RESEARCH ARTICLE OPEN ACCESS

Efficient Privacy Preserving Integrity Checking Model for Cloud Data Storage Security

Dr. Madhu B G[1], Dr. Naveen Kumar B [2], Ms. Sinchana J [3], Mr. Srinath K S [4]
[1]Department of Computer Science and Engineering Adichunchanagiri Institute of Technology
[3]Philips innovation campus
[2]Department of Information Science and Engineering Sahyadri College of Engineering
[4]Department of Computer Science and Engineering Sahyadri college of Engineering

ABSTRACT:
Cloud computing is one of the significant development that utilizes progressive computational power and upgrades data distribution and data storing facilities. With cloud information services, it is essential for information to be saved in the cloud and also distributed across numerous customers. Cloud information repository is involved with issues of information integrity, data security and information access by unapproved users. Hence, an autonomous reviewing and auditing facility is necessary to guarantee that the information is effectively accommodated and used in the cloud. In this paper, a comprehensive survey on the state-of-art techniques in data auditing and security are discussed. Challenging problems in information repository auditing and security are presented. Finally, directions for future research in data auditing and security have been discussed.

***KEYWORDS:* SHA-512, cloud security, data security, RSA algorithm, Merkley hash tree, Bcrypt algorithm, Blowfish**

Date of Submission: 15-01-2021 Date of Acceptance: 30-01-2021

I. INTRODUCTION

Cloud computing as the definition given by National Institute of Standards and Technology (NIST) is “an enabling ubiquitous model, provides us network access based on demand and offers a convenient aid of collection of configurable resources like servers, networks and storage applications, various services, which has the ability to be quickly provisioned and released with minimum effort from service provider interface or minimal management effort”. Cloud computing is based on accessing resources via the Internet.

The three common cloud service models offered by cloud is software as a service (SaaS), Infrastructure as a service (IaaS) and Platform as a service (PaaS). Out of this, Cloud storage is the main important service offered by cloud computing, ithelps users to move their data in to the cloud from local storage systems. It is an easy and cost effective way to store and manage the data. Drop box and GoogleDocs is an example of cloud storage system and it now becomes the essential feature in storage offerings.

Providing security for data is a major concern in cloud storage systems even though it comes with attractive benefits. The internal and external threats cause the data in cloud to be deleted or corrupted or tampered. In specific any external adversary tries to alter the content of the stored data and convinces the owner of the data that their data stored in cloud is correct and intact. This is being done for high profit. Hence it becomes essential to verify the correctness and integrity of the outsourced data moved to cloud.

In fig 1.1 the cloud when combined with “computing”, the meaning gets bigger and fuzzier. Some analysis and vendors define cloud computing narrowly as an upload as an updated version of utility computing: basically virtual servers available over the Internet. Others go very broad, arguing anything we consume outside the firewall is “in the cloud”, including conventional outsourcing.

Windows
Exchange Server
Your Applications
Office
Security
Data backup
Office
Home
Anywhere

Figure 1: Cloud Environment

Our focus here would be to create a Secure Cloud Ecosystem that leverages from the benefits of both symmetric and asymmetric encryption. We make use of RSA (Asymmetric) and AES (Symmetric) algorithms for carrying out data encryption. We aim at creating a comprehensive Cloud Environment that has security measures at all levels from creating and storing username and password, multifactor authentication, transmission of user data and data encryption.

II. LITERATURE SURVEY

Several schemes and auditing protocols have been introduced for protecting the integrity of outsourced data [8- 17]. Two schemes PDP [8] and POR [9] were introduced initially to cross check the integrity of outsourced data. They followed public auditing method on RSA based homomorphic linear authenticators. Ateniese et.al [10] provided scalable PDP to work with dynamic datas such as insertion, updation and deletion based on Hash functions and encryption technologies. It does not allow block insertion and has a fixed bound on no of challenges generated.

Erway [11] and Jules [9] provided dynamic auditing protocols to guarantee possession of data and data is retrievable. They do not consider the case of updation operation. Shacham and waters [12] proposed a solution for the above based on pseudorandom functions, BLS signature and their protocol supports public auditing. T.subha and S.Jayashri [13] described a method to perform dynamic data operations and their scheme supports public auditing by introducing TTPA (trusted third party auditor) in hybrid cloud. Wang et al described methods to achieve storage correctness, public auditability, privacy-preserving, batchauditing, error localization, error recovery and dynamic data operations.

