{2}DPDP scheme to be secure and lightweight by modifying ODPDP scheme. Experimental results certify the performance of our scheme.

III. Preliminaries

This section introduces cryptographic building blocks used in P^{2}DPDP scheme such as IO for integrity verification, MLA for dynamic updates and CDH based ring signature scheme for privacy-preserving.

A. Industinguishability Obfuscation (IO)

Indistinguishability obfuscation is a notion that obfuscates any two distinct (equal size) programs that implement identical functionalities but computationally indistinguishable from each other [35].

Assume $\{C_l\}$is a circuit class with security parameters l. A uniform PPT algorithm iO having input l, circuit C ∈ $\{C_l\}$ and outputs a circuit C' is called indistinguishable obfuscator if the following conditions are fulfilled:

1) For all security parameters l, Circuit C, and input x, we have probability as:

$$\Pr[C'(x)=C(x)]=1, \text{ where } C'=iO(l,C) \tag{1}$$

Equation (1) satisfies the completeness property of IO. It states that circuit C' must behave exactly same as circuit C if C' is generated by an independent invocation of iO on C.

2) For any (not essentially uniform) PPT adversaries D, for all security parameter l ∈ N, for all pairs of circuits C_0, $C_1 \in C_l$, there exists a negligible function Negl such that if C_0(x) = C_1(x) for all inputs x, then.

$$|\Pr[D(iO(l, C_0))=1]- \Pr[D(iO(l, C_1))=1]|\leq Negl(l) \tag{2}$$

Equation (2) satisfies the indistinguishability property of IO. It states that the secrets embedded in obfuscated program cannot be extracted by D.

Zhang et.al. [36] proposed integrity verification using IO and one-way function MAC.

B. Multi-Leaf Auhentication (MLA)

ODPDP scheme [21] proposed MLA for dynamic updates. This scheme uses RBMT instead of MHT to support authentication of indices of leaf nodes. In RBMT, no need to store height value as in MHT. Each node contains only two fields (r,h) where r is the rank of a node which is the number of leaf nodes reachable from node ω and h is a hash value of that node. Mainly, the rank of a leaf node is 1 i.e., r=1. The second element h is defined as in (3):

$$h = \begin{cases} H_2(m_i), if\ \omega\ is\ the\ ith\ leaf\ node \\ H_1(r||\omega.left.h||\omega.right.h), if\ \omega\ is\ a\ nonleaf\ node \end{cases} \tag{3}$$

where H_1 and H_2 be a secure collision-resistant hash function and || denotes concatenation. The outsourced file F is divided into n blocks such as F= $\{f_1, f_2 \ldots, f_n\}$. The ith element f_i is bind to the ith leaf node of RBMT by storing the hash value of f_i at node ω_i using H_2 in (3). Thus, leaf nodes are already sorted from left to right by their indices. For each non-leaf node, ω.left.hash and ω.right.hash indicates the hash value of the left node and right node respectively calculated using H_1 in (3). RBMT can be constructed for given n data blocks. A Merkle root h_{root} is sufficient to check the integrity of dynamic updates in a tree because of the dependency of all data blocks. Fig. 1 shows the RBMT constructed over 14 data blocks. In Fig.1, when multiple leaf nodes are challenged using MHT such as ω_3 and ω_7, the proofs generated are $\boldsymbol{\Omega}_3$= $\{\omega_{26}, \omega_{22}, \omega_{15}, \omega_4, \omega_3\}$ and $\boldsymbol{\Omega}_7$= $\{\omega_{26}, \omega_{21}, \omega_{17}, \omega_{18}, \omega_7\}$. During verification using MHT, some repeated and unnecessary nodes have to be retrieved and processed that incur large computation costs. But using MLA solution, if multiple challenged nodes are (3,7,8,10,13), the corresponding multi-proof $\sqcup_{\boldsymbol{p}}$ is:

$$\sqcup_{\boldsymbol{p}} = \{\omega_3, \omega_7, \omega_8, \omega_{10}, \omega_{13}, \omega_{16}, \omega_{18}, \omega_{19}, \omega_{21}, \omega_{22}, \omega_{23}, \omega_{24}, \omega_{25}, \omega_{26}\}$$

where every necessary node appears just once which reduces computation cost as well as support multiple updates in batch mode.

C. CDH based Ring Signature Scheme

To achieve privacy during auditing, Thokcham [34] used CDH based ring signature which is one of the unforgeable and anonymous technique. No centralized entity is involved i.e., no concept of group manager. This scheme comprises two algorithms: Ring_sign and Ring-verify.

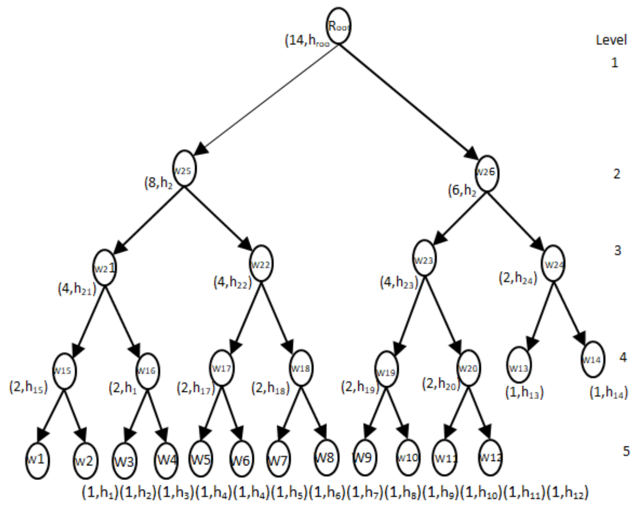

Fig. 1. RBMT Authentication Tree.

