International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6765
***Article Received**: 15 March 2023 **Revised**: 28 April 2023 **Accepted**: 18 May 2023*

A Secure Storage Management & Auditing Scheme for Cloud Storage

Subhash G. Rathod[1], Ashish K. Bhise[2], Nisar S. Shaikh[3], Yogesh B. Dongare[4], Suhas R. Kothavle[5]
[1]Department of Computer Engineering
Marathwada Mitramanda's Institute of Technology, Pune, India
subhash.rathod@mmit.edu.in
[2]Department of Artificial Intelligence & Data Science
Marathwada Mitramanda's Institute of Technology, Pune, India
ashish.bhise@mmit.edu.in
[3]Department of Artificial Intelligence & Data Science
Marathwada Mitramanda's Institute of Technology, Pune, India
nisar.shaikh2022@mmit.edu.in
[4]Department of Computer Engineering
Marathwada Mitramanda's Institute of Technology, Pune, India
yogesh.dongare@mmit.edu.in
[5]Department of Computer Engineering
Marathwada Mitramanda's Institute of Technology, Pune, India
suhas.kothavale@mmit.edu.in

Abstract— Cloud computing is an evolving domain that provides many on-demand services that are used by many businesses on daily basis. Massive growth in cloud storage results in new data centers which are hosted by a large number of servers. As number of data centers increases enormous amount of energy consumption also increases. Now cloud service providers are looking for environmental friendly alternatives to reduce energy consumption. Data storage requires huge amount of resources and management. Due to increasing amount of demand for data storage new frameworks needed to store and manage data at a low cost. Also to prevent data from unauthorized access cloud service provider must provide data access control. Data access control is an effective way to ensure data storage security within cloud. For data storage cost minimization we are using DCT compression technique to ensure data compression without compromising the quality of the data. For data access control and security asymmetric cryptographic algorithm RSA is used. For data auditing we have used MD5 with RSA to generate digital signatures, In proposed work we tried to cover all attributes in terms of efficiency, performance and security in cloud computing.

Keywords – Cloud Computing, Cloud Storage, Resource Consumption, Data Access Control, Data Security, Data Privacy.

I. INTRODUCTION

Cloud computing domain is the fastest developing business model that provides on demand resources such as data storage, business software's such as zoom, Email, Skype. Many Businesses can take advantage of the various cloud services remotely and access their personal data on any device. Cloud computing is the fastest growing technology that is gaining popularity day by day. Cloud computing allows users to access applications and data stored on remote servers, rather than on their own local computer. This means that users can access their data and applications from any device with an internet connection, without having to install the software or store the data on their own device. Cloud computing also allows for more efficient use of resources, as multiple users can access the same application or data simultaneously.

Additionally, cloud computing can provide cost savings by reducing the need for hardware and software [4][6][21].

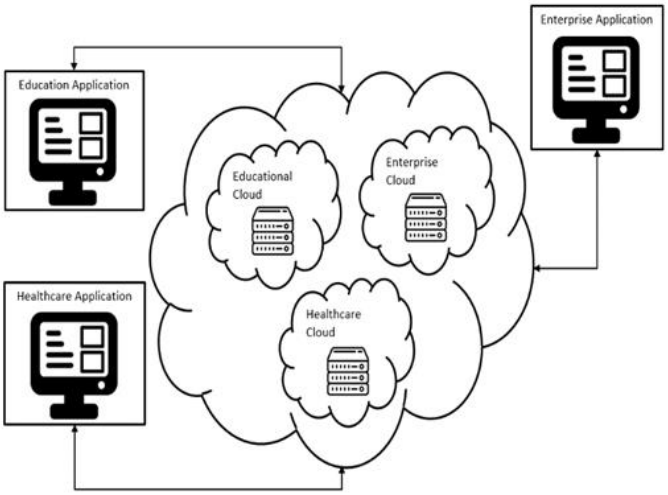


Figure 1.1 Cloud Platforms & Storage Model

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6765
***Article Received**: 15 March 2023 **Revised**: 28 April 2023 **Accepted**: 18 May 2023*

There are various types of cloud platforms that are designed as per the need and requirement in the market some of them are mentioned below.

