Auditing over secured cloud data with data dynamics and data sharing

MSc Research Project
Cloud Computing

Pruthviraj Bhamare
Student ID: x20190182

School of Computing
National College of Ireland

Supervisor: Sean Heaney

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Pruthviraj Bhamare
Student ID:	20190182
Programme:	Cloud Computing
Year:	2022-23
Module:	MSc Research Project
Supervisor:	Sean Heeney
Submission Due Date:	15/12/2022
Project Title:	Data Auditing over secured cloud data with data dynamics and data sharing
Word Count:	3556
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	15th December 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Data Auditing over secured cloud data with data dynamics and data sharing

Pruthviraj Bhamare
20190182

Abstract

Nowadays, practically every internet user either directly or indirectly utilizes cloud services, such as Gmail or Google Drive. Professional cloud service customers are willing to save their data in cloud storage systems. However, during the same period, there is an increase in the requirements for the accessibility, confidentiality, and incorruptibility of data in cloud storage with the data operations. The data storage expense is too more, creating it hard for the files to execute data operations like removal and addition and sharing efficiently while many of the techniques are presented, despite the fact that various cloud data auditing approaches have been published. The developed system is using NR-MHT to execute data operations and conduct integrity checks of data in order to address these problems. The system uses PDP approach to assure that the data in cloud storage is authentic. The design of the developed model does not include a external auditor which is one of the achievement of this study. Instead of TPA, Key server is introduced. In the developed system, sharing of files and revocation of users are handled via (CP-ABE) algorithm. In this developed system data owner can perform the data operations and can share data with the other users of the desktop application on cloud storage. Additionally, we carry out a comparison between existing system to formally confirm that the developed system is capable of meeting all objectives of the proposed system. Final analysis of trial findings reveals that the system is quite efficient with upload time of files, cost communication and data auditing time as it reduces the overhead and time with extra features compared to previous schemes.

1 Introduction

The importance of cloud computing is grown and grown everyday since it has been established in the market of IT industry. Each day new data is generating and to handle that data new technology or products must be needed because security of the data is a huge concern in the industry. Several enterprises have relieved the stress of local file storage and administration by moving data from locally to cloud. Users who has their information or files on cloud still have severe worries about maintaining the confidentiality, security or integrity of the transferred data, though, owing to the lack of power and less access over the data. Although cloud storage has numerous advantages, it is inexorably hampered by major security problems Chen et al. (2014). Particularly, rights and controlling of transferred data are divided, denying data holder direct access control on their data. It results, all activities involving their outsourced information is managed by the cloud server. The cloud server frequently abides by rules negotiated with the data holders. The server, meanwhile, has a number of motivations for acting inappropriately in ways that benefit from it. Furthermore, the server could consciously remove few transferred blocks of data which are used very few times in an effort to profit from doing so, or it might fraudulently hide data errors which are difficult for users to find.

Further, since this requires more memory and likely to have a detrimental affection on the storage services given to another owners or users, the cloud server is not able to promptly add the new blocks of data since it has been called by the owners. Not to mention, the cloud server could purposely defy the request to delete the information and keep certain versions of the transferred data in order to find untapped commercial potential. In different terms, while using cloud storage to store information is cost-effective, moreover ineluctably raises serious security issues for the data users, including issues with data authenticity, data addition, data removal, and data exchange Yang et al. (2022).

Previous models, but not NR-MHT, presented the notion of data dynamics using various types of algorithms Yang et al. (2022). However, their research does not handle all forms of data actions, such as data sharing. They developed the notion of data removal and addition on Number Ranked-Merkel Hash Tree.

The issue of integrity and confidentiality is addressed by the secure encryption technique. Data integrity is the assurance that information is accurate and reliable. An inspector, a third-party auditor or server, acting on user's behalf, may be given the responsibility of file integrity checking. Sending data to other users of the cloud is a rising tendency in cloud storage. Problems with data security and access control occur when particularly sensitive information is uploaded to the cloud.

1.1 Research Question

There have been instances where sharing data with outside auditors has raised significant safety concernsTiwari and Gangadharan (2018). Because of this removal of external auditor from the scenario of data management in the cloud is the real case and has a need to study more thoroughly. Reminding the importance of data security in data management with the sharing of data which led to a research question, "How to provide efficient data auditing over secured data along with data dynamics and user sharing ?"

In this work, the proposed or developed system is solving the issue of third-party auditor by removing it and also developed data management desktop-based application which provides a data security and integrity based on the NR-MHT. This auditing system requires less processing power for generating the audits.

The objectives of developed application are the activities on the data, such as user can upload data on cloud that is insertion, delete data from the cloud storage, and able to share data with other group members or users of the application in order to keep it safe. No data will be handled by a Third-Party Auditor. Instead of it the system has Key-Server which manages the keys of every group or user for a file. The system also uses PDP, or provable data possession, to ensure that the data's integrity is upheld or not.

The CP-ABE is used for data encryption and sharing of files or folders with the multiple users or groups in cloud. A user's authorization will be revoked if they act improperly or participate in any problematic behaviour within a group of cloud storage. An CP-ABE-based user revocation technique is included in the application.

The major contribution of this research is that the developed approach would lessen processing complexity, provides data integrity checking and enabling data dynamics including file sharing, removal, modification and make actual cloud-based data sharing situations feasible.

1.2 Report Structure

The other sections of this work are as follows. In Section 2, we present the literature work. The topic statements and methods are then presented in precise detail in Section 3. Then, in Section 4, we demonstrated how user sharing with user revocation and data auditing over protected data are accomplished. The implementation is put into place in Section 5. In Section 6, we performed experiments on the system's performance in comparison to the existing system, and Section 7 draws this research to a conclusion.

2 Related Work

Information transmission between different terms like dependability, safety, and tampering are just a few of the terms used to describe "data integrity." Data possessors who outsource their data lose direct physical control over it since the management and possession of the data are separated. As an outcome, the cloud server managing all processes involving transferred information. Data integrity checks are a growing trend in the field of outsourced information protectionWang et al. (2021).

Following is a list of areas for this survey:

1. Integrity and Data Security.
2. Auditing Data
3. Data Operations
4. User Revocation
5. Data Integrity Auditing with MHT

2.1 Integrity Auditing

In the area of Integrity and Data Security, several works have been done in the past but in the 2019, Garg et al. (2020) developed an effective audit protocol for cloud storage. This method aims to decrease client computation complexity during the beginning phase of the protocol's system. The standard provides dynamic types of actions and is publicly verifiable because its foundation is the properties of bilinear pairings. This standard provides verification for publicly available data. By simulating a contest between the defender and the opponent, this method has been shown to be secure in ROM on the basis of CDHP stabilization.

A digital signature algorithm by Kavin and Ganapathy (2021) for utilizing elliptic curves to maintain the integrity in the cloud.This work offers EDSA to ensure the integrity checks of data while it is stored on a cloud server. The proposed EDSA is constructed using the Elliptic Curves generated by using an improved equation. The proposed EDSA additionally generates two elliptic curves by employing the enhanced equations in this system. These elliptic curve points served as a public key for transaction signature and validation. The results of the many tests conducted demonstrated the efficiency of the recommended EDSA in terms of key generation, signature, and verification time.