Ring-sign: This algorithm takes as input given message M. For a size of ring n, each group member chooses a secret key *Sk* = x_i which belongs to Z_p and public key *Pk* = g^{x_i}.

- Signer t uses global parameter d, u_0,u_1 .....u_l ∈ G_1 of *l* random elements.
- Signer *t* will choose random r_i ∈ Z_p for all other members of the group and generates $s_i = g^{r_i}$. Signer *t* again computes signature on behalf of group

$$s_t = (d. \prod_{i=1,i\neq t}^{n} Pk_i^{r_i}\ (u_0 . \prod_{j=1}^{l} u_j^{m_j})^{-r_{n+1}})^{1/x_t} \quad (4)$$

In (4), signer *t* computes signature s_t using his private key X_t with different parameters such as: public keys of *n* group members *PK*, the global parameter *d*, u_0,u_1 .....u_l, message *M* divided into *l* elements (M_1 .....M_l), random number *r*. Final signature is σ = (s_1, s_2 .....s_{n+1}).

Ring-verify: As per (5), verifier verifies the signature using received ring signature σ, message M, and public keys *PK* of all members. The verifier checks the following equality.

$$\prod_{i=1}^{n} e(s_i , Pk_i) . e(s_{n+1}, u_0 \prod_{j=1}^{l} u_j^{m_j}) = e(g,d) \quad (5)$$

In (5), using public keys PK, signature (s_1, s_2 .....s_{n+1}), and received message *M*, recomputes $\prod_{i=1}^{n} e(s_i , Pk_i)$. *e* ($s_{n+1}$, $u_0 \prod_{j=1}^{l} u_j^{m_j}$). This recomputed part is verified using global parameter *d* and *g*.

IV. Proposed Scheme

A. System Model

The framework for our P^{2}DPDP scheme is as shown in Fig. 2. It consists of three entities: Cloud User, CS, and TPA.

- Cloud User: Cloud user is one of the members of user group who can share a file in a group. Users can check the integrity of shared files through an audit log generated by TPA.
- CS: an entity having the capability of computation and storage at its end. It is having the responsibility to maintain and manage outsourced files.
- TPA: an external entity that works on behalf of users and expertise in verifying the integrity of outsourced data.

The workflow for P^{2}DPDP scheme from Fig. 2 is as follows:

1) Any cloud user from a group outsources data file on CS. Before outsourcing, user calculates the tag for each block, signs the blocks using a ring signature scheme. 2) The user constructs RBMT tree using hash values of file blocks. User generates circuit for auditing program, obfuscates it, and sends to CS. Shares MAC key to TPA as well as sign a contract with auditor regarding verification activity. 3) Based on the frequency mentioned in the contract, TPA generates challenges and performs auditing activity. During an audit, using CDH based ring signature, TPA verifies group signature using public keys of all members in a group. 4) Generates log file for each activity. 5) Users can check the log entry at any time to verify the integrity of an outsourced file. 6) For dynamic updates, user sends the update command uc to the TPA. 7) TPA updates the RBMT tree according to uc and sends updated proof to CS for verification. If verification successful, CS sends signed proof to user. 8) User verifies proof and if successful, send updated data blocks ui to CS. CS updates data blocks accordingly.

B. Design Goals

To achieve privacy-preserving during integrity verification of outsourced data, proposed scheme P^{2}DPDP should satisfy the following design objectives:

1) Public verification: to allow an external auditor to verify the integrity of outsourced data without downloading the data file.

2) Privacy-Preserving: data or user identity must not be revealed to TPA during auditing.

3) Data Dynamic Support: integrity verification process must support dynamic updations on outsourced data such as insert, delete and modify operations.

4) Lightweight: verification process must create minimum communication and computation overhead on user and TPA.

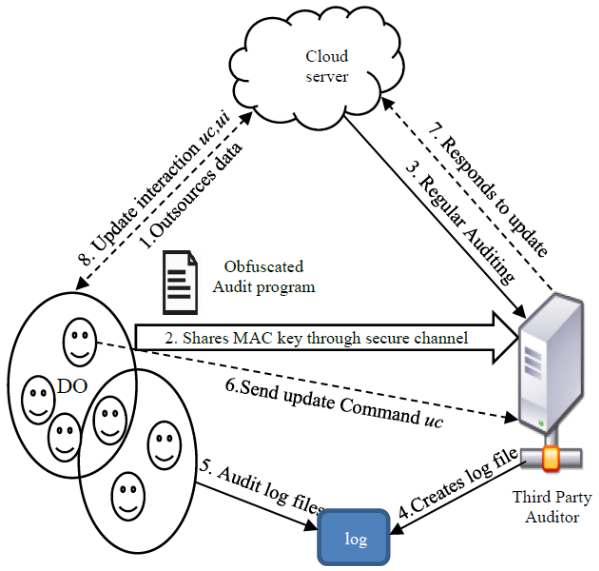

Fig. 2. Architecture of P2DPDP Scheme.

C. P^2DPDP Scheme

Our proposed P^2DPDP scheme works in four phases: Setup, Store, AuditData, and AuditLog.