In all the above schemes the cloud user delegates the integrity checking process to a third party auditor (TPA). An auditor is capable and trusted entity is allowed to audit the integrity of the outsourced data in cloud server. Existing schemes mostly focused on verifying the integrity of the data stored in remote servers.

Wang et.al [14-16] focuses on privacy preserving, which is an important property during the verification process. An adversary alters the cloud data arbitrarily and is able to provide a strong and valid proof in order to pass the verification process.

Adversary fools the auditor by generating valid proof and ensures the data owner to believe the data is well maintained and stored by the cloud server and integrity is preserved. Actually the data has been corrupted by the adversary [17].

Adversary needs to remember only how the data is altered, do not need to register the tags or content of the data. In this paper, we try to suggest a solution to resolve this problem by maintaining the features of original scheme proposed in Oruta [15].

III. METHODLOGY

This section consists of methodology of the propose system. Figure 2 depicts the entire architecture of the proposed method. The system architecture comprises of various physical entities that constitute the entire ecosystem. Here we would be talking about all different actors that constitute the Cloud, their roles, basic functionalities and the security services which our system provides. The following figure exemplifies our system architecture.

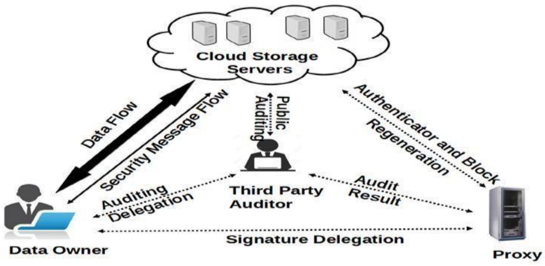


Figure 2: The system architecture

The list of security services which our Secure Cloud Ecosystem ensure are:

➢ **Trusted Authentication**: Only a legitimate user will be allowed to access services and data being hosted onCloud.

➢ **Authorization**: The system ensures proper authorization by only allowing system admin to have access to decryption keys. It is only the Cloud admin who is aware of the salted value added to every user password and Decryption Key before being saved in the database.

➢ **Data Encryption**: The system makes use of Hybrid Encryption by allowing RSA algorithm to encrypt user data. The proposed system leverages the benefits of both symmetric and asymmetric data encryption. We make use of RSA2048 for our encryption process.

➢ **Hashing**: SHA512 and b-crypt functions are used for securing user password.

➢ **Key Management**: The Private Key is encrypted and salted and safely stored into the database. The decryption keys are also saved soon after the encryption gets over. The SHA512 key is protected using keyed-hash message authentication code (HMAC).

IV. ALGORITHMS USED FOR PRESERVING DATA INTEGRITY

In order to provide security for the data stored first step is to generate keys for each file uploaded, when a file is uploaded user is provided with a pair of private key and public key and in the second step we can check for the integrity of the cloud by sending an auditing request to a third part auditor different algorithms used here are:

- RSA algorithm
- Merkley hash tree
- Bcrypt algorithm
- Blowfish algorithm

a. RSA algorithm

RSA algorithm is mainly used for key generation the different steps in this algorithm is given below:

1. Choose two distinct prime numbers p and q.

- For security purposes, the integers p and q should be chosen at random, and should be similar in magnitude but differ in length by a few digits to make factoring harder. Prime integers can be efficiently found using a primality test.

2. Compute n = pq.

- n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.

3. Compute □(n) = gcd(φ(p), φ(q)) = gcd(p − 1, q − 1), where □ is Carmichael's totient function. This value is kept private.

4. Choose an integer e such that 1 < e < □(n) and gcd (e, □(n)) = 1; i.e., e and □(n) are co prime.

5. Determine d as d ≡ e^{-1} (mod □(n)); i.e., d is the modular

multiplicative inverse of e modulo □(n).