The validation and signature processes in this method are administered and maintained by a third-party auditor. The cloud user shouldn't be worried because the documentation won't be replicated or kept by an outside auditor. Unauthorized users cannot access any papers without the cloud user's consent; the TPA is the only party in charge of document verification. Due to the system's usage of Third-Party Auditor, which makes it possible for assaults on the platform or the data, this new study on data integrity also has certain flaws.

2.2 Data Auditing

An auditing scheme developed by Zhang et al. (2021) to figure out the mistakes. This is a blockchain based system. To enable customers to confirm the accuracy of their transferred data in cloud storage environment auditing is still desired. The bulk of approaches, however, rely on trustworthy institutions like the cloud platform planner and TPA, making it difficult to identify dishonest service carriers after service disputes.

The purpose of this blockchain-based data auditing system is to ensure data security and effectively address service concerns. They employ smart contracts to detect service conflicts and the blockchain to log interactions between users, providers, and administrators during the data auditing as evidence, which compels dishonest organizations to publicly identify dishonest service providers. The blockchain system with homomorphic valid tags enables minimal cost batches identification without TPA. This blockchain-based data auditing technique was created to get accurate audits in cloud storage that lacks TPA and prevent bad organizers. This was a novel strategy to safeguard data integrity auditing in cloud storage, but it has certain drawbacks, including high computational complexity and a lack of capabilities for intelligence sharing.

2.3 Data Operations

A system in which data used to share with the limited number of users in a cloud network developed by Li et al. (2018). The system uses ABE algorithm. To handle complex issues of mobile users with less number of resources this algorithm is used. This approach eliminates the majority of the analysis work by adding technique public parameters in addition to shifting the calculations for partial encryption offline. An open cipher-text testing stage is conducted ahead of the decoding stage, which removes the huge numbers of the computational cost brought on by incorrect encrypted message.

A new system named A2 B2 C2 system which gives individuals whose accounts are specified in a certain receiver set the ability to view the related ciphertext without violating the access policy, and it also has the functionality of disguised access policy dveloped by Xiong et al. (2018).

Every PHR giver is given a collective identity and a list of characteristics, and the shared information is encoded in line with the stated access policy and set of group IDs using their proposed A2 B2 E2. The shared information can only be decoded by users who have the appropriate access rights and are a part of the selected group. Furthermore, the access control guidelines associated with the encrypted message will not be disclosed by anybody, not even the approved PHR data receivers.

2.4 User Revocation

A real requirement for common data monitoring in cloud-based storage systems is user revocation. Whenever members behave improperly or quit the group, they must be thrown out.

A simple auditing solution with safe revocation for shareable data stored in the cloud developed by Rabaninejad et al. (2019). Researchers developed an open shared data monitoring approach and applied a novel proxy re-signature mechanism in this system to safeguard user privacy and thwart collusion. The only user side computations required by their method are basic ones for online stage data block signing. Furthermore, their system enables functions with dynamic data and user groups.

Regardless of how often users distribute the data blocks, their system's communication and computation costs at the host and TPA sides stay stable in contrast to earlier techniques for auditing shared information.

Several open auditing solutions are built on asymmetric key infrastructure or cryptography have been proposed to confirm the correctness of information which has been outsourcing. However, they have difficulties with key management. Finding a method for achieving user data privacy and revocation is another problem in multiple user sharing auditing.

In attempt to deal with these problems, AB based open audit system for shareable files in cloud is developed by Gudeme et al. (2021). In this approach, people who sign data blocks while not disclosing their names, and an exclusive asymmetric key instead of each user's personal asymmetric keys are used to check the integrity. The approach also intended to implement user revocation through proxy re-signatures.

2.5 Data Integrity Auditing with MHT

For data readjustment and validation, a particular type of data architecture is used: the Merkle hash tree. The data model of a tree comprises non-leaf nodes, each of which is a hash of the endpoints it includes as descendants. Each leaf node is similarly long and as further to the left as it is possible to be. Hash functions are used to maintain the file's authenticity. Because of fast development of the storage infrastructure, users are eager to keep records there, though at the same moment, the requirements for the safety, authenticity, and accessibility of file storage are increasing. Ignoring the fact that a number of cloud auditing techniques have been proposed, efficient dynamic data updating is hindered while the majority of systems are running because of the inefficiency involved with data storage.

Yang et al. (2022) created a method for erasing files of information from cloud service that can be shown to have taken place. The challenges of verified removal, flexible addition, and fidelity checks of outsourced data were discussed in this study. A number-rank-based Merkle hash tree (NRMHT), which researchers created, is a novel verification data model that can operate on dynamic data. This is an existing system which gives an area of research to proposed solution.

Because NR-MHT will keep many blocks of data in each new node, the concern of the length growing linearly with the total number of data blocks may be effectively resolved. The approach utilizes NR-MHT to give a new data removal technique that relies on dynamic data addition and successful data integrity audits. With this scheme, the data holder can delete any superfluous outsourced blocks of data forever to save inventory cost while also adding new blocks of data to effectively modify the outsourced data set. In the meanwhile, researchers looked at the safety to properly demonstrate that their proposed technique able to fulfill all objectives of security requirements with excluding a 3rd party auditor.

For the encryption of the data, the system uses AES algorithm. System Architecture of this existing system involves two entities, first is Cloud Server and other is Data Owner.

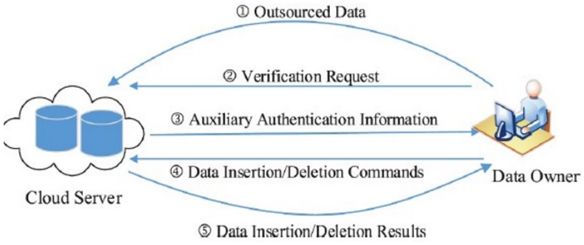

① Outsourced Data
② Verification Request
③ Auxiliary Authentication Information
④ Data Insertion/Deletion Commands
⑤ Data Insertion/Deletion Results
Cloud Server
Data Owner

Figure 1: The system architecture existing system

The entity that owns the files is one whose resources are so constrained that it is incapable of maintain and retain enormous volumes of files on local hard disks. Therefore, in order to drastically cut the demand for computation and local storage capacity, the individual who owns the information is eager to store the enormous volumes of information on a cloud platform. Then, the data owner could opt to freely add new files or delete unnecessary ones. Lastly, because users wouldn't trust the cloud service, data owners prefer to check the outcomes of data removal and additions.

A cloud provider has almost infinite computer power, networking speed, and huge storage. In order to help data holders with restricted assets, the cloud provider will provide pay when you need which is on-demand data storage solutions. Because of the separation of outsourcing of data administration and possession, the cloud server will handle addition and removal requests for the data holder in the period.

A brand-new system devoid of TPA developed byYang et al. (2022). However, despite all of this study and development, not all data dynamic actions, such as file sharing and user revocation, are fulfilled. This an existing system which lacks with these features also uses an AES algorihtm for encryption which is not efficient with the time. In some domains, we can perform research and create an architecture that includes user revocation, safe MHT-based data integrity audits, and all information dynamics activities. In order to accomplish data audit over protected cloud storage with data dynamics and data sharing, the developed system focuses on achieving this goal.