Setup: Let G and G_T be two multiplicative groups produced by g with order p contains bilinear map e: G x G → G_T. User

U selects a signing key pair (ssk, spk), α, v where $\alpha \rightarrow Z_p$ and v = g^{α} Є G. *U* picks random elements $u_1, u_2 \ldots\ldots u_s$ and fixes pseudorandom permutation, function key $\pi_{key}(\ )$ and $f_{key}(\ )$ respectively. The secret and public parameters are *sk*=(*α*, *ssk*) and *pk*=(*v*, *spk*, $u_1, u_2 \ldots\ldots u_s$). Group members randomly select private key as $x_i \in Z_p$ Using key generation of CDH based ring signature scheme and $Pk_i = g^{x_i} \in G$ as a public key.

Store: Initially according to (6), *U* divides the file *F* into blocks *n* and sector *s* as in.

$$F = \{f_{i,j}\}_{1\leq i\leq n,\ 1\leq j\leq s} \quad (6)$$

Tag Generation:

U chooses random element name for file and computes file tag as

τ = name||n||$u_1, u_2 \ldots\ldots u_s||sig_{ssk}$ (name||n||$u_1, u_2 \ldots\ldots u_s$)

and data tag as in (7).

$$\sigma_i = (H(i||name) \cdot \prod_{j=1}^{s} u_j^{f_{ij}})^{\alpha},\ i \in [1,n] \quad (7)$$

In (7), User calculates data tag for each bock *i* of file F using random elements of each sector $u_1, u_2 \ldots\ldots u_s$ and hash value of block number and file name. Here *H* is any secure hash function. *U* generates processed data *M* as *M ={M,ϕ}* where

$\phi = \{\sigma_i, \tau\}_{i \in [\mathbf{1},\mathbf{n}]}$

Constructing RBMT:

U first calculates hash values for each block of file F using H2.

$h_i = H_2(m_i)$ where 1≤ i ≤n

Then generate tree TR using RBMT on ordered hash values. Each leaf node w_i stores the corresponding hash value h_i.

Ring Signature Generation:

U has to sign a block on behalf of a group using CDH based ring signature scheme. *U* randomly chooses u_0, $r_i \in Z_p$ and compute signature for all other group members except *U* denoted by *j* in (8).

$$S_i = g^{r_i} \text{ for i= } \{1,2,\ldots,n+1\}/\{j\} \quad (8)$$

Where, n - number of members in a group

j - serial number of the member in the signature who

is signing it

U computes signature for every member using respective random number *r* of that user. Then computes h= H(ϕ||T) where T is a timestamp. *U* again computes S_j using (4).

The signature at time T is

$$\phi^T = (S_1, S_2 \ldots\ldots S_{n+1}) \quad (9)$$

U outsources *M* and ϕ^T calculated as in (9) on CS. In (9), $S_1, S_2 \ldots\ldots S_{n+1}$ is the signature generated for *n* users by *U* on behalf of group at time *T*.

Outsourcing Auditing Task:

During this task, U chooses a MAC key k and passes it to TPA using a secure network. U also produces a circuit $\boldsymbol{Audit_K}$ as described below.

```
                    $Audit_K$
Input: $\tau$, $\{(i, v_i), \mu_j\}_{i \in [1,n], j \in [1,s]}$, $\sigma$, $\{v, g, spk\}$
Constant: MAC Key $K$

Verify $\tau$
 If Invalid
 Output $\perp$
 Else
        Deconstruct $\tau$ to get $name, u_1, u_2 \ldots\ldots u_s$
      If Ver $(\{(i, v_i), \mu_j, u_j\}_{i \in I, j \in [1,s]}, \sigma, name)$=1
           Output $MAC_k(name||\{(i, v_i)_{i \in I}\})$
 Else
 Output $\perp$
```

This circuit is similar to auditing program which generate the MAC using embedded MAC Key K based on given input. Uniform PPT algorithm iO proceeds with audit circuit $Audit_k$ as input and generates public parameter P as P=iO($Audit_K$).

Ver $(\{(i, v_i), \mu_j, u_j\}_{i \in I, j \in [1,s]}$, σ, name) in circuit denotes checking of (10):

$$e(\sigma, g) = e(\prod_{i \in I} H(i||name)^{v_i} \cdot \prod_{j=1}^{s} u_j^{\mu_j}, v) \quad (10)$$

- U outsources RBMT tree TR with signature Sig_{ssk}(TR) to TPA. If verification is successful, TPA accepts TR else rejected considering the malicious author.

Agreeing Parameters:

- All group members need to sign on a public parameter $P_{pub} = \{q, h_{root}\}$ where *q* is the number of data blocks and h_{root} is a Merkle root of *TR*.
- Contract *CT* is established between *U* and TPA as

CT = {BI,Fr,b), where

BI- block index from which auditing work will start.

Fr- Frequency at which TPA launches a challenge.

b- number of challenged data blocks for checking.

AuditData Protocol: This protocol mainly deals with checking the integrity of outsourced data and log generation by TPA.

Auditing Phase:

- TPA generates a challenge message $Q^{(b)} = \{b, K_1^{(b)}, K_2^{(b)}\}$ using data blocks *b* to be audited. TPA

Generates pseudorandom permutation and function keys $\{K_1{}^{(b)}, K_2{}^{(b)}\}$ respectively and send to CS.