- This means: solve for d the equation d*e ≡ 1 (mod □(n)).
- e having a short bit-length and small Hamming weight results in more efficient encryption – most commonly e = 2^{16} + 1 = 65,537. However, much smaller values of e (such as 3) have been shown to be less secure in some settings.
- e is released as the public key exponent.
- d is kept as the private key exponent.

b. Merkley hash tree

For the traversal techniques, we need an algorithm, that computes efficiently the nodes of the tree. Assume a binary tree with 2n leafs. The height H of a node, is defined by the distance of the node to a leaf. So the root has the height H = n, while the leafs have the height H = 0. We define the node a, i,j as the jth node from the left (starting with j = 0) of the height i. So a0,0 is the leftmost leaf of the tree, and an,0 the root. To compute a node of the height H = h, 2h −1 nodes must be computed. The tree hash algorithm needs 2h−1 operations, to calculate a node of the height h, while saving as few nodes at once as possible. The main idea of the treehash algorithm is to calculate the needed subtree from left to right and only saving the nodes, that are still needed.

This is done, by using a stack. At first the stack only consists of the leftmost leaf. Then the next leaf is added. The algorithm now checks whether the last two nodes on the stack are of the same height or not. If they are of the same height, the two nodes are removed from the stack, and their parent is built and pushed on the stack. If the last two nodes on the stack are of different height, then a new leaf is pushed on the stack. This step is repeated, until the node of the wanted height has been generated.

Algorithm: TREEHASH (start, maxheight)

```
1.  Set leaf = start and create empty stack.

2.  Consolidate: If top 2 nodes on the stack are equal height:

    •   Pop node value P(nright) from stack..
    •   Pop node value P(nleft) from stack..
    •   Compute P(nparent) = f(P(nleft||P(nright)).
    •   If height of P(nparent) = maxheight, output P(nparent).
    •   Push P(nparent) onto the stack.
3.  New Leaf: Otherwise:

    •   Compute P(nl) = LEAFCALC(leaf).
    •   Push P(nl) onto the stack. • Increment leaf.
4.  Loop to step 2.
```

c. Bcrypt algorithm

The bcrypt algorithm runs in two phases In the first phase, Eks Blowfish Setup is called with the cost, the salt, and the password, to initialize eks blowfish's state.

Most of bcrypt’s time is spent in the expensive key schedule. Following that, the 192-bit value “Orphean BeholderScry Doubt" is encrypted 64 times using eksblowfish in ECB mode with the state from the previous phase. The output is the cost and 128-bit salt concatenated with the result of the encryption loop.

Algorithm:

```
1.  Bcrypt(cost,salt,pwd)
2.  State□EksblowfishSetup(cost,salt,key)
3.  ctext□“OrpheanBeholderScryDoubt"
4.  repeat(64)
5.  ctext□EncryptECB(state,ctext)
6.  return Concatenate(cost,salt,ctext)
```

d. Blowfish

Blowfish is a variable-length, a new secret-key block cipher. It is a Fiestal network, iterating a simple encryption function 16 times. Its main features are:

- Block cipher: 64-bit block.
- Variable key length: 32 bits to 488 bits.
- Much faster than IDEA and DES.
- Unpatented and royalty free.
- No license required

Subkeys:

Blowfish uses a large number of sub keys. These keys must be precomputed before any data encryption or decryption.

1. The P-array consists of 18 32-bit sub keys:

P1, P2,..., P18.

2. There are four 32-bit S-boxes with 256 entries each: S1,0, S1,1,..., S1,255;
S2,0, S2,1,..,, S2,255;
S3,0, S3,1,..., S3,255;
S4,0, S4,1,..,, S4,255.

Flowchart for privacy preserving model

Flowcharts represents the steps in which the model works and this model has two different flowcharts for users and auditor.

In fig 3 user description flow chart is explained at first the user login or register into the cloud once the credentials are met they can upload the file or download the file or request the auditor for integrity check of the file.