3 Methodology

The issue of integrity and confidentiality is addressed by the secure encryption technique. Data integrity is the assurance that information is accurate and reliable. To ensure about the files or data which has been outsourced on the cloud is stored in the encrypted format.

In this study, the desktop-based application uses two different algorithms to achieve the successful data integrity check of the data in data management which is on cloud. The First one is Merkle Hash Tree and the other one is CP-ABE.

The developed system is created using the Merkel Hash Tree (MHT), which checks the integrity of data or file which has been outsourced on the cloud. Different MHT systems, including HB+- MHT, batch-leaves-authenticated MHT, and NR-MHT, have been done in earlier work of other researchers. But this developed desktop application based on the Number-Rank-based Merkle hash tree (NR-MHT), which allows for data operations. Because NR-MHT allows for the maintenance of many data blocks at every leaf node, the problem of the height growing linearly with the entire number of data blocks effectively resolved. MHT ensured that Third-Party Auditor is not involved in order to carry out the audits in cloud storage. Data integrity audits takes less processing time using the developed auditing solution.

In NR-MHT, each node potentially store a large amount of data blocks. In the meanwhile, each leaf node's data block count could change. Thus, it may support operations involving dynamic data, such as data addition and deletion. Moreover, The amount of data blocks cannot indefinitely grow the height of NR-MHT. In order to flexibly vary the height of NR-MHT, the data holder must adjust the amount of data blocks saved by each leaf node in accordance with overall number of data blocks. As a result, the problem of data block growth leading to a fast rise in MHT height will be effectively resolved, keeping the height of NR-MHT within a suitable range. As a result, NR-MHT is particularly applicable for potential higher dynamic data storage systems Yang et al. (2022).

The proposed system follows the PDP approach as it verifies both static and dynamic files' data integrity. It is divided in two types.

1. Static PDP approach: The static PDP approach validates the cloud-based static data files. The static files forbid data change. This method can use in fields where there are no changes made to the original files, such as actual studies, library data, archival data, etc.

2. Dynamic PDP approach: The applications needs to support the features such as inserting, updating, and deleting data. The auditing approach permits data update while reducing user burden.

The second algorithm CP-ABE is used for the providing the feature of data sharing and user revocation. It is also used for the encryption. Revocation is carried through without affecting other users who haven't been removed thanks to the straightforward revocation architecture of CP-ABE.

The following is an explanation of the three primary goals of applying this technique for this system:

3.1 Data Integrity Auditing

The cloud server should accurately and fully maintain the outsourced data for the advantage of the data owner. The data owner can immediately discover if the blocks of outsourced data have been tainted thanks to data integrity audits.

3.2 Data Dynamic Operations

The user may input new data, edit current data, or remove data, among other tasks. The developed system can successfully do each of these tasks. These procedures on data dynamics assist in data audits. Since the files or information has been outsourced, the user needs an auditing system which allows them to check the correctness of their data without having to retrieve it. There are various information audit methods that have been created to date.

3.3 User Revocation

A user's access control will be revoked if they act improperly or participate in any suspicious behavior within a group of cloud storage. An CP-ABE-based user revocation technique is included in the system. User identification is used as a key generation characteristic in the CP-ABE system.

4 Design Specification

The developed system has the framework which consists of five entities and they are as follows:

1. Desktop Application
2. Cloud
3. Key Server
4. Data Owner
5. Data User

Desktop Application: This application will help user to store and share his/her data on the cloud having algorithms like NR-MHT and CP-ABE. Through this application audits will be generated and user management, group management can be done effectively. This application lets user store or share any kind of multimedia files on cloud with no limits of the size in terms of memory.

Cloud: The developed system uses Amazon EC2 instance. The cloud storage is provided to the user or data owner. Through the desktop application user can use the services of cloud with a guaranteed safety and integrity checks of user's data. Data and Audit information is stored on cloud which can be shared with the owner with the help of the application. Users within a group can view and download data which is stored on the cloud if they have an access to those groups or certain files.

Key Server: Within a centralized key management system, the Key Server generates user keys and group keys for the cloud. Additionally, it keeps the users list and alters or updates the key. These keys will be generated for each group and each user for the particular file which is uploaded by data owner. It has the most important role in this architecture as it removes the involvement of third-party auditor.

Data Owner: On the cloud, there are mainly two sorts of data. The first category is client data—that is, data that users generate on their own computers and then upload to the cloud for processing and archiving. The second kind of data is produced by providers on the cloud platform using a range of tools, such as software, secure encoding, etc.

In this system, data owner has the role of carry out the data integrity checks by checking audits of the particular file. Through the desktop application data owner can

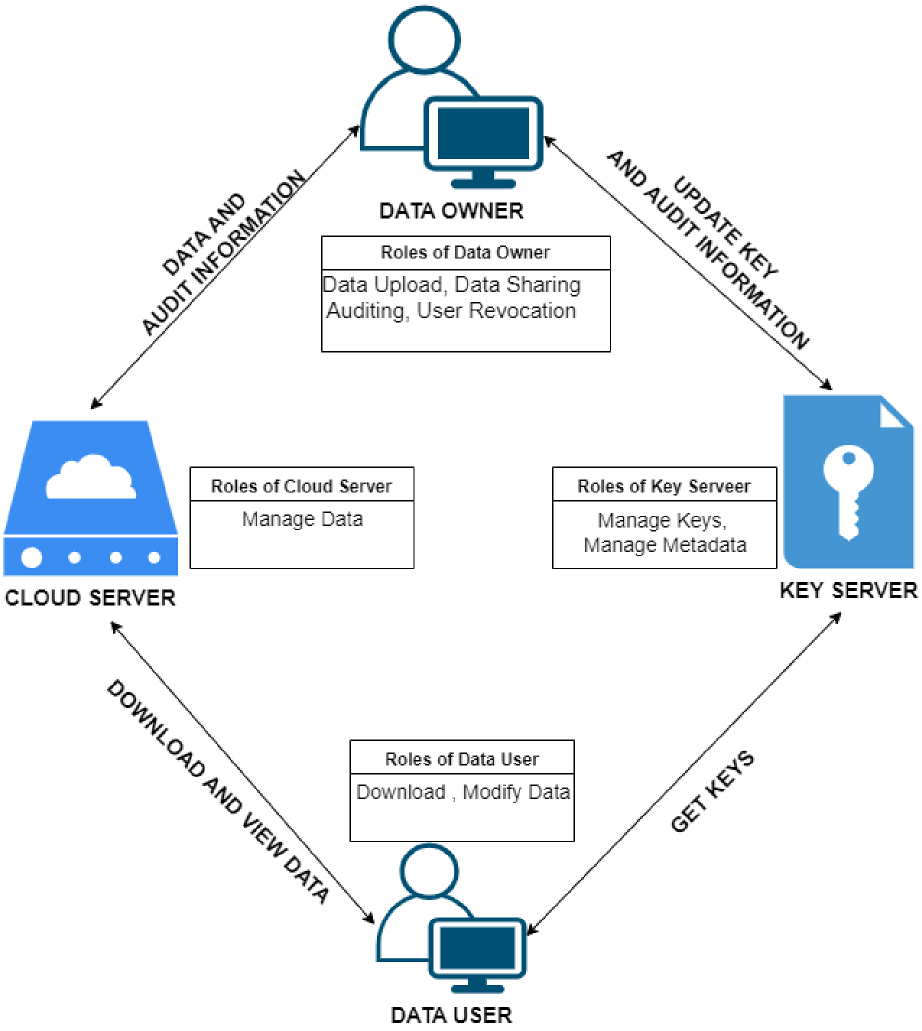


Figure 2: Architecture of the System

able to monitor the activities of the data user. By checking audits of the files, it is possible for data owner to identify the corrupted files.