- CS computes i=$\pi K_1{}^{(b)}$(ξ) and v_i= f$K_2{}^{(b)}$ (ξ) where ξ Є [1, *b*] and *b* is the size of I (Input blocks to be audited). Based on public parameters and corresponding σ_i calculated using (7), τ, $f_{i,j}$ and *b*, CS computes:

$\sigma^{(b)}= \prod_{i\in I} \sigma_i{}^{v_i}$, $\mu_j= \sum_{i\in I} v_i f_{ij}$ and

$$PRF^{(b)}= P\,(\tau,\ \{(i, v_i), \mu_j\}_{i\in I}\ , \sigma^{(b)},\ \{v, spk\})\ (11)$$

Equation (11) shows the proof *PRF* calculated by CS for challenged blocks *b*. *PRF* is calculated by executing audit circuit $Audit_K$ i.e. P=iO$(Audit_K)$ using input parameters *file tag* τ, $\{(i, v_i), \mu_j\}$ *σ*, {*v*, *g*, *spk*}. CS send this $PRF^{(b)}$ and $Sig_{sk_{CS}}(PRF^{(b)})$ to TPA for verification.

- Using CDH based ring signature process, TPA verifies the group signature based on input signature ϕ^T, public keys (Pk_1,Pk_2 ….. Pk_n) of all members in a group, F^T and public parameter u_0 . TPA first calculate *h=H(ϕ||T)*. Then verifies signature using (5).

- To verify the integrity of data, TPA computes *i*= $\pi K_1{}^{(b)}$ (ξ) and v_i= f$K_2{}^{(b)}$(ξ) and compare MAC with $PRF^{(b)}$ calculated by CS using (11).

$PRF^{(b)} = MAC_k$(name||{(i, v_i) $_{i\in I}$ }).

Log Generation:

After every audit, either successful or fail, TPA creates a log record of his auditing work.

$L^{(b)}$=**{t,** $Q^{(b)}$**,** $PRF^{(b)}$**,** $Sig_{sk_{CS}}(PRF^{(b)})$**}**

and saves in his local file *Log_File*.

AuditLog Protocol:

- *U* generates challenge using random subset B of file block indices and sends it to TPA. For each *b* Є B, TPA finds challenge $Q^{(b)}$, proof $PRF^{(b)}$ from his log file. Computes i, $\boldsymbol{v_i}$ from $Q^{(b)}$. TPA generates multi_audit proof $\sqcup_p$ using h_{root} of *TR* and generates the proof of appointed log for subset *B* using (12).

$$PRF^B = \{\ \sqcup_p,\ \text{i},\ v_i\ ,\ PRF^{(b)}\ \} \qquad (12)$$

In (12), *B* indicates the challenge generated by *U* during *AuditLog*. Elements *i*, $\boldsymbol{v_i}$denotes the challenge retrieved through log file for blocks *b* and $PRF^{(b)}$ indicates proof retrieved from log file for blocks *b*. $\sqcup_p$ is a multi-audit proof generated from RBMT.

- TPA send a signed proof with his signature$Sig_{sk_{TPA}}(PRF^{(B)})$ to *U* for verification. After verifying the signature, *U* computes new PRF as.

PRF_{new} = MAC_k**(name||{**$(i, v_i)_{i\in B}$**})**

- *U* compares if PRF_{new} = $PRF^{(B)}$. If matched, verification of outsourced data is successful else verification fails. *U* also verify $\sqcup_p$ using h_{root} of *TR*.

D. *Support for Dynamic Updates*

P²DPDP scheme support three types of update operations such as: deletion, insertion, and modification on blocks. If we perform these updates one by one, it will incur a large computation overhead at the auditor side to generate and verify the hash tree. To reduce this overhead, P²DPDP is based on a MLA scheme using RBMT proposed by ODPDP which can handle updates in batch instead of one by one.

Initially, *U* computes all the hash and tag values of the new file block in *Store* phase, generates the RBMT tree, and set public parameter as P_{pub} ={q, h_{root} }. *U* sends the update command *uc* to CS and TPA. *U* also generates audit circuit same as basic scheme but with modified verification function VerD $(\{(\text{i},\text{v}_\text{i}),\mu_\text{j},\text{u}_\text{j},\sigma,name\}_{\mathbf{i\in I,j\in[1,s]}})$ which consists of checking following equation:

$$e(\sigma,g)=e(\textstyle\prod_{i\in I} H(\sum_{j=1}^{s} m_{ij})^{v_i}\ .\ \prod_{j=1}^{s} u_j{}^{\mu_j},v)$$

After receiving uc command, TPA updates leaf nodes, other affected nodes and generate an updated tree TR^* and it's Merkle root h_{root^*}. TPA then sends the updated signed proof up to CS. CS verifies up by executing the audit circuit. U can also later check the correctness of up. If verification is successful, U sends updated information *ui* to CS and CS updates the processed data.

V. Evaluation

In this section, P²DPDP scheme is evaluated by showing correctness proof, security analysis, performance and experiment analysis.

A. *Correctness Proof*

$Audit_K$ is an audit circuit generated by U during Store phase. Upon execution of this audit circuit, CS generates MAC based on challenge, block tag σ and using global parameters. Audit circuit contains verification function which denotes the realization of (10). Correctness proof for (10) is as follows:

$$e(\sigma,g) = e(\textstyle\prod_{i\in I} \sigma_i{}^{v_i},\ g)$$

$$=\mathbf{e}(\textstyle\prod_{i\in I}(H(i||name).\prod_{j=1}^{s} u_j{}^{f_{ij}})^{v_i},\ \boldsymbol{g})$$

$$= \mathbf{e}(\textstyle\prod_{i\in I}(H(i||name)^{v_i}).\ \prod_{i\in I}\prod_{j=1}^{s} u_j{}^{f_{ij}v_i})\ ,\ \boldsymbol{g})$$

$$= \mathbf{e}(\textstyle\prod_{i\in I}(H(i||name)^{v_i}).\prod_{j=1}^{s} u_j{}^{f_{ij}})\ ,\ v_i)$$

B. *Security Analysis*

Storage Correctness:

Theorem 1: If the CS successfully passes the verification

from an auditor, then data outsourced on CS must be intact.