1.1. Education Cloud Platform

The cloud-based education system has the potential to revolutionize the education sector. It offers a number of advantages to both students and educators. Students can access their course materials from any location and at any time. They can also interact with their teachers and classmates online. Educators can also use the cloud to store

1.2. Enterprise Cloud Platform

An enterprise cloud can provide a single point of control to manage infrastructure and applications in any cloud or data center. This can provide a unified operating environment, which can improve efficiency and agility. Additionally, an enterprise cloud can help consolidate resources and reduce costs. Most of the cities have free internet access and people can connect to internet without any security. So, it is very important to have a proper security system while using cloud services. Some of the security features that should be considered while using an enterprise cloud is multi-factor authentication. This is a security feature that requires more than one form of identification to log in to.

1.3. Healthcare Cloud Platform

The platform helps to aggregate and normalize data from different sources, including EHRs, claims data, and clinical data. The platform also enables healthcare organizations to develop and deploy custom applications and analytics to improve care delivery and population health. Healthcare cloud platform can help to manage health data securely. It can help to share data securely between different providers, organizations, and patients. Platform can help to improve patient experiences and health outcomes.

In traditional networking setup, the server is fixed in terms of resources so whenever company needs to scale up cost of buying new resources also increases but in in cloud computing multiple servers are already in place you just need to rent resources as you need there is no extra cost of setup. So when your company outgrows the current server, you can simply add more resources by subscribing to more cloud services. It also increases efficiency for example in a traditional networking setup, you may have to purchase and install software licenses for each user. But with the cloud, you can simply subscribe to the software you need [1].

Cloud computing has many advantages over traditional computing infrastructure such as scalability, elasticity, and pay-as-you-go pricing. But, there are also some disadvantages to cloud computing. One disadvantage of cloud computing is that it can be more expensive than traditional computing infrastructure. With traditional infrastructure, you can purchase the hardware and software you need and then use it for as long as you want. With cloud computing, you typically need to pay for a subscription or pay as you go, which can be more expensive in the long run. Another disadvantage of cloud computing is that it can be less reliable than traditional infrastructure. If the cloud provider experiences an outage, you may not be able to access your applications or data. With traditional infrastructure, you typically have a backup plan in place in case of an outage. Finally, cloud computing can be more complex than traditional infrastructure. You need to understand the cloud provider's terms and conditions and how to use the provider's tools and applications. With traditional infrastructure, you typically only need to understand how to use the applications and tools provided by the hardware and software vendors. In some cases environmental events such as earthquake or floods can cause major harm to user's data. Cloud servers are also subject to external attacks that makes them vulnerable in terms of data security and privacy.

Attributes	Traditional Computing	Cloud Computing
On Demand Resource and Services	Not Possible	Possible
Resource Management	Centralized	Decentralized
Initial Infrastructure Cost	High	Low
Security	High	Low (Depend on the type of cloud)
Resource Usage Cost	High	As per the Usage

Table 1 Traditional Computing and Cloud Computing Differentiation

The introductory chapter provides background information regarding the research. The second chapter provides a comprehensive review of the literature and existing technologies. Chapter 3 provides explanation and justification for research methodology used, Chapter 4 presents descriptive statistics and the result of the analysis, followed by conclusion.

II. RELATED WORK

2.1. Data Access Revocation using Mix & Slice approach

This system introduces data confidentiality and transmission along with data access revocation. Proposed system it contains two storage providers, data owner, and multiple data consumer. Author assume that the cloud service provider is untrusted and vulnerable as there is risk of man-the-middle attack [1].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6765
***Article Received**: 15 March 2023 **Revised**: 28 April 2023 **Accepted**: 18 May 2023*

Author presented scenario where cloud storage providers are not honest and can be malicious and vulnerable to external attacks on the other hand the data transfer channel between users is safe because the data owner send the data in fragments. The fragmentation is done by using symmetric encryption method with all-or-nothing (AON) mode.

These fragments are distributed and stored over two different cloud. And secrete key used to recover data into original form is securely passed among authorized users.

2.2. SecACS Framework

In SecACS framework Cloud service provider contains two main modules. First module is responsible for storing data securely. Second module respond to data audit requests and respond user with data integrity proof.

There are following entities in the proposed SecACS framework:

- Trusted key generation center (TKGC). It is responsible for generating secrete keys and some public parameters.
- Data user: Data users send request to cloud service providers and verify the proof got from CSP as a response.
- Data owner: Data owners are can outsource data to cloud service provider also can update data.
- Secure channel: This channel is responsible for distributing the secrete key securely.

First, trusted key generation center generates cryptographic key and using secure communication channel distribute it to data users and owner. Data owner can outsource data blocks by generating tags for individual data over cloud. After that users can send request to CSP to audit data and in response CSP provide proof that is generated by using data blocks and their tags, generated proof can be verified by data users to check data integrity [2].