Data User: The person who can access, modify, and download data from cloud storage is referred to as a data user. Data owners control the functions of data consumers. Data users can be the part of the groups which are created by the data owners.

Phases of developed system

The designed system's life cycle includes the following stages or modules:

4.1 Initialization

The system will register the data owners during this start-up phase. In addition, it creates groups of people. A system registration and the generation of relevant encryption key pairs like public and private keys for the folder which is created by data owner. It utilizes a key server for this stage.

- Registration of Data Owner/User
- Groups can be created
- Folder Creation
- Keys will be generated

The data owner first automatically generates the symmetric and asymmetric key pair and related asymmetric key certificate by creating folders

4.2 Data Outsourcing

Data Outsourcing works in the following way:

a. The data holder uploads the chunk of data which is about to transfer to the cloud server using an outsourcing algorithm.

b. The owner first creates an NR-MHT and gets a Merkle root using the transferred data set.

c. The owner then deletes the backup copy of the transferred set of data and sends the whole NR-MHT to the cloud server to reduce computational complexity and storage expenses.

The actual disks of the cloud server are subsequently used to store the transferred data set. Data uploading and tree creation are two aspects of data outsourcing.

4.2.1 Tree Creation:

For the purpose of maintaining the transferred data set, the data controller creates a number-rank-based Merkle hash tree (NR MHT). In further detail, Each leaf node in the owner's number-rank-based Merkle hash tree (NR MHT) contains a subfile.

4.2.2 Data Uploading:

The owner of the data transferring their data to the cloud. The data's creator creates a signature using the Merkle root. Every file or folder has a separate key that is saved in the key server upon data upload.

The data holder creates a signature on the Merkle root using the ECDSA signature creation technique with the Merkle root's hash value. Thereafter, the data holder assigns the cloud server full responsibility for the number rank-based Merkle hash tree (NR-MHT). The data holder only keeps the Merkle root as well as the signature on local storage, while deleting the local copies of the transferred data set.

4.3 Data Dynamics Operation

4.3.1 Data Insertion:

a. The data insertion demand is created while to add new file blocks to the uploaded data.

b. The owner then delivers the required data blocks on the server along with the insertion request.

c. The server adds data and sends a proof to the owner that blocks are added successfully.

d. The owner checks the results of the data insertion to make sure the cloud server carried out the proper data addition process.

When owner inserts a block of data, the NR-MHT tree is changed, the hash value changes, and both the cloud as well as the user's side of the tree are revised. The user of the data must then retrieve it from the cloud, compute the hash values again, and decide if values are identical or not. However, it will first get the previous NR-MHT value to see whether it has changed or not.

4.3.2 Data Deletion:

This technique of deleting data can be carry out by both the data holder and the cloud server.

a. To remove the undesirable data blocks or files, the user files a data removal request and transmit it to the cloud server.

b. After that, the cloud server carries out the request to erase the data and wraps up the removal procedure.

d. The cloud server subsequently transmits any related deletion evidence to the owner or user.

e. Blocks of necessary data are correctly erased.and the data owner reviews the data removal evidence.

When owner delete a data block, the cloud's NR-MHT tree is changed, the hash value changes, and both the cloud and the user's end of the tree are updated. The user of the data must then retrieve it from the cloud, compute the hash values again, and decide if the data is identical or not. But once again, it will retrieve the original NR-MHT and ascertain whether or not its value has changed.

4.3.3 Data Sharing:

The primary benefit of this approach is data sharing between users, which no researchers have previously addressed in work on MHT-based data integrity audits. Sharing of data only takes place inside a group. This provides a high degree of security for cloud storage, thus. The design of this system introduces a key server for key sharing across several users. The important information for users, groups, and key certificates is stored on the key server. Additionally, the NR-MHT audit data is preserved.

Data Sharing algorithm works in the following way:

a. A group's user and key are created by the data owner and stores it in the key server.

b. Since each file or folder creates its own key and it is impractical to deliver each file individually to a person, sharing a single file is not a viable idea. Instead, folders of files and groups of users are formed.

c. The application will allow group of users or user to access those folders or files.

d. The key server will hold the keys for each folder which is being shared with users.

e. File blocks gets created by the system.

f. These blocks are get encrypted by CP-ABE.

g. Now, the data user verifies its key to receive that data.

h. Only then data will be shared if both keys match.

4.3.4 User Revocation:

If a user's access is revoked, an updateable ABE enables them to alter their encryption key. The owner receives the revised key token. The data must be downloaded by the owner, who must then re-encrypt it using the new NR-MHT token. In the past, all processing was carried out by the system; however, the developed system only does process at owner's ends.

The User revocation occurs in the phases described below:

a. The data owner must choose which user accounts to terminate.
b. Next, system modifies the key.
c. Retrieve the Cloud logs. Re-encrypt using a fresh key from the key server.
d. Create a fresh NR-MHT after encoding.
e. Determine the hash value.
f. Send data blocks, keys, NR-MHT data, and group details to a cloud server
g. Send the key server a new hash value.
h. Verifying if the values are the same for the client and the cloud.

4.4 Data Integrity Auditing

The main important procedure of integrity checking of data get done by the algorithm of NR-MHT.

The owner verifies the integrity and accessibility of uploaded files or data to ensure that the data piece is legitimately maintained.

In particular, the owner can willingly select a folder or file to inspect for tampering. The owner then obtains the subfile and any necessary supplementary authentication data. The owner re-calculate a new Merkle root for comparing with the one created by the data owner prior to data uploading in order to checks if the cloud server truthfully holds the uploaded data set. Remain aware that every block of outsourced data is linked to the Merkle root.

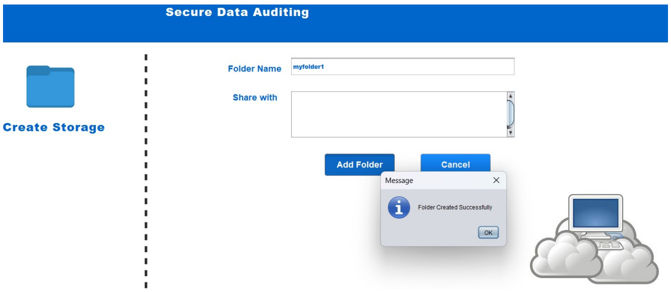
The selected hash-function is safe and one-way and exhibits properties like pre - image irrevocability, collision tolerance, and puzzle friendly. Additionally, the overall number of exported subfiles in a cloud storage system is relatively vast, and the owner selects a subfile at random. As a result, the cloud server cannot successfully create a bogus data block if the uploaded data blocks of files have been tampered, such as when certain data blocks have been deliberately removed or altered. It indicates that the cloud server does not completely preserve the set of data that was outsourced. As a result, the technique we've proposed can provide data integrity audits.