Proof: Assume user U outsourced data at CS using Store protocol. But due to some problems, data at CS accidentally corrupted or deleted. With P²DPDP scheme, malicious CS can't pass its verification. We prove this by game sequence as below:

1) Based on a contract signed between U and TPA, TPA generates a challenge using AuditData Protocol.

2) For the challenged blocks, CS computes i, v_i. Using

3) Public parameters and (k_1, k_2), CS computes $\widetilde{PRF}$ and send to TPA.

4) TPA computes i= π k_1(ξ) and v_i= f k_2(ξ) where ξ Є [1,c] and c is the size of I (Input blocks to be audited).

5) Using MAC_k, TPA calculates PRF and compares it with $\widetilde{PRF}$.

6) CS wins if TPA passes the verification even if PRF ≠ $\widetilde{PRF}$.

But in above game, it's very difficult for malicious CS to cheat auditor because of HMAC scheme during verification.

Liability:

Theorem 2: An honest auditor can demonstrate that he did his work correctly in case of any disputes.

Proof: To prove the liability of the auditor, we consider two situations: when an auditor is honest or an auditor is dishonest. Consider first the auditor is honest. As per the contract between auditor and User, the auditor generates a challenge $Q^{(\boldsymbol{b})}$. User can reconstruct the challenge since the contract consists of number of data blocks to be audited. User can check the value of PRF by recalculating (11). Honest auditor generates a log file named Log_File which is the evidence for all the auditing work completed by auditor. So honest auditor can prove his liability by this Log_File.

Compare to this, if the auditor is malicious and not doing his work properly, user can use AuditLog Protocol to verify the behavior of auditor. By regenerating challenge, user can check the AuditLog file anytime and auditor can't deny his misbehavior.

Privacy-Preserving:

Theorem 3: From the server's response to the challenge message, TPA not able to infer any information such as data and identity of user.

Proof: During verification, user U first generates an audit circuit (which is nothing but an auditing algorithm program which is supposed to be originally executed by TPA). U obfuscates this circuit by embedding MAC key K and send to CS. For each verification, CS computes the inputs based on the challenge message and executes the obfuscated program. CS generates the MAC tag and sends it to TPA. TPA has to only verify the MAC tag to check the integrity of outsourced data. TPA needs to calculate i and v_i based on challenged blocks using the HMAC scheme. So, it is computationally infeasible for TPA to infer any information or user data using P^2DPDP.

P^2DPDP uses CDH based ring signature scheme to share any data among group members. In this scheme, user who want to share a file, computes signature on this data with his own private key using (4). During verification, TPA can verify the signature with public keys of all users. Using this scheme, TPA can check whether the signature is computed by a valid user of group or not but scheme can't reveal individual user identity to verifier. Thus because of CDH based ring signature scheme and IO, P^2DPDP proved to be privacy-preserving.

C. *Performance Analysis*

We first evaluate the performance of P^2DPDP scheme which shows the privacy-preserving, lightweight auditing process. Also, we compare the performance of P^2DPDP with existing schemes.

The main important functionalities which we have considered for this work are: public auditing, dynamic data operations, privacy-preserving, and group support. Table I shows the functionality comparison of P^2DPDP scheme with existing schemes.

To evaluate the performance of P^2DPDP scheme, we evaluate the communication cost between CS and TPA during the proof generation and verification phase of *AuditData* protocol. Communication cost between the user and CS is not important since user uploads the data entirely to CS initially and user can verify the integrity of outsourced data during AudiLog protocol. During proof generation, TPA generates a challenge message $\{k_1, k_2\}$ for b number of blocks where k_1 and k_2 are transformed keys of HMAC. After receiving challenge message, CS generates proof PRF by calculating HMAC through the obfuscated program. So, communication overhead for proof generation is bH where H is any secure hash operation. After generating the proof PRF using HMAC, CS sends it to TPA. During verification, TPA checks the integrity of outsourced blocks using MAC key and verify

PRF = $\boldsymbol{MAC_k}$(name||{(i, $\boldsymbol{v_i}$) $_{i∈I}$ })

So TPA has to calculate only HMAC. TPA also verifies the user signature by verification algorithm of CDH based ring signature. Using (5), TPA verifies the users *n*. So, the total communication overhead during verification is H_{Zp}+ *n* where H_{Zp} is hashing operation into Z_p.

User can check the integrity of outsourced data or performance of TPA during *AudiLog* protocol. So total Communication overhead during *AuditLog* is also H_{Zp}. Table II shows the comparison of P^2DPDP scheme with the existing scheme. Fig. 3 shows the comparison between ORUTA, CORPA, and our P^2DPDP scheme for communication overhead in KB with respect to group size.

We can't compare this cost with ODPDP scheme since it doesn't support user groups. The comparison demonstrates the noticeable and constant performance of communication cost between CS and TPA during auditing in P^2DPDP scheme.

