Multilayered Approach for Data Security in Cloud Storage

S. Shanawaz Basha[1], N. Musrat Sultana[2]
Author[1], Author[2]
[1]*Programmer, Chaitanya Bharati Institute of Technology, Hyderabad*
[2]*Assistant Professor, Mahatma Gandhi Institute of Technology, Hyderabad*
[1]*syedshanawaz.basha@gmail.com*
[2]*musratsultana_cse@mgit.ac.in*

Abstract

In today's world, a vast amount of data is stored on the cloud, necessitating robust protection against unauthorized access. To ensure data privacy and security, various algorithms are employed. The primary goal of any system is to achieve confidentiality, integrity, and availability (CIA). However, existing centralized cloud storage systems often fall short of providing these CIA properties. To address this issue, decentralized cloud storage, combined with blockchain technology, is utilized to enhance data security and storage methods. This approach effectively safeguards data from tampering or unauthorized deletion. In blockchain, data is organized into a chain of blocks, each containing its own hash value stored in the subsequent block, which significantly reduces the risk of data alteration. The SHA-512 hashing algorithm is employed for this purpose, as hashing is essential for ensuring data security in various applications such as message digests, password verification, digital certificates, and blockchain. The combination of these methods and algorithms enhances data security and reliability. Additionally, the Advanced Encryption Standard (AES) is used for encrypting and decrypting data due to its robust features, further enhancing data security.

Keywords: *CIA, Decentralized Cloud Storage, SHA-512, Advance Encryption Standard, Encrypt, Decrypt.*

1. Introduction

In today's era of huge information, the exchange and sharing of data have increased dramatically. The potential risk of unauthorized individuals gaining access to confidential information remains an ongoing worry for experts in data communication. The rapid evolution of network structures has facilitated the seamless transmission of data, multimedia content over the Internet. This encompasses sensitive data like military maps, one time passwords and corporate credentials being conveyed through online channels. It is imperative to address security concerns, particularly when utilizing classified text as there exists a vulnerability whereby hackers might exploit weak points within public networks to illicitly acquire information. To effectively tackle the security challenges tied to hidden data, the need arises for the formulation of robust algorithms that can safeguard data when transmitted over the internet.

An effective approach involves adopting Advanced Encryption Standard, a technique that ensures the security of content while traversing the Internet. This method guarantees the confidentiality and integrity of such data throughout its online transmission.

The AES Encryption algorithm, also referred to as the Rijndael algorithm [1], operates as a symmetric block cipher algorithm featuring a block or chunk size of 128 bits. Unlike the previous standard DES, AES does not incorporate a Feistel network in its structure. AES transforms the individual 128-bit blocks using encryption keys of 128, 192, or 256 bits. After encrypting these blocks separately, they are combined to create the ciphertext. Its popularity and security features have contributed to its widespread use as a trusted encryption method [2].

Upon storing data on a cloud service platform, data owners forfeit control over it. While this technology provides numerous advantages, it concurrently introduces fresh security challenges, particularly those associated with data integrity. Data integrity represents one of the most pivotal elements in any system. To assure the integrity of data that has been outsourced, data owners should activate auditing mechanisms. Auditing, a process involving analysis and verification, conducted by either an internal or external auditor, serves the purpose of identifying security vulnerabilities within a system. In our research paper, we employ the auditing process to assess the data integrity of the outsourced data.

These processes involve mathematical calculations that can also be practically demonstrated, ensuring that data retains its confidentiality, integrity, and availability (CIA properties). During encryption and decryption, data is divided into chunks or blocks. There are various algorithms available for these processes, categorized into two main types:

Upon conducting an in-depth study on the AES algorithm, researchers have identified several vulnerabilities that need attention to enhance its security. These findings are as follows:

i) Increasing the number of rounds can improve the security of the data block, as observed by many researchers.

ii) The AES algorithm is designed to operate with 32-bit blocks, which limits its compatibility with 64-bit systems.

iii) The process of deciphering (decryption) is slower than the process of enciphering (encryption).

iv) The addition of the final round, excluding the mix-column step, does not significantly enhance security. Authors of AES have stated that its insertion or deletion has no effect on overall security.

v) The encryption and decryption processes for large amounts of data (thousands of bits) require different amounts of time due to the numerous rounds involved.

2. Related Work

In traditional cloud storage where all data is store at only one place called centralized cloud storage, the chances of data security is less as compared to decentralized cloud storage due to owner of centralized cloud could monitor the data and it can be altered or theft. With quick requirement to access data each individual wants to retrieve their data instantly and more securely. For this reason decentralized cloud storage is developed. In decentralized cloud storage data is stored in more than one place, this itself reduce the chances of data reaching to the data stealers[2].Because data stealers unaware of where the rest of the data is stored, due to this reason decentralized cloud storage is popular and used widely. The main objective of DCS is to provide independent data nodes with security and availability of data.

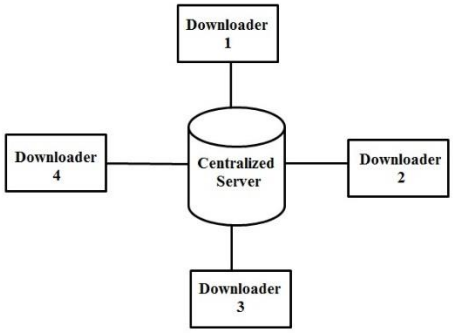

Figure 1. Traditional Centralized Cloud Storage

As illustrated in Figure 1, all data users who wish to access their data rely on a single server. This dependency can create a bottleneck, resulting in longer access times due to the high load on the single server. There is also a limit to the number of users who can connect to the server simultaneously. Applications that require many users to connect and access the server at the same time are not feasible with this setup. Additionally, if the server experiences hardware failure or setup issues, the data stored on it becomes inaccessible. In such situations, data recovery can be extremely difficult. The maintenance costs for this server are also high. Furthermore, this data storage technique charges the highest possible prices for their services. To overcome these challenges associated with centralized cloud storage, decentralized cloud storage is used. This approach eliminates the