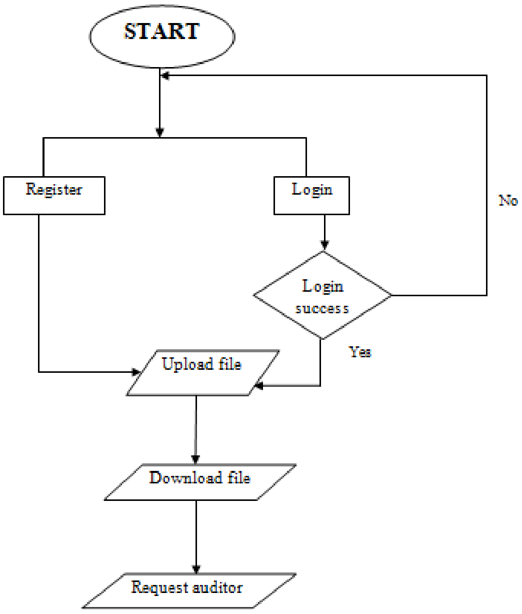


Figure 3: User description flowchart for Privacy Preserving Model

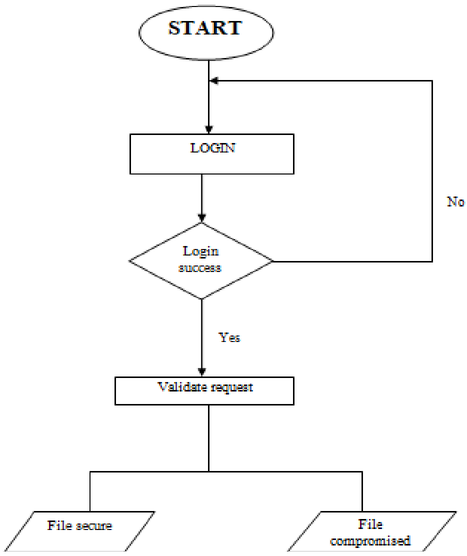


Figure 4: Auditor description flowchart for Privacy Preserving Model

Fig 4. describes the auditor flow chart at first the auditor login into the cloud and validate all the requests and give the result whether the file is secure or compromised.

V. EXPERIMENTAL RESULTS

In this section we discuss the results obtained from the simulation of web service discovery process. The Figure 5 shows the login page of the Efficient Privacy Preserving Integrity Checking Model. Login page gives the interface for User and Auditor to login into the system by providing username and password.

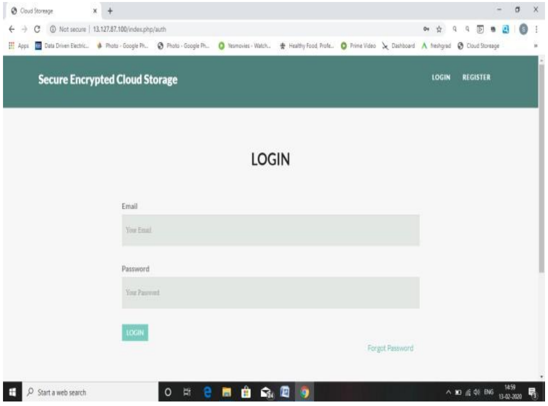


Figure 5: Login Page

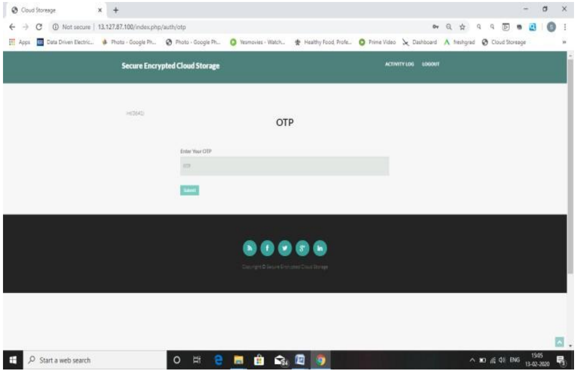


Figure 6 : Login successful

Figure 6. depicts successful login into the system of user module. When the username and password is correct the users get OTP for the registered phone number.

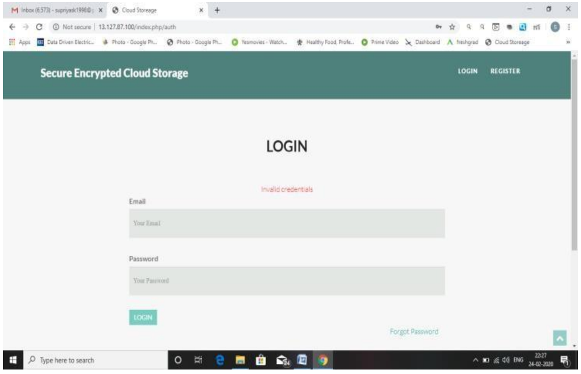


Figure 7: Login Unsuccessful

Figure 7 shows unsuccessful user login. This happens when username and password are in correct.