TABLE I. THE COMPARISON OF AUDITING FUNCTIONALITIES

Scheme	Third-Party Auditor	Dynamic Data Operation	User Group Support	Privacy Preserving
ORUTA [5]	Homomorphic Authenticators (HA)	Index Hash Table	Ring Signature	Homomorphic Authenticable Ring Signature (HARS)
Tian et al. [20]	Homomorphic Verifiable Authenticators (HVA)	Dynamic Hash Table	Group Signature	Random Masking
ODPDP [21]	Efficient HVA	Rank Based Merkle Tree (RBMT)	NA	NA
CORPA [23]	HA	Merkle Hash Tree (MHT)	Group Signature (GS)	HA+GS
Thokcham and Saikia[34]	Vector Commitment	MHT	CDH based Ring Signature	CDH based Ring Signature
Zhang et al. [36]	Indistinguishability Obfuscation (IO)	MHT	NA	NA
Proposed Scheme P^{2}DPDP	Indistinguishability Obfuscation (IO)	Rank Based Merkle Tree (RBMT)	CDH based Ring Signature	CDH based Ring Signature

TABLE II. COMMUNICATION COST

Scheme	Proof Generation	Proof Verification	
		AuditData	***AuditLog***
ORUTA [5]	$(s+nb)E+nbM+bsM+sH$	$(2s+b)E+(2s+b)M+nM+bH+(n+2)p$	NA
ODPDP [21]	$(2q+s+1)\ E$	$(b+s+1)\ E+2p$	$3E$
CORPA [23]	$H+M$	$2E+P+bE+bM$	NA
Proposed Scheme P^{2}DPDP	bH	$H+n$	$H+n$

s: Total no. of sectors
n: Total no. of users in a group
b: No. of challenged blocks
q: Total no. of blocks
E: Exponentiation operation
M: Multiplication Operation
H: Hash Operation
P: Pairing Operation

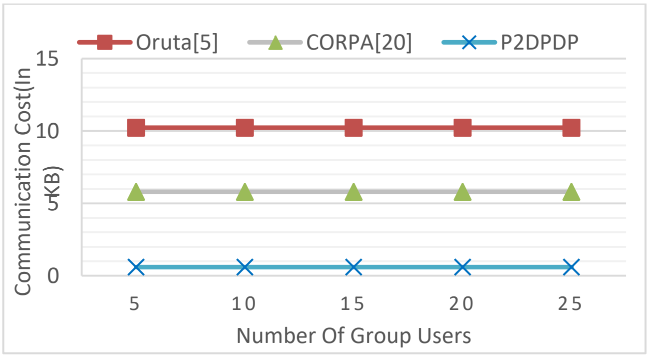

Fig. 3. Communication Cost w.r.t. Group Size.

D. Experimental Results

This section proves the performance of P^{2}DPDP system in terms of different experiments. We deployed our P^{2}DPDP scheme on a system comprising Windows 8.1with an Intel Core i5-5200U CPU functioning at 2.20 GHz, 4.0 GB RAM. Python is used for module implementation of P^{2}DPDP scheme. The hash algorithm is instantiated using SHA256. Fig. 4 shows the impact of number of users in a group on verification time during AuditData protocol. It shows that verification time is independent of the number of users. By creating a group of 25 users, we have compared the results with Oruta and CORPA scheme. We have not considered ODPDP scheme for comparison since it doesn't support user groups. All results are average of 5 runs. For 20 users in P^{2}DPDP scheme, the Verification time is 0.15 seconds. While Oruta, and CORPA are 2.24, and 1.75 seconds respectively. Result proves the effectiveness and lightness of our scheme because of reduced and constant verification time at auditor side during AuditData protocol. Since our P^{2}DPDP scheme is based on ODPDP scheme, it is mandatory for us to compare results with this scheme. In both the schemes, verification is performed during AuditData and AuditLog protocol. Initially we compare the results of both schemes during AuditData where TPA performs audit based on contract and generate log entries. Fig. 5 shows the proof generation and verification time in seconds with respect to challenged data blocks during AuditData protocol.

To compare the results with ODPDP scheme, in P^{2}DPDP scheme, the block size is kept fixed i.e., 16KB. Total outsourced data is 1GB. Fig. 5(a) shows the constant proof generation time in P^{2}DPDP scheme as compared to ODPDP. Fig. 5(b) shows the gradual increase in proof verification time as number of challenged blocks is increasing in P^{2}DPDP as compared to ODPDP scheme. Fig. 6 shows the performance of P^{2}DPDP during *AuditLog*. It presents the computation time and communication cost required by user to verify the past work of TPA under the number of checked log entries.

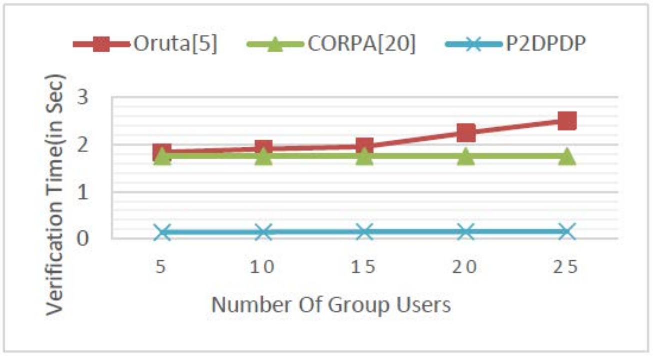

Fig. 4. Impact of Number of Group users on Verification Time.

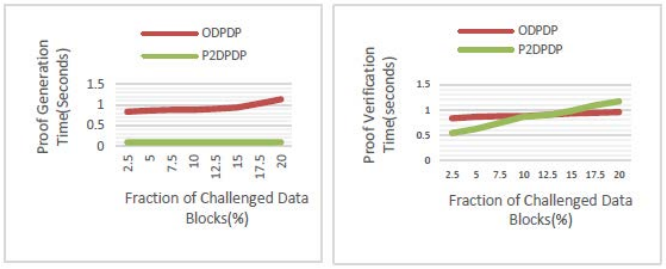

Fig. 5. (a) Proof Generation Time during *AudiData* Protocol w.r.t. Challenged Data Blocks; (b) Proof Verification Time during *AudiData* Protocol w.r.t. Challenged Data Blocks.