2.3. ID-Based Privacy-Preserving & Integrity Verification

In this framework there are four types of entities actively participating first is Key Generation Center (KGC) which is responsible for generating keys and distributing it to all users. Second is group users which includes data owner and other users in the network, data owners can outsource data over cloud while other users on the network can access, download and modify shared data blocks these users can perform insert, update and delete operation on the block.

On behalf of users request third party auditor can verify integrity of data and data privacy against TPA is provided by using proposed ID based data auditing protocol [3].

III. PORPOSED MODELLING

We considered a data access control system in multi-authority cloud storage, as described in Fig. 3.1 There are 5 kinds of entities within the system: a security key generation and distribution module which is responsible for generating cryptographic keys and distributing them among data owners and users, data owners are responsible for outsourcing data over cloud, data users can only request for data and download data if respective owner of that data permits it, lastly the third party auditor an audit the data to check integrity of that data [19].

3.1. Owner Process

3.1.1. Compression

This module reduces storage size by half. This module compresses images using the DCT (Discrete Cosine Transform) algorithm with quality threshold ratio. We use DCT because tt is particularly useful for reducing the amount of data required to represent a signal without losing information. The DCT is a widely used tool in image and video processing that achieves high compression ratios with minimal loss of quality. It works by transforming a signal from the spatial domain to the frequency domain, where it is represented as a set of coefficients. The most significant coefficients are then kept and the rest are discarded, resulting in a much smaller representation of the original signal.

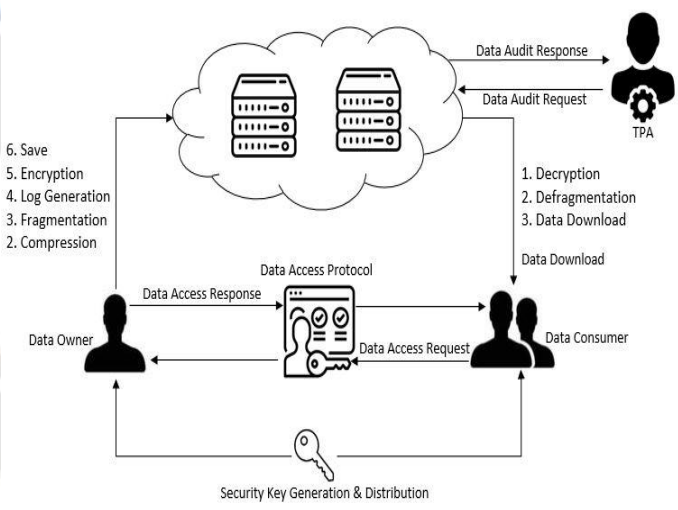


Figure 3.1 Proposed System Model

3.1.2. Fragmentation

The module works by splitting the original file into chunks and generating a log file with the order in which those chunks should be re-assembled. When the user wants to retrieve the file, they can re-assemble the chunks using the log file. This makes it difficult for an attacker to get access to the original file, as they need to have access to the log file in order to reconstitute the chunks.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6765
Article Received: 15 March 2023 Revised: 28 April 2023 Accepted: 18 May 2023

3.1.3. Log Encryption

The log file is created by fragmentation module this log file contains the information about reconstructing the image from fragments, this log file in encrypted using AES algorithm. The encryption of the log file provides an additional layer of security and ensures that the log files are not accessed by unauthorized personnel. It also ensures that the log files are not modified or tampered with in any manner [20].

3.2. User Process (Download Module)

3.2.1. Log Decryption

Before reconstructing the original image file, system need order of fragments with which fragments can be joined. Before extracting the sequence from file system needs to decrypt the log file because it has been encrypted and stored on cloud so it is unreadable to unauthorized user.

3.2.2. Data Defragmentation

Data defragmentation module reads the log file decrypted by decryption module and finds out the location of the fragments from the log. Each fragments is then used and recreate the original image file by joining the fragments together in the correct order. Finally, the original image is reconstructed.

3.3. Security keys generation phase

When new user is registered then the system generates unique cryptographic keys for registered user, these unique keys are generated by RSA algorithm by executing KeyGen function. It generates two keys first is private key for owner and public key which is shared among authorized users.

For data verification purpose, we are using RSA and MD5 algorithm. MD that is message digest is special cryptographic hash function that helps creating a 32-bit message digest. In proposed system MD5 and RSA is used for digital signature applications. RSA stands for Rivest Shamir Adleman. It is a public-key cryptography algorithm and is widely used for secure data transmission. It is also used in various digital signature schemes. In our framework hash is created for each uploaded data over cloud. Different security attacks like brute force have been investigated by RSA and Hybrid Algorithm.