Figure 8 depicts the registration form for the new users. In which the user must provide the credentials.

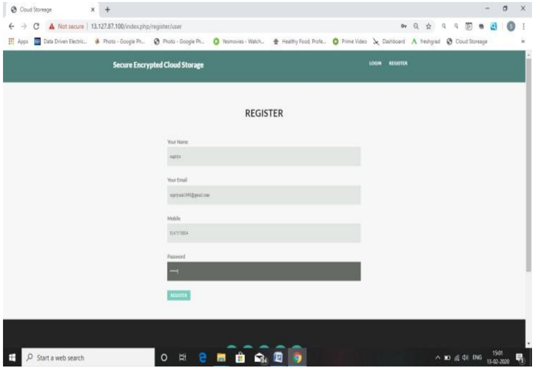


Figure 8: Register

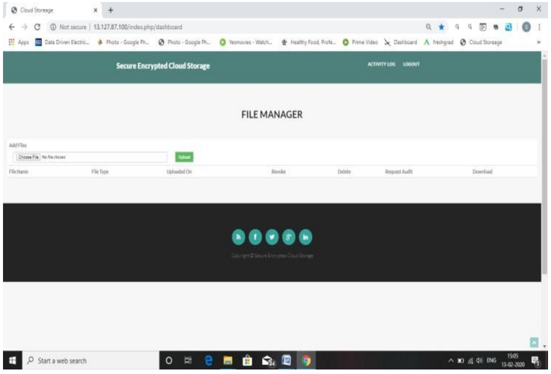


Figure 9: File Manager

Figure 9. shows the file manager. This page shows all the activities that can be performed by the user.

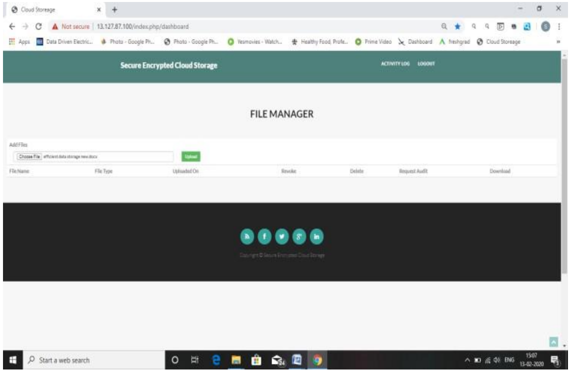


Figure 10: File upload

Figure 10. shows the File upload option for the user. User has to select a particular file that must be uploaded to the cloud.

Figure 11. shows the Download feature provided to the users. Here the user can download the selected file by providing the perfect private and public keys which are downloaded when the file is uploaded.

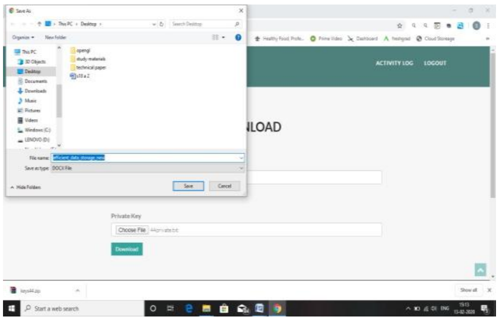
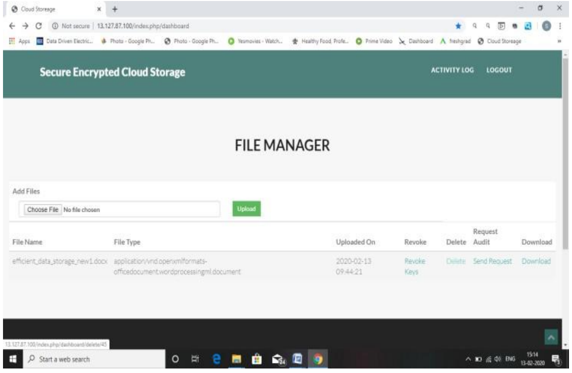