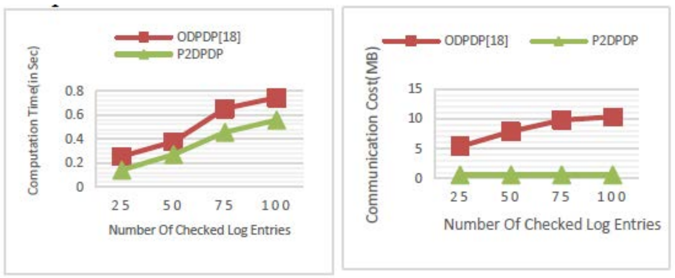

Fig. 6. (a) Computation Time for user to Check Log Entries w.r.t. Number of Checked Log Entries; (b) Communication Cost for user to Check Log Entries w.r.t. Number of Checked Log Entries.

For experiments, we have analyzed the computation time of our scheme up to 100 log entries. As expected, computation time is increasing linearly with number of checked log entries. But computation time of P^2DPDP scheme is reduced as compared to ODPDP scheme.

Fig. 6 shows the communication cost during *AuditLog* protocol. Result shows that P^2DPDP scheme is giving better performance in terms of communication cost compared to ODPDP.

VI. Conclusion

In most of the previous auditing scheme, Cloud user and TPA are actively involved during verification process which may create additional burden in terms of time and cost. This paper proposes P^2DPDP scheme for cloud storage in which there is no need of user during verification process. TPA generates challenges based on the contract signed between TPA and user. TPA also generates log which can be audited by user as per his convenience. P^2DPDP scheme also create the light-weight verification process so as to reduce the computation burden of TPA using new cryptographic technique, Indistinguishablity Obfuscation and MAC. P^2DPDP support and manages user groups using CDH based ring signature scheme. CDH based ring signature is anonymous scheme which preserves the identity of users from TPA during auditing. P^2DPDP scheme supports dynamic updates in batch mode using MLA solution proposed by ODPDP scheme which is based on RBMT.

Security analysis and experiments show that P^2DPDP scheme is secure, lightweight and privacy-preserving. Communication cost during auditing between CS and TPA is almost constant and reduced compared to Oruta and CORPA since TPA has to just calculate MAC and compare it with MAC received from CS. Verification time is also reduced and constant compare to existing schemes. Experimental results reveal that verification time is independent of number of group users. Results of *AudiLog* protocol shows that P^2DPDP scheme is performing better in terms of communication time and cost as compared to ODPDP scheme. CDH based ring signature generates certificates which need to be processed during verification leads to increase computation time. In terms of future work, we plan to modify P2DPDP scheme using certificateless signature schemes to reduce computation time.

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] C. Wang, S. Chow, Q. Wang, K. Ren and W. Lou, “Privacy-Preserving Public Auditing for Secure Cloud Storage”, IEEE Transactions on Computers, Vol. 62, no.2, pp.362-375, 2013.

[3] B. Wang, H. Li and M. Li, “Privacy-Preserving Public Auditing for Shared Cloud Data Supporting Group Dynamics”, in Proc. IEEE International conference on Communications, pp. 1946-1950, 2013.

[4] K. Yang and X. Jia, “An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing”, IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No. 9, pp. 1717-1726, 2013.

[5] B. Wang, B. Li and H. Li, “Oruta: Privacy-Preserving public Auditing for Shared Data in the Cloud”, IEEE Transactions on Cloud Computing, Vol.2, No. 1, pp. 43-56, 2014.

[6] Y. Yu, Y. Mu, J. Ni, J. Deng and K. Huang, “Identity Privacy-Preserving Public Auditing with Dynamic Group for Secure Mobile Cloud Storage”, in Proc. International Conference on Network and System Security, LCNS 8792, pp. 28-40, 2014.

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

[8] H. Tian, Y. Chen, C. Chang, H. Jiang, Y. Huang, Y. Chen and J. Liu, “Dynamic Hash Table based Public Auditing for Secure Cloud Storage”, IEEE Transactions on Service Computing, Vol. 10, No.5, pp. 701-714, 2017.

[9] G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu and R. Hao, “Enabling Public Auditing for Shared Data in Cloud Storage Supporting Identity Privacy and Traceability” Journal of Systems and Software, Vol. 113, pp. 130-139, 2016.

[10] M. Ma, J. Weber and J. Berg, “Secure Public-Auditing Cloud Storage Enabling Data Dynamics in the Standard Model”, in Proc. IEEE Conference on Digital Information Processing, Data Mining and Wireless Communications, pp. 170-175, 2016.

[11] B. Kang, J. Wang and D. Shao, “Attack on Privacy-Preserving Public Auditing Schemes for Cloud Storage”, Mathematical Problems in Engineering, vol. 2017, Article ID 8062182, 2017.

[12] M. Krishna, B. Harika, P. SasiPriya, V. Nikitha, K., and Bharath M, “Homomorphic cryptography”, Journal of Advanced Research in Dynamical and Control Systems, Vol.10, No. 4, pp.129-136, 2018.

[13] T. Tu, L. Rao, H. Zhang, Q. Wen and J. Xian, “Privacy-Preserving Outsourced Auditing Scheme for Dynamic Data Storage in Cloud”, Security and Communication Networks, vol. 2017, Article ID 4603237, 2017.