3.4. TPA process

The auditing process is a secure and efficient way of ensuring the data integrity of files stored on cloud servers. It is also one of the most effective ways to detect any unauthorized changes or modifications to the data stored on cloud servers.

In proposed system auditing process has three phases first is key generation which generate two secret keys that is public key and private key, then signature generation where data owner generate signature using data and owners private key third phase is verification that is TPA generate signature by using owners public key and data. Signature generated by owner and signature generated by TPA gets compared if the signature matches then data is not violated or edited by unauthorized users if signature does not match then data is violated.

3.5. Algorithms

3.5.1 KeyPair Generation

function generateAsymmetricKeyPair(int keybitsize) → (Byte prkey, Byte pukey):
Step 1: Create a new AsymmetricKeyPairGenerator instance with RSA algorithm
keypair_generator = new AsymmetricKeyPairGenerator(RSA)
Step 2: Create a new SecureRandomNumGen instance with SHA1PRNG algorithm and SUN provider
random_generator = new SecureRandomNumGen (SHA1PRNG, SUN)
Step 3: Set the seed of the random generator to the current time in milliseconds
current_time = System.currentTimeMillisecond()
random_generator.setSeed(current_time)
Step 4: Initialize the key pair generator with the given key bit size and the random generator
keypair_generator.initialize(keybitsize, random_generator)
Step 5: Generate an asymmetric key pair and return the private and public keys as bytes
key_pair = keypair_generator.generateAsymmetricKeyPair()
prkey = key_pair.privateKey.toBytes()
pukey = key_pair.publicKey.toBytes()
return (prkey, pukey)

3.5.2 Image Upload (Owner Process)

function uploadImage(byte[] image, int number_of_fragments, String OwnerEmailID) → (byte[][] subchunk, byte[] logfile):
Step 1: Let byte image
This step is unnecessary as the input image is already passed to the function.
Step 2: Create a new image encoder with JPEG format
i_encoder = GetEncoder(ImageFormat.Jpeg)
Step 3: Compress the input image with the specified quality factor
QF = 80 // Example quality factor value
image_compressed = i_encoder.Compress(image, QF)
Step 4: Get the size of the compressed image
image_size = image_compressed.length
Step 5: Divide the compressed image into subchunks of equal size
subchunk_size = image_size / number_of_fragments
subchunk = SplitByteArray(image_compressed, subchunk_size)
Step 6: Get the private key of the owner using their email ID
prkey = GetKey(OwnerEmailID)
Step 7: Encrypt each subchunk using RSA with the owner's private key
logfile = new byte[][]
foreach fs in subchunk:

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6765
Article Received: 15 March 2023 Revised: 28 April 2023 Accepted: 18 May 2023

Step 8: Create a new cipher instance with RSA algorithm
cipher = Cipher.getInstance(RSA)
Step 9: Initialize the cipher in encryption mode with the owner's private key
cipher.init(Cipher.ENCRYPT_MODE, prkey)
Step 10: Encrypt the subchunk using the cipher and save it
cipherBytes = cipher.doFinal(fs)
SaveToFile(cipherBytes.getName(), cipherBytes)
Step 11: Write the filename of the encrypted subchunk to the log file
BufferWrite(cipherBytes.getName(), "logfile.txt")
Step 12: End foreach
Step 13: Read the log file and encrypt its content using RSA with the owner's private key
plaintext = ReadFile("logfile.txt")
cipher = Cipher.getInstance(RSA)
cipher.init(Cipher.ENCRYPT_MODE, prkey)
ciphertext = cipher.doFinal(plaintext)
Step 14: Save the subchunks in the log file
SaveToFile("subchunks.log", subchunk)
return (subchunk, logfile)