Figure 11: Download File Successful

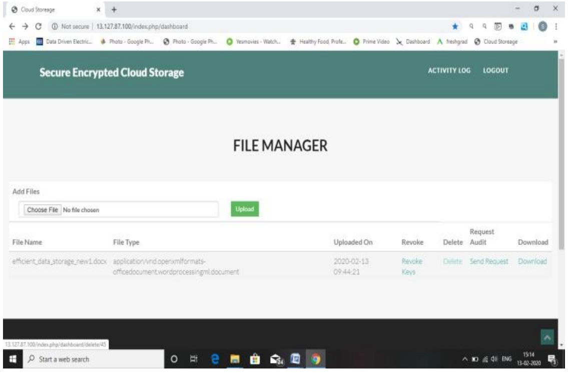


FILE MANAGER

File Name	File Type	Uploaded On	Revoke	Delete	Request Audit	Download
efficient_data_storage_new1.docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document	2020-02-13 09:44:21	Revoke Keys	Delete	Send Request	Download

Figure 12: Delete File

Figure 12 shows the delete feature of the application. Here user can delete any selected file from the cloud. This leads to permanent deletion of the file and cannot be restored.

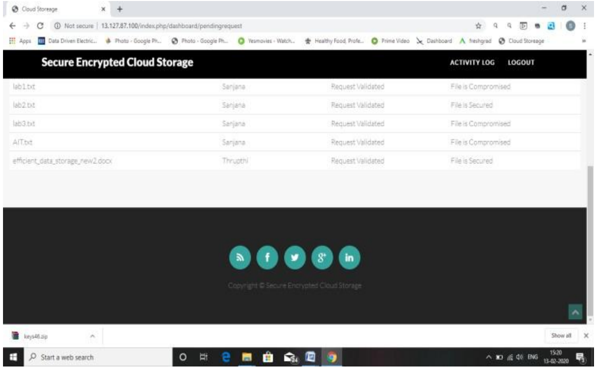


FILE MANAGER

File Name	File Type	Uploaded On	Revoke	Delete	Request Audit	Download
efficient_data_storage_new1.docx	application/vnd.openxmlformats-officedocument.wordprocessingml.document	2020-02-13 09:44:21	Revoke Keys	Delete	Send Request	Download

Figure 13: Revoke keys

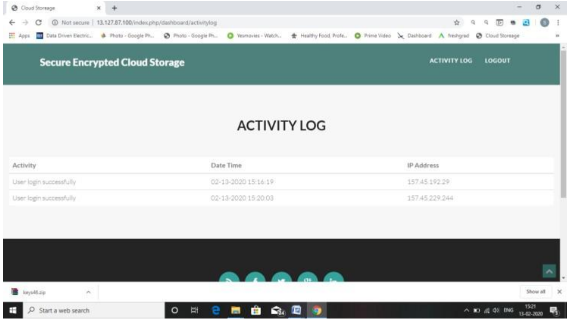
Figure 13 shows the revoke key feature of the application. The user can use this feature to revoke the private and public keys for a particular file. This enhances the security of the file in the cloud.


Cloud Storeage
Not secure | 13.127.87.100/index.php/dashboard/pendingrequest
Secure Encrypted Cloud Storage
ACTIVITY LOG
LOGOUT
Copyright © Secure Encrypted Cloud Storage
keys46.zip
Show all
Start a web search

lab1.txt	Sanjana	Request Validated	File is Compromised
lab2.txt	Sanjana	Request Validated	File is Secured
lab3.txt	Sanjana	Request Validated	File is Compromised
AIT.txt	Sanjana	Request Validated	File is Compromised
efficient_data_storage_new2.docx	Thrupthi	Request Validated	File is Secured

Figure 14: File manager for auditor

Figure 14 shows the file manager page of the auditor. When the user sends a validation request for the auditor. The request is shown in this page and this request is validated by the auditor. If the file is secured the file secured message is displayed. In case of hacking then file compromised message is displayed.