5 Implementation

By eliminating Third-Party Auditor from the scenario of data management on the cloud, which was included in the previous research of the data integrity auditing. This study's main objective is to offer secured data auditing with the additional features like data sharing within a group of users or to a single user, and user revocation which was not included in the existing system.

To develop and implement this system we used two algorithms, NR-MHT for secured data auditing, CP-ABE for encryption and features like data sharing. There two parts of this development one is desktop side application and second is two servers on the EC2 instance of AWS, which is cloud server of the system. At the cloud end MYSQL 8.1 database is used for data sroeage and Java based REST API's created for communication. For data management, data owner used desktop application to store, to it share with other users in secure way and to check the integrity of the data. For the development of desktop application, Netbeans 8.2 IDE is used with JDK11.

First owner or user has to register itself in the system then they has to create a folder in which they can store their multimedia files and and upload them on to cloud where they can share it with other users.



Folder Name: myfolder1

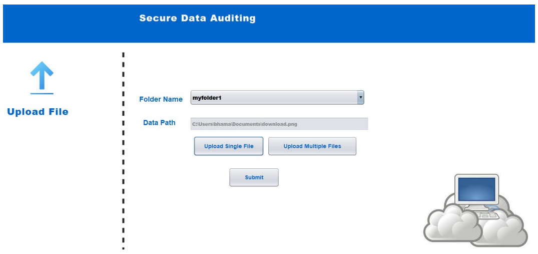
Folder Created Successfully

Figure 3: Folder Created

Here, myfolder1 is folder which is created by the data owner wih username user1. Now we will have a look what happens when folder gets created and files uploaded on the cloud by the data owner or user.

There are two types of server on cloud one is for storing the secured encrypted files and other one is our Key server which stores the encryption keys for particular folder.

After creation of folder multiple encryption keys gets created for the particular folder and files stored inside it. This encryption done using the CP-ABE algorithm. After folder creation data owner can manage the folder from the Manage Folder, where he/she can grant access on his/her folder to other users of desktop application. The owner can also revokes the access of the user from the folder. This is user revocation, now how can user revocation is achieved ? The user revocation achieved with the CP-ABE algorithm. First data owner has to choose the user account which he/she wants to revoke access from. Then system modifies the key. Then system download the logs and re-encrypt using a fresh key from the key server. The whole tree gets updated after the encoding. New hash value gets generated. After this process, system sends data blocks, keys, NR-MHT data and group details to a cloud server. Key server receives the new hash value. After verifying the values of client and server the revocation takes place in the proposed system.

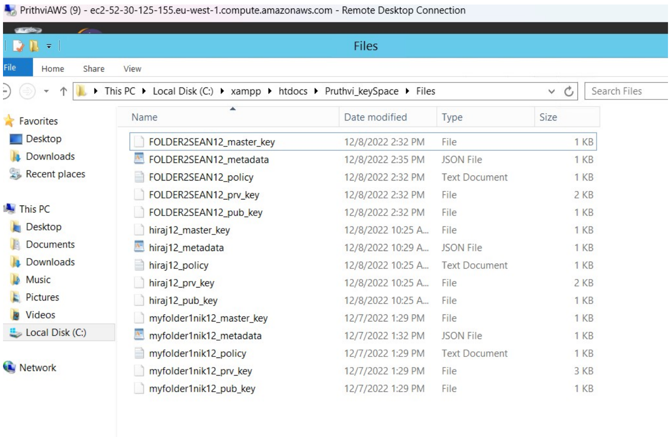


Folder Name: myfolder1

Data Path: C:\Users\bhama\Documents\download.png

Figure 4: File uploading

Now when we upload a file, the file will get divide into the blocks. Every block will get encrypt and then metadata gets created. Metadata means the hash value. Every file gets divide into the blocks and has its hash value, by adding the hash values of files stored in the particular folder signature value gets created for that folder. This metadata and encryption keys for the particular folder gets stored in the key server space. This process is done by the use of NR-MHT algorithm. User can also upload multiple files at time.

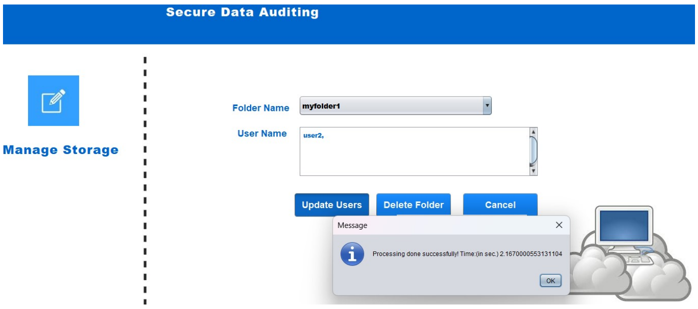

PrithviAWS (9) - ec2-52-30-125-155.eu-west-1.compute.amazonaws.com - Remote Desktop Connection
Files
File
Home
Share
View
This PC ▸ Local Disk (C:) ▸ xampp ▸ htdocs ▸ Pruthvi_keySpace ▸ Files
Search Files
Favorites
Desktop
Downloads
Recent places
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Network

Name	Date modified	Type	Size
FOLDER2SEAN12_master_key	12/8/2022 2:32 PM	File	1 KB
FOLDER2SEAN12_metadata	12/8/2022 2:35 PM	JSON File	1 KB
FOLDER2SEAN12_policy	12/8/2022 2:32 PM	Text Document	1 KB
FOLDER2SEAN12_prv_key	12/8/2022 2:32 PM	File	2 KB
FOLDER2SEAN12_pub_key	12/8/2022 2:32 PM	File	1 KB
hiraj12_master_key	12/8/2022 10:25 A...	File	1 KB
hiraj12_metadata	12/8/2022 10:29 A...	JSON File	1 KB
hiraj12_policy	12/8/2022 10:25 A...	Text Document	1 KB
hiraj12_prv_key	12/8/2022 10:25 A...	File	2 KB
hiraj12_pub_key	12/8/2022 10:25 A...	File	1 KB
myfolder1nik12_master_key	12/7/2022 1:29 PM	File	1 KB
myfolder1nik12_metadata	12/7/2022 1:32 PM	JSON File	1 KB
myfolder1nik12_policy	12/7/2022 1:29 PM	Text Document	1 KB
myfolder1nik12_prv_key	12/7/2022 1:29 PM	File	3 KB
myfolder1nik12_pub_key	12/7/2022 1:29 PM	File	1 KB

Figure 5: Keys and MetaData on cloud

In figure 5, we can see the encryption keys and metadata file in which all the information of hash values and signature value for myfolder1 is stored. This key server removes the Third Party Auditor from the scenario and this was our goal. The original file will get store in the other storage server in the encrypted format. This is the whole scenario when during folder creation and file uploading.