3.5.3 Image Download (User Process)

function downloadImage(int image_id, String OwnerEmailID) → byte[]:
Step 1: Get the path of the log file containing the encrypted subchunks
path = getLogFilePath(image_id)
Step 2: Read the ciphertext from the log file
ciphertext = ReadFile(path)
Step 3: Get the public key of the owner using their email ID
pukey = GetKey(OwnerEmailID)
Step 4: Create a new cipher instance with RSA algorithm
cipher = Cipher.getInstance(RSA)
Step 5: Initialize the cipher in decryption mode with the owner's public key
cipher.init(Cipher.DECRYPT_MODE, pukey)
Step 6: Decrypt the ciphertext using the cipher
plaintext = cipher.doFinal(ciphertext)
Step 7: Concatenate the subchunks into the full image
key = GetKey(UserEmailID) // Example: Get the private key of the user using their email ID
image = new byte[]{}
foreach subchunk in plaintext
Step 8: Create a new cipher instance with RSA algorithm
cipher = Cipher.getInstance(RSA)
Step 9: Initialize the cipher in encryption mode with the user's private key
cipher.init(Cipher.ENCRYPT_MODE, key)
Step 10: Read the encrypted subchunk from file
temp = ReadFile(subchunk[index])
Step 11: Decrypt the subchunk using the cipher and append it to the full image
image += cipher.doFinal(temp)
Step 12: End foreach
Step 13: Return the full image
return image
streaming the image data to the client's web browser using the response object.
Step 14: Get the output stream of the HTTP response
OutputStream output = response.getOutputStream()
Step 15: Write the image data to the output stream
output.write(image)
Step 16: Close the output stream
output.close()

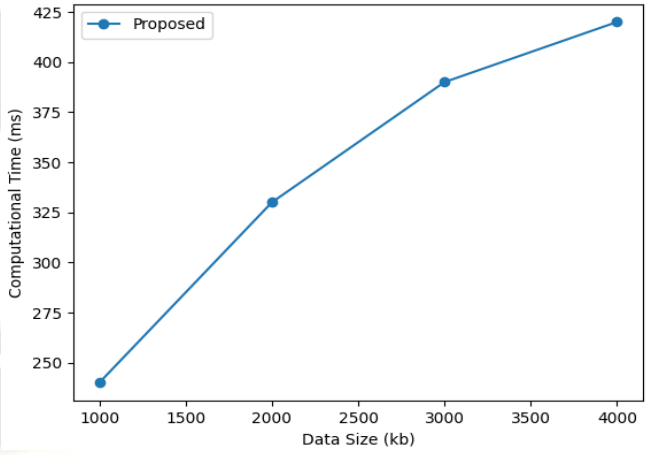
IV. RESULTS AND DISCUSSIONS

4.1 Dataset and Experimental Setup

For image compression implementation we can use any dataset like ImageNet which is a subset of Kaggle dataset, we do not need label dataset as we are not doing any object detection here, we are only reducing current image physical size by more than 50% without visually reducing image quality.

4.1.1 Efficiency Analysis

We first measure the efficiency of the proposed architecture in terms of time require to complete the task of data compression, fragmentation because both tasks are heavy and consume time as per the size of the image. As shown in table 4.1 original image size is more than 6 MB bit if we compress it with quality factor of 0.1 image size will get reduced to 196 KB but the quality of image gets compromised. While compressing we found the middle ground in terms of image quality of compressed file and size of the compressed file.



While compressing we found the middle ground in terms of image quality of compressed file and size of the compressed file.

We are compressing image with quality factor of 0.2f which not only maintains image quality but also reduces storage size by more than 50%.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6765
***Article Received**: 15 March 2023 **Revised**: 28 April 2023 **Accepted**: 18 May 2023*

Original Size	Compressed Size	Quality Factor	Time to compress
6.29 MB	197 KB	0.1f	865 ms
6.29 MB	308 KB	0.2f	783 ms
6.29 MB	429 KB	0.3f	789 ms
6.29 MB	548 KB	0.4f	787 ms

Table 4.1 Image size and Compressed Size with Quality Factor

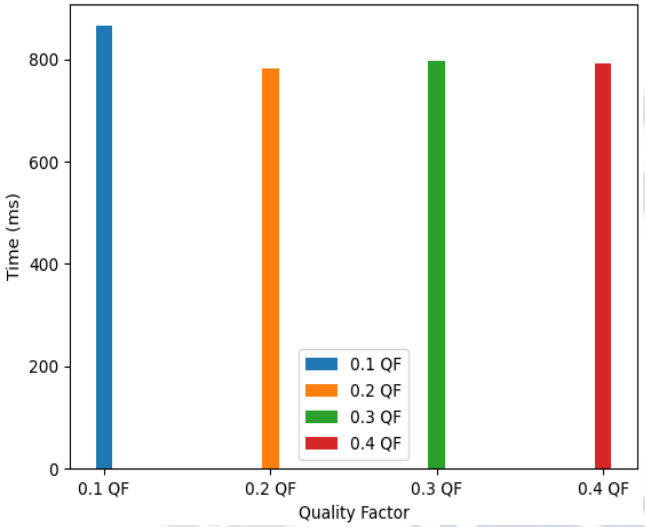


Fig. 4.2 Computation time required for variable data size

4.1.2 Audit Computational Analysis

Figure 4.2 shows the computational time required to audit the data with variable data sizes. As depicted as the data size increases the computational time for auditing data also increases.