Cloud Storeage
Not secure | 13.127.87.100/index.php/dashboard/activitylog
Secure Encrypted Cloud Storage
ACTIVITY LOG
LOGOUT
keys46.zip
Show all
Start a web search

ACTIVITY LOG

Activity	Date Time	IP Address
User login successfully	02-13-2020 15:16:19	157.45.192.29
User login successfully	02-13-2020 15:20:03	157.45.229.244

Figure 15: Activity log for auditor

Figure 15 shows the activity log page of the auditor. This page keeps track of all the activities performed by the auditor and also the IP address from which he has logged in.

VI. CONCLUSION

we present an extensive survey on data auditing and security in distributed computing. With data storage and shared data, auditor performs efficient auditing with group user revocation. Existing mechanisms provide efficient integrity auditing of shared data, user revocation and supports batch auditing. Mechanisms need to be implemented to reduce the overhead introduced by a huge number of customers in the cluster. In public auditing for collaborative information, the auditor performs efficient auditing and also preserves confidentiality of the shared information saved in cloud. The schemes preserve the privacy of the identity of the customer and supports batch auditing.

REFERENCES

[1]. P.Mell and T.Grance, “The NIST Definition of Cloud Computing”, NIST Special Publication - 800145, 2011; http://csrc.nist.gov/publication/nistpubs/800-145.

[2]. M.Lillibridge, S.Elnikety, A.Birrell, M.Burrows and M.Isard, “A Cooperative Internet Backup Scheme,” proc. USENIX Ann.Technical Conf., pp.29-41, 2003.

[3]. Yong.Yu, Lei Niu, Guomin Yang, Yi Mu, Willy Susilo, “On the security of auditing mechanisms for secure cloud storage”, Future Generation Computer Systems 30(2014), 127-132.

[4]. Lee Cheng-Chi, Lai Yan-Ming, Hsiao Chin-Sung, Cryptanalysis of a simple key assignment for access control based on polynomial. J Inf Secur Appl 2013; 18(4):215-8.

[5]. Li H, Dai Y, Tian L, Yang H, " Identity-based authentication for cloud computing, Lecture Notes of Computer Science (LNCS), vol. 5931; 2009, p.157-166.

[6]. Yuan Zhang, Chunxiang Xu, Jining Zhao, Xiaojun Zhang, Junwei Wen, "Cryptanalysis of an Integrity checking scheme for cloud data sharing" , Journal of Information Security and applications (2015), I-6.

[7]. Dolev Danny, Yao Andrew C, "On the security of public key protocols. Inf Theory, IEEE Trans Inf Theory March, 1983; 29(2):198-208.

[8]. G.Ateniese, R.Burns, R.Curtmola, J.Herring, L.Kissner, Z.Peterson, and D.Song, "Provable data possession at untrusted stores,"in Proc.of CCS'07. Newyork, NY, USA: ACM, 2007, pp.598-609.

[9]. A.Juels and B.S.Kaliski, Jr.,"Pors: proofs of retrievability for large files," in Proc.of CCS'07. Newyork, NY, USA: ACM, 2007, pp.584-597.

[10]. G.Ateniese, R.D.Pietro, L.V.Mancini, G.Tsudik, "Scalable and efficient provable Data possession," in: Proc. of SecureComm 2008, pp. 1-10.

[11]. C.C.Erway, A.Kupcu, C.Papamanthou, R.Tamassia, "Dynamic provable data possession," Proc. of CCS2009, pp.213-222.

[12]. H.Shacham and B.Waters, "Compact proofs of retrievability," in Proc. of ASIACRYPT'08. Melbourne, Australia: Springer-Verlag, 2008, pp.90-107.

[13]. T.Subha and Dr.S.Jayashri, "Data Integrity Verification in hybrid cloud using TTPA," Lecture Notes in Electrical Engineering 284,Springer, pp 149- 159.

[14]. Q.Wang, C.Wang, J.Li, K.Ren, and W.Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. of ESORICS'09. Saint Malo, France: Springer- Verlag, 2009, pp.355-370.

Dr. Madhu B G, et. al. "Efficient Privacy Preserving Integrity Checking Model for Cloud Data Storage Security." *International Journal of Engineering Research and Applications (IJERA)*, vol.11 (1), 2021, pp 38-44.