Now we will have look at how our feature of data sharing with other users done. With this system, data or folder of files can be shared with other users of desktop application. The owner of the folder can share folder with other users.



Folder Name: myfolder1

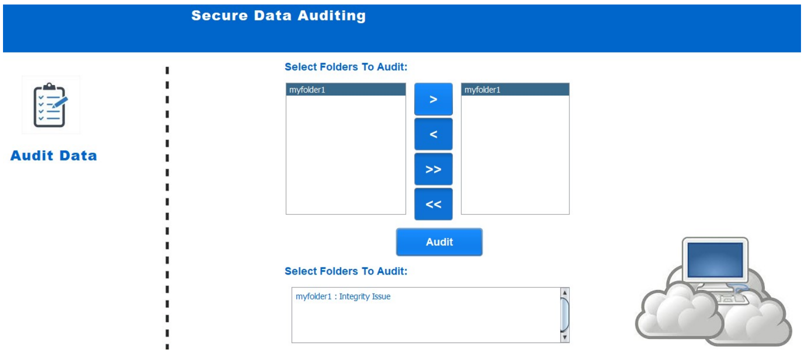
User Name: user2,

Processing done successfully! Time:(in sec.) 2.1670000553131104

Figure 6: File Sharing

In figure 6, owner user1 is sharing myfolder1 with user2. Now what happens, since each folder creates its own key and it is impractical to deliver each file individually to a person, sharing a single file is not a viable idea. Instead, folders of files and groups of users are formed. The application will allow group of users or user to access those folders or files. The key server will hold the keys for each folder which is being shared with users then blocks of file gets created by the system. These blocks are get encrypted by CP-ABE. Now, the data user download the keys to receive that data.Based on the CP-ABE policy user can download and decrypt the data. This data sharing is done with the use of CP-ABE algorithm. The shared files or folder will be in the inbox of the user's desktop application.

Now the main feature of the system is auditing of the data which is carried out by the data owner on the desktop application. This data integrity auditing is done with help of NR-MHT algorithm. Files are stored in the cloud space of other server, if someone tampers in any sense, the file at the cloud space then desktop application will show the integrity issue with the particular folder.



Secure Data Auditing

Select Folders To Audit:

myfolder1

myfolder1

Select Folders To Audit:

myfolder1 : Integrity Issue

Figure 7: Audit Data

In desktop application, owner has to select a particular folder and then has to click on the audit. Then what happens after that the owner then obtains the subfile and any necessary supplementary authentication data. The owner re-calculate a new Merkle value for comparing with the one created by the data owner prior to data uploading in order to check if the cloud server truthfully holds the uploaded data set. Remain aware that every block of outsourced data is linked to the Merkle root. If the any file from the particular folder is tampered then the data blocks of that files changes and then hash value also changes. If the metadata of the server side and desktop side doesn't match, then desktop application shows the data integrity issue. While auditing is done on the folder so if it has number of files stored in it, it means the auditing will be done for all files. This explains that the proposed system supports the bash auditing as well. In figure 7, we can see myfolder1 is checked for audit and we can see clearly that data is tampered on the cloud side because it is showing integrity issue.

To perform data dynamic operations on the system, NR-MHT algorithm is used. There are five operations of it, insertion, deletion, sharing and user revocation. As we have seen the implementation of user sharing and revocation, now we will see how data insertion and data deletion is performed.

When the owner wants to add new data to the system, owner has to sends the necessary data blocks and insertion request. The cloud inserts data and send proof to the owner. Owner can verify the uploaded files on his/her side. When user wants to delete files from the cloud server, user has to send a request to the server and after that server removes the data and owner can verify the same on his/her side.

When user inserts or deletes the block of data, the whole NR-MHT tree changes, the metadata also changes for the both sides i.e. server end and user end. Then user has to download it from the cloud. Then user has to compute hash values again and has to verify that both values are same or not. To determine if it has changed or not, the system will get the previous NR-MHT value.

Table 1: Server-Side Application Specification.

Software Specification	**Tools**
Operating System	Windows Server 2012 R2
Hosting On	AWS EC2
Development Tool	Eclipse JEE Neon
Server	Apache Tomcat 7
Database	MySQL 8.1
Hardware Specification	
RAM	32 GB
Instance Size	t2.2xlarge
CPU	Intel(R) Xenon(R) CPU E5-2686 v4 @2.30GHz 2.30GHz

The system will be implemented in java using jdk 11. For system implementation, Client-server architecture is used. Java swing components are used to build an end-user desktop application. The HTTP protocol is used by this application to connect with the cloud server. The tool ehich is used for the development of this desktop application is Netbeans 8.2 IDE with JRE 16 installed in it. This application supports windows only. The system configuration for the server side application development is mentioned in table 1.

6 Evaluation

Evaluation, provides experimental results to show that our system is efficient and takes reduced processing time with the added extra features to the existing system developed by Yang et al. (2022).

After putting proposed plan into practice and give the precise assessment we performed experiments. The laptop used for all of the tests has a Windows operating system, 16 GB of RAM, and two Intel(R) Core(TM) i7 processors clocked at 3.6 GHz. Three experiments done on the proposed system.

6.1 Analysis of time for files with different sizes:

Figure 8, represents the comparison of time taken by the system for performing operations on the files with different sizes. First we took 5 files with varying size - 2Mb, 4Mb, 6Mb, 8Mb and 10Mb. In this experiment, every file is uploaded in single repository i.e. folder which has its encryption keys and parameters. By the comparison it suggests that upload time increases when size of the file increases if user wants modify the existing file then time taken for every file is almost similar or close to each other.

Figure 8 also suggests that when user wants to share the folder with other user or wants revoke the access of other user, the time taken for these operations for each file of different size is nearly the same. It increases when file with more MB's shared with more number of users. Download and upload time is depend on the speed of internet where system is running on. During the experiments, download speed of every file is almost close to each others. The average downloading speed was 12 seconds. Because of this

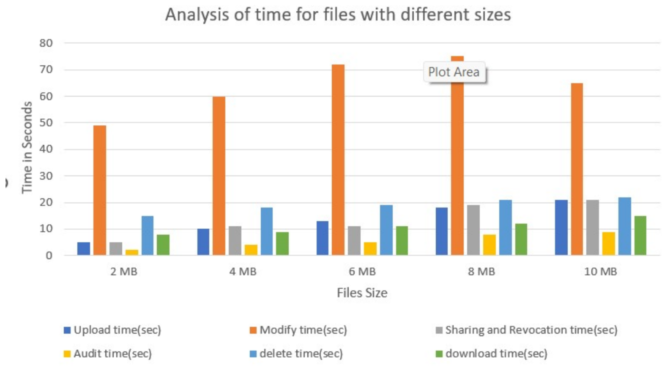


Figure 8: Comparison of different files with different size in terms of time

experiment, got to know how system reacts when the size of the files are different. Main output of auditing time is also similar for every file as it reduces the overhead of the system. This experiments tells us that system works efficiently in most of the features.