Figure 4.3 shows the comparative analysis between SecACS auditing scheme [3] and proposed scheme, as depicted above proposed scheme takes less computational time as compared to SecACS. Also as the number of audit queries increases the computational time for auditing also increases.

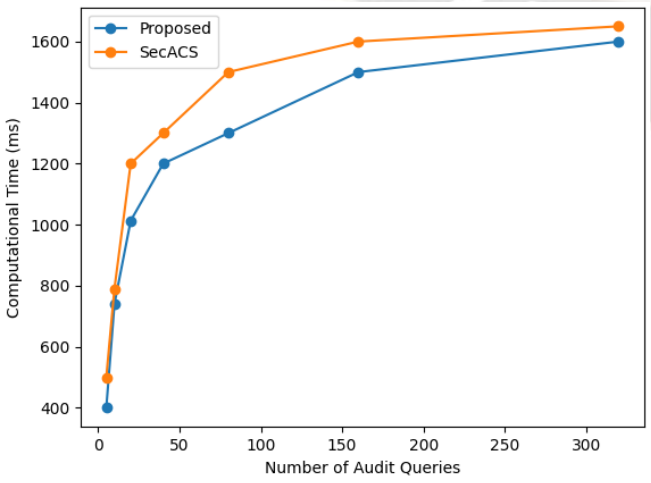


Fig. 4.3 Computation time required for number of audit queries

4.1.3 Storage Cost

For storage cost analysis we have used images of various physical sizes as a digital media, and various users upload specific amount of data. As shown in fig. 4.4 proposed framework can save up to 60% of storage space by compressing data. Here we have taken high resolution images with storage size ranging from 1 MB to 5 MB. Fig 5 shows the difference between storage sizes of original image and storage size of compressed image.

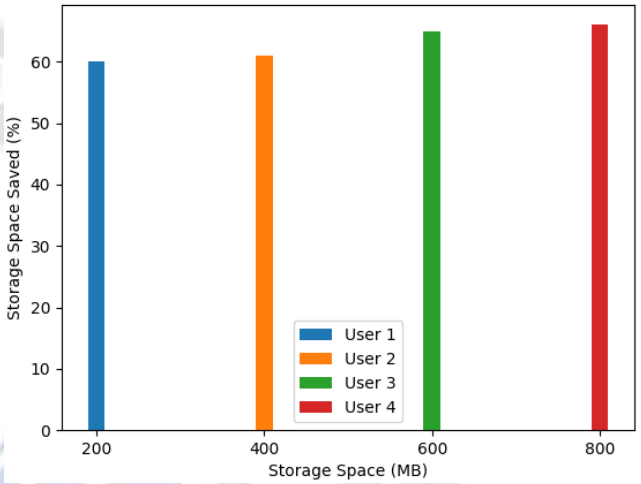


Fig 4.4 Storage space saved

V. CONCLUSION

The proposed architecture is implemented by keeping some parameters in mind like resource consumption, data security, privacy and integrity, we have implemented data optimization module using DCT compression, Data security module using data fragmentation and RSA. Data integrity module using MD5 with RSA. With this system we have achieved cost efficient storage with security and privacy to owner's data. Here Data owners can provide access restriction over their data.

Result analysis shows that we able to achieve effectiveness and robustness while compressing, we managed to reduce image storage size by more than 50% and not compromised image quality. Data correctness is analyzed by evaluating reconstructed image in terms of PSNR value. PSNR value of reconstructed image is more than 20dB which indicates that image quality of reconstructed image is maintained. Adding another security layer and providing data integrity on top of proposed architecture is our important future work to be pursued.

REFERENCES

[1] Aritra Dutta, Rajesh Bose, Swamendu Kuma Chakraborty, Sandip Roy, Haraprasad Mondal, Computational science Brainware University, Kolkata India "Data Security Mechanism for Green Cloud", IEEE 2021

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6765
***Article Received**: 15 March 2023 **Revised**: 28 April 2023 **Accepted**: 18 May 2023*

[2] Ding ManJiang 1, Cao Kai 1, Wang ZengXi 2, Zhu LiPeng 3, 1. State Grid Jiangsu Tendering Co., Ltd, Nanjing, China 2. Jiangsu Electric Power Information Technology Co., Ltd, Nanjing, China 3. Global Energy Interconnection Research Institute Co., Ltd, Beijing, China, "Design of a Cloud Storage Security encryption Algorithm for Power Bidding System", IEEE 2020

[3] Katarzyna KAPUSTA, Han QIU, and Gerard MEMMI LTCI, Telecom ParisTech, Paris, France "Secure Data Sharing with Fast Access Revocation through Untrusted Clouds" 978-1-7281-1542-9/19/$31.00 ©2019 IEEE.