[14] C. Xu, X. Shen, L. Zhu and Y. Zhang, “A Collusion-Resistant and Privacy-Preserving Data Aggregation Protocol in Crowdsensing System”, Mobile Information Systems, vol. 2017, Article ID 3715253, 2017.

[15] X. Cao, H. Li, L. Dang and Y. Lin, “A Two-Party Privacy-Preserving Set Intersection Protocol against Malicious Users in Cloud Computing”, Computer Standards and Interfaces, Vol. 54, No. 1, pp. 41-45, 2017.

[16] L. Tamma, and S. Ahamad, “A novel chaotic hash-based attribute-based encryption and decryption on cloud computing”, International Journal of Electronic Security and Digital Forensics, Vo. 10, No.1, pp. 1-19, 2018.

[17] N. Vurukonda, S. Trijan Kumar, J. R. Reddy, A., Adithya, and S. Boddu, “A secure attribute-based encryption scheme in cloud computing”, International Journal of Engineering and Technology (UAE), Vol. 7, No. 2, pp. 90-92, 2018.

[18] L.Huang, G. Zhang and A. Fu, “Privacy-Preserving Public Auditing for Non-Manager Group Shared Data”, Wireless Press Communications, pp. 1277- 1294, 2018.

[19] J. Li, H. Yuan and Y. Zhang, “Certificateless Public Integrity Checking of Group Shared Data on Cloud Storage”, IEEE Transactions on Services Computing, vol. 14, no. 1, pp. 71-81, 1 Jan.-Feb. 2021.

[20] H. Tian, F. Nan, H. Jiang, C. Chang, J, Ning and Y. huang, “Public Auditing for Shared Cloud Data with Efficient and Secure Group management”, Information Sciences 472, pp. 107-125, 2019.

[21] 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.

[22] Y.Ming and W. Shi, “Efficient Privacy-Preserving Certificateless Provable Data Possession Scheme for Cloud Storage”, IEEE Access, vol 7, pp. 122091-122105, 2019.

[23] R. Rabaninejad, M. Attari, M. Asaar and M. Aref, “A Lightweight Auditing Service for Shared Data with Secure User Revocation in Cloud Storage”,IEEE Transactions on Services Computing, 2019.

[24] W. Shen, J. Qin, J. Yu, R. Hao and J. Hu, “Enabling Identity-based Integrity Auditing and Data Sharing with Sensitive Information Hiding for Secure Cloud Storage”, IEEE Transactions on Information Forensics and Security”, Vol. 14. No.2, pp. 331-346, 2019.

[25] 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.

[26] Y. Yu, Y. Li, B. Yang, W. Susilo, G. Yang and J. Bai, “Attribute-based Cloud Data Integrity Auditing for Secure Outsourced Storage”, IEEE Transactions on Emerging Topics in Computing, vol. 18, No.2, pp. 377-390, 2020.

[27] Y. Zhang, J. Yu, R. Hao, C. Wang and K. Ren, “Enabling Efficient User Revocation in Identity-based Cloud Storage Auditing for Shared Big Data”, IEEE Transactions on Dependable and Secure Computing, Vol. 17, No. 3, pp. 608-619, 2020.

[28] J. Gudeme, S. Pasupuleti and R. Kandukuri, “Attribute-based Public Integrity Auditing for Shared Data with Efficient User Revocation in Cloud Storage”, Journal of Ambient Intelligence and Humanized Computing 12, pp. 2019-2032, 2020.

[29] N. B Gayathri, G. Thumbur, V. P. Reddy, and Md, Z. U. Rahman, “Efficient Pairing-free Certificateless Authentication Scheme with Batch Verification for Vehicular Ad-hoc Networks”, IEEE Access, Vol. 6, pp. 31808-31819, 2018.

[30] K. H. Vamshi, and G. Swain, “Identification and avoidance of malicious nodes by using certificate revocation method”, International Journal of Engineering and Technology (UAE), Vol. 7, No. 4.7, pp.152-156, 2018.

[31] K. Zhao, D. Sun, G. Ren and Y. Zhang, “Public Auditing Scheme with Identity Privacy Preserving based on Certificateless Ring Signature for Wireless Body Area Networks”, IEEE Access, Vol. 18, pp. 41975-41984, 2020.

[32] X. Yang, M. Wang, T. Li, R. Liu and C. Wang, “Privacy-Preserving Cloud Auditing for Multiple Users Scheme with Authorization and Traceability”, IEEE Access, Vol 8, pp. 130866-130877, 2020.

[33] J. Ni, K. Zhang, Y. Yu and T. Yang, “Identity-based Provable Data Possession from RSA Assumption for Secure Cloud Storage”, IEEE Transactions on Dependable and Secure Computing, 2020.

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

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

[36] 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 Tractions on Information Forensics and Security, Vol.10, No.3, pp.676-688, Mar.2017.

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

[38] S. Chaudhari and G. Swain, “Efficient and Secure Group based Collusion Resistant Public Auditing Scheme for Cloud Storage”, International Journal of Advanced Computer Science and Applications, Vol.12, No. 3, pp. 472-481, 2021.

[39] B. Dhote, & Krishna Mohan G.,“Trust and security to shared data in cloud computing: Open issues”, Advances in Intelligent Systems and Computing, Vol. 870, pp.117-126, 2019.

[40] B. Tirapathi Reddy, C. Rao and M. V. P., “Privacy preserving proof of ownership for data in cloud storage systems” International Journal of Engineering and Technology (UAE), Vol. 7, No. 2.8, pp.13-17, 2018.