6.2 Analysis and comparison of upload time:

Figure 9, shows the analysis and comparison of upload time between existing system and developed system. X-axis shows the No of files uploaded on cloud server and y-axis shows the time taken for uploading files. For uploading files on to the cloud server previous system uses an AES algorithm which is slower than CP-ABE Previous system takes more time to upload number of files as it generates encryption keys and metadata for every file, while our proposed system uses single repository for uploading multiple files. Because of this time taken for uploading 10 files of different sizes in previous system took 122 seconds where proposed system took 108 seconds which is lower than the previous system. Same result are found when size of files increased in number of files.

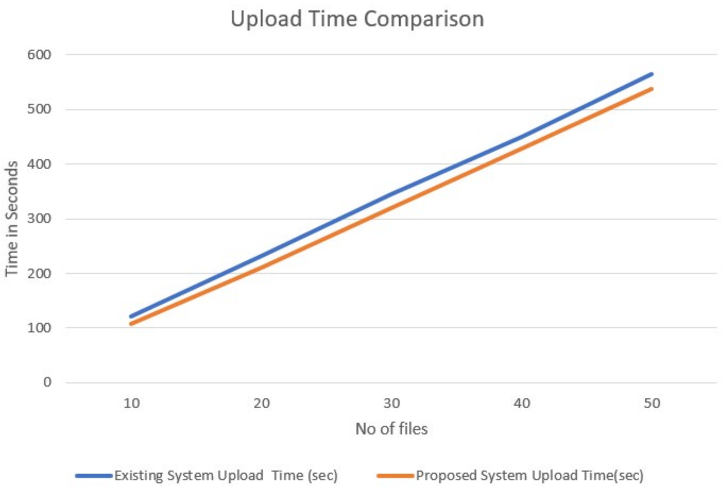


Figure 9: Comparison of existing system and proposed system in terms of Upload time

In proposed system, File encryption and signature creation for all files in a repository or folder where files are stored are done using folder-specific keys such public, private keys and keys parameters. The system gives access policies to folder rather than each files, this is done using CP-ABE encryption algorithm. As a result, each key creation procedure is less in overhead.

6.3 Analysis and Comparison for Data auditing in terms of communication cost and auditing time:

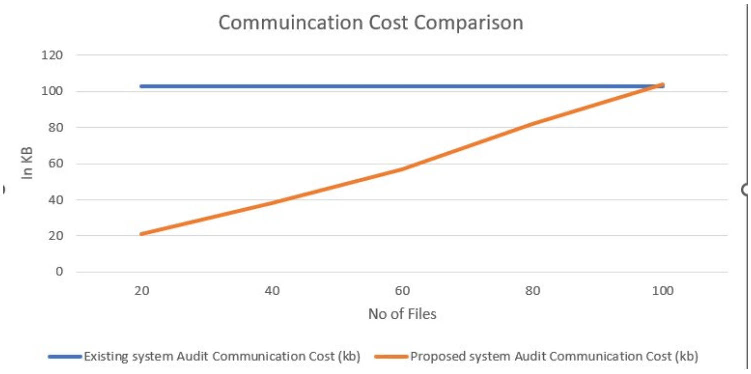


Figure 10: Comparison of existing system and proposed system in terms of communication cost

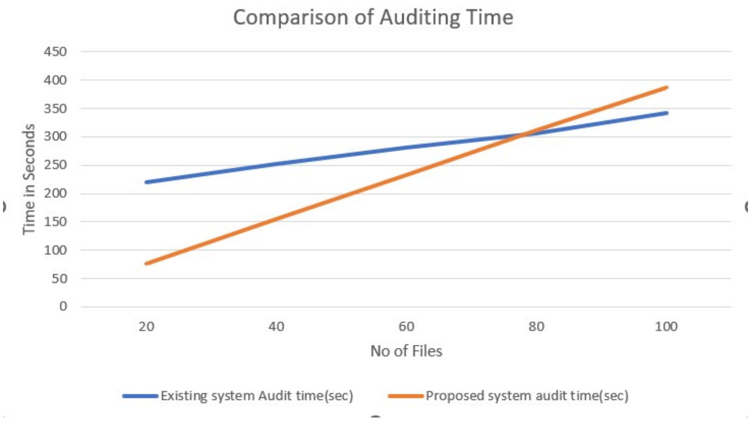


Figure 11: Comparison of existing system and proposed system in terms of Auditing Time

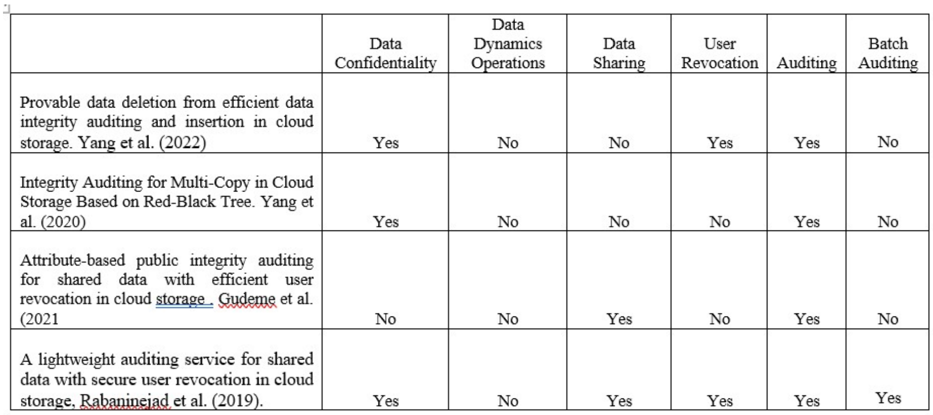
In figure 10, The communication cost is almost same for every batch of files in previous system, for 20 files it is 103 Kb, for 40 it is the same and above more . But what happens in the case of proposed system if we are uploading 20 files in the system we can put them in 5 folders which that is in multiple repositories, so that metadata will be calculated for those five folders separately. But in case of previous system metadata will be calculated for each file again and again which affects the communication cost because communication cost is associated with the transactions of the data from server to the client at every audit check.

In Figure 10, graph shows only files which are more in numbers like in experiment 100 files were taken so, we have to create more folders in proposed system only then communication cost matches to the previous system. Previous system was multiple repository system, so every time it has to download the data tree. But in proposed system it is single folder system so it doesn't need to download the whole data tree again and again. File size also affects the data blocks allocated to it. So this results into the reduced communication cost for proposed system. Communication cost also makes a difference in Auditing time.

In figure 11, the experiment carry out for the auditing time. Auditing of 20 files of different sizes took 76 seconds in proposed system but in existing system it takes 220 seconds. Likewise when 40 files of different sizes are audited, developed system took 155 seconds to complete auditing of 40 files. In experiments it is observed that when number of files increased auditing time increased. But proposed system is useful when user or owner wants to few files or particular file to be audited. It doesn't need to audit every file.