[4] Li Li, Jiayong Liub "SecACS: Enabling lightweight secure auditable cloud storage with data dynamics" 2214-2126/© 2020 Elsevier Ltd. All rights reserved.

[5] Reyhaneh Rabaninejad, Seyyed Mahdi Sedaghat, Mohamoud Ahmadian Attari, Mohammad Reza Aref "An ID-Based Privacy-Preserving Integrity Verification of Shared Data Over Untrusted Cloud" K. N. Toosi University of Technology Department of Electrical Engineering Tehran, Iran, 978-1-7281-5937-9/20/$31.00 ©2020 IEEE

[6] Premlata Singh, Sushil Kr. Saroj "A Secure Data Dynamics and Public Auditing Scheme for Cloud Storage" Department of Computer Science & Engineering, Madan Mohan Malaviya University of Technology Gorakhpur, India 978-1-7281-5197-7/20/$31.00 ©2020 IEEE

[7] Jian Wang, Kehua Wu, Chunxiao Ye, Xiaofeng Xia, Fei Ouyang ∗Colleage of Computer Science, Chongqing University, Chongqing, China "Improving Security Data Access Control for Multi-Authority Cloud Storage" 978-1-7281-4328-6/19/$31.00 ©2019 IEEE

[8] YANG Zhen, WANG Wenyu, HUANG Yongfeng, and LI Xing, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China "Privacy-Preserving Public Auditing Scheme for Data Confidentiality and Accountability in Cloud Storage" 2019 Chinese Institute of Electronics. DOI:10.1049/cje.2018.02.017 ©2019 IEEE

[9] Fei Chen, Fengming Meng, Tao Xiang, Hua Dai, Jianqiang Li, Jing Qin "Towards Usable Cloud Storage Auditing" 1045-9219 (c) 2020 IEEE

[10] SI HAN, KE HAN, AND SHOUYI ZHANG Department of Science and Technology, China University of Political Science and Law, 102249 China "A Data Sharing Protocol to Minimize Security and Privacy Risks of Cloud Storage in Big Data Era" 2169-3536 2019 IEEE.

[11] Leyou Zhang, Yilei Cui , and Yi Mu , Senior Member, IEEE "Improving Security and Privacy Attribute Based Data Sharing in Cloud Computing" 1937-9234 © 2019 IEEE

[12] T. A. Mohanaprakash, Dr.J.Andrews Department of CSE, Sathyabama Institute of Science and Technology, Chennai 600119, Tamilnadu, India "Novel privacy preserving system for Cloud Data security using Signature Hashing Algorithm" 978-1-7281-1576-4/19/$31.00 ©2019 IEEE

[13] YE TAO, PENG XU, and HAI JIN, National Engineering Research Center for Big Data Technology and System, Services Computing Technology and System Lab "Secure Data Sharing and Search for Cloud-Edge-Collaborative Storage" 10.1109/ACCESS.2019.2962600, IEEE Access

[14] Zhuoran Ma, Jianfeng Ma, Yinbin Miao, Ximeng Liu, Tengfei Yang, School of Cyber Engineering, Xidian University, Xi'an 710071, China "Privacy-Preserving Data Sharing Framework for High-Accurate Outsourced Computation" 978-1-5386-8088-9/19/$31.00 ©2019 IEEE

[15] Wenxiu Ding, Member, IEEE, Rui Hu, Zheng Yan, Senior Member, IEEE, Xinren Qian, Robert H. Deng, Fellow, IEEE, Laurence T. Yang, Senior Member, IEEE, and Mianxiong Dong, Member, IEEE "An Extended Framework of Privacy-Preserving Computation with Flexible Access Control" 1932-4537 (c) 2019 IEEE

[16] HAN YU, XIUQING LU, AND ZHENKUAN PAN, College of Computer Science and Technology, Qingdao University, Qingdao 266071, China, "An Authorized Public Auditing Scheme for Dynamic Big Data Storage in Cloud Computing" r 10.1109/ACCESS. 2020 IEEE