The results shows that proposed system took less auditing time. It happened because in an existing system every file has its own repository or folder so every file creates its hash value and encryption keys. Due to a single folder-specific and associated metadata in our proposed solution, a single signature value is created for all files included in a certain folder. By doing this, the overhead associated with retrieving each file's signature value from the key server during the proof creation process is reduced. The developed system works well while batch auditing a single repository and multi-repository or folder auditing. At the time of audit user can download metadata only for required repositories. The NR-MHT processing time get reduced when number of tree nodes reduced. Thus, in proposed system auditing time is reduced. In figure 11, we can see the auditing time is more than the existing time only when the number of files are more but for less number files it is very efficient than previous system. Figure 10 and 11 explains the scenario of comparison between previous system and proposed system in terms of communication cost and Auditing time.

6.4 Comparison of Properties:



	Data Confidentiality	Data Dynamics Operations	Data Sharing	User Revocation	Auditing	Batch Auditing
Provable data deletion from efficient data integrity auditing and insertion in cloud storage. Yang et al. (2022)	Yes	No	No	Yes	Yes	No
Integrity Auditing for Multi-Copy in Cloud Storage Based on Red-Black Tree. Yang et al. (2020)	Yes	No	No	No	Yes	No
Attribute-based public integrity auditing for shared data with efficient user revocation in cloud storage . Gudeme et al. (2021	No	No	Yes	No	Yes	No
A lightweight auditing service for shared data with secure user revocation in cloud storage. Rabaninejad et al. (2019).	Yes	No	Yes	Yes	Yes	Yes

Figure 12: Comparison of Properties with different research

Figure compares several methods that make use of these strategies and their characteristics. Our proposed strategy offers data confidentiality, data dynamics operations, data sharing, user revocation, auditing and bash auditing. The existing system has drawbacks of data sharing and user revocation along with bash auditing. Comparing to these developments our proposed system has less overhead as per the results.

6.5 Discussion

The proposed system has dynamic PDP approach to provide the number of features like data dynamics operations such as data insertion, data deletion, data modification with the extra features of data sharing and user revocation efficiently with less communication cost and auditing time. Data insertion, deletion and modification carried by the NR-MHT algorithm. By Figure 9 and experimental results shows system takes less time to upload than the existing system Yang et al. (2022) developed which does not have features like data sharing, bash auditing and user revocation.

The main feature is to provide secured and efficient data auditing. The number of experiments performed to get results and results are satisfying the goals of the proposed system. Figure 10 and 11 shows that communication cost and auditing time is less for proposed system. In proposed system, data is organised in repository or folder specific structure which can hold number of files in it. This reduces the overhead of the system during data uploading and data auditing, we can see it in the figure 10 and 11. But in the case of existing system which Yang et al. (2022) developed, the system was creating NR-MHT or hash value for single file at a time. So if we want to audit multiple files we have to audit them one by one. This increased the communication cost and auditing time of existing system. The existing system has AES encryption technique which is slower than CP-ABE algorithm, so upload and encryption time is less in proposed system, figure 9 indicates the same.

In proposed system, key server is introduced to maintain the confidentiality and integrity of the data by removing the third-party auditor. Now auditing will be done on the client side. That means data owner can perform data auditing without interacting with Third-Party Auditor.There is a user access policy for specific key generation for every file. It reduces the tree size and data communication cost when partial data is audited. It also reduces the overhead of data auditing.

Compare to existing system, proposed system has data dynamics operation with user sharing along with user revocation, based on the results or findings proposed system works efficiently in data uploading and data auditing in terms of time and communication cost.

7 Conclusion and Future Work

In this study, we studied that the proposed or developed system can perform secured data auditing along with data dynamics operations and user sharing. The system uses NR-MHT algorithm to provide integrity checks for data and CP-ABE algorithm used to provide encryption, data sharing and user revocation for these features it follows PDP approach. In proposed system, every uploaded file generates a hash value and NR-MHT trees gets formed. But the system is folder specific so by adding up all the hash values of files one signature values gets created for the folder where all files are stored. Because of this system takes less time for performing file uploading and data auditing. The data owners can securely store their data into the cloud server while they can check the integrity of their data on their end. This is done because of the introduction of key server in proposed system. The encryption keys for the folder are stored in the key server as it removes the role of third-party auditor. By performing data integrity checks with the help of NR-MHT data confidentiality is maintained which was one of the objective of proposed system. The experiments and result findings shows that the system takes less time to perform data integrity audits and it is effectively efficient.

One of the goal of proposed system was that owner or user can share data with the other users of the desktop application system and also owner can revoke access of any user from the shared folders securely and efficiently. This has been achieved using CP-ABE algorithm.

It is possible to introduce data de-duplication for the future work on this proposed system. Based on the experiments results, it is verified that proposed system having extra features like data sharing and user revocation has higher efficiency, effectiveness, usefulness in secure data uploading, data integrity auditing in terms of cost communication and time. The findings also demonstrate that proposed strategy is much more effective than the previously existing techniques.

References

Chen, X., Li, J., Huang, X., Li, J., Xiang, Y. and Wong, D. S. (2014). Secure outsourced attribute-based signatures, *IEEE transactions on parallel and distributed systems* **25**(12): 3285–3294.

Garg, N., Bawa, S. and Kumar, N. (2020). An efficient data integrity auditing protocol for cloud computing, *Future Generation Computer Systems* **109**: 306–316.

Gudeme, J. R., Pasupuleti, S. K. and Kandukuri, R. (2021). Attribute-based public integrity auditing for shared data with efficient user revocation in cloud storage, *Journal of Ambient Intelligence and Humanized Computing* **12**(2): 2019–2032.

Kavin, B. P. and Ganapathy, S. (2021). A new digital signature algorithm for ensuring the data integrity in cloud using elliptic curves., *Int. Arab J. Inf. Technol.* **18**(2): 180–190.

Li, J., Zhang, Y., Chen, X. and Xiang, Y. (2018). Secure attribute-based data sharing for resource-limited users in cloud computing, *Computers & Security* **72**: 1–12.

Rabaninejad, R., Ahmadian, M., Asaar, M. R. and reza Aref, M. (2019). A lightweight auditing service for shared data with secure user revocation in cloud storage, *IEEE Transactions on Services Computing* .

Tiwari, D. and Gangadharan, G. (2018). Seccloudsharing: Secure data sharing in public cloud using ciphertext-policy attribute-based proxy re-encryption with revocation, *International journal of communication systems* **31**(5): e3494.

Wang, Q., Zhou, F., Xu, J. and Xu, Z. (2021). Efficient verifiable databases with additional insertion and deletion operations in cloud computing, *Future Generation Computer Systems* **115**: 553–567.

Xiong, H., Zhang, H. and Sun, J. (2018). Attribute-based privacy-preserving data sharing for dynamic groups in cloud computing, *IEEE Systems Journal* **13**(3): 2739–2750.

Yang, C., Liu, Y., Zhao, F. and Zhang, S. (2022). Provable data deletion from efficient data integrity auditing and insertion in cloud storage, *Computer Standards & Interfaces* **82**: 103629.

Zhang, C., Xu, Y., Hu, Y., Wu, J., Ren, J. and Zhang, Y. (2021). A blockchain-based multi-cloud storage data auditing scheme to locate faults, *IEEE Transactions on Cloud Computing* .