[17] Nikolaos Doukas, Oleksandr P. Markovskyi, Nikolaos G. Bardis Department of Mathematics and Engineering Science, Hellenic Military Academy, Vari – 16673, Greece "Hash function design for cloud storage data auditing" 0304-3975/© 2019 Elsevier

[18] Nureni Ayofe Azeez, Charles Van der Vyver School of Computer Science and Information Systems, Faculty of Natural and Agricultural Sciences, Vaal Triangle Campus, North-West University, South Africa. "Security and privacy issues in e-health cloud-based system: A comprehensive content analysis" 1110-8665/2018 Production and hosting by Elsevier

[19] Rathod, S., Khobragade, R. N., Thakare, V. M., Walse, K. H., & Pawar, S. (2022, September). Lightweight Auditable Secure Cloud Storage With Privacy Enabled Data Storage Optimization. In 2022 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS) (pp. 1-6). IEEE.

[20] Rathod, S., Khobragade, R. N., Thakare, V. M., Walse, K. H., & Pawar, S. (2022, September). Model for Efficient Data Storage on Public Cloud. In 2022 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS) (pp. 1-5). IEEE.

[21] Subhash Gulabrao Rathod, Dr.K.H.Walse, Dr. R N khobragade, Dr. Vilas Thakare , & Sushama L. Pawar. (2022). PRESERVING PRIVACY & MAINTAINING SECURITY FOR SHARED DATA OVER PUBLIC CLOUD: A SURVEY. International Journal Of Advance Research And Innovative Ideas In Education, 8(3), 4971-4976.

[22] Jianghong Wei , Wenfen Liu, and Xuexian Hu "Secure and Efficient Attribute-Based Access Control for Multiauthority Cloud Storage" IEEE SYSTEMS JOURNAL, VOL. 12, NO. 2, JUNE 2018

[23] Zhan Qin, Jian Weng, Yong Cui, Kui Ren, "Privacy-preserving Image Processing in the Cloud" 10.1109/MCC.2018. IEEE

[24] Kaiping Xue, Senior Member, IEEE, Weikeng Chen, Wei Li, Jianan Hong, Peilin Hong "Combining Data Owner-side and Cloud-side Access Control for Encrypted Cloud Storage" 1556-6013 (c) 2018 IEEE

[25] Jianting Ning, Zhenfu Cao, Senior Member, IEEE, Xiaolei Dong, Kaitai Liang, Member, IEEE, Lifei Wei, and Kim-Kwang Raymond Choo, Senior Member, IEEE "CryptCloud+:

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 6
DOI: https://doi.org/10.17762/ijritcc.v11i6.6765
***Article Received**: 15 March 2023 **Revised**: 28 April 2023 **Accepted**: 18 May 2023*

Secure and Expressive Data Access Control for Cloud Storage" 1939-1374 (c) 2017 IEEE

[26] Cong Wang, Qian Wang, and Kui Ren, Wenjing Lou Department of ECE Illinois Institute of Technology , Department of ECE Worcester Polytechnic Institute "Ensuring Data Storage Security in Cloud Computing" 978-1-4244-3876-1/09/$25.00 ©2009 IEEE

[27] Cong Wang, Student Member, IEEE, Qian Wang, Student Member, IEEE, Kui Ren, Senior Member, IEEE, Ning Cao, and Wenjing Lou, Senior Member, IEEE, "Toward Secure and Dependable Storage Services in Cloud Computing" 1939-1374/12/$31.00 2012 IEEE

[28] Syam Kumar P, Subramanian R Department of Computer Science, School of Engineering & Technology Pondicherry University, Puducherry-605014, India,"An Efficient and Secure Protocol for Ensuring Data Storage Security in Cloud Computing" IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 1, November 2011

[29] CONG WANG1 (Member, IEEE), BINGSHENG ZHANG2 (Member, IEEE), KUI REN2 (Senior Member, IEEE), AND JANET M. ROVEDA3 (Senior Member, IEEE) Department of Computer Science, City University of Hong Kong, Hong Kong "Privacy-Assured Outsourcing of Image Reconstruction Service in Cloud" IEEE TRANSACTIONS ON CLOUD COMPUTING VOL:1 NO:1 YEAR 2013

[30] Subhash G. Rathod, R N khobragade, Vilas Thakare, Sushama L. Pawar. (2022). Security for Shared Data Over Public Cloud for Maintaining Privacy. Mathematical Statistician and Engineering Applications, 71(4), 7167–7173. Retrieved from https://www.philstat.org/index.php/MSEA/article/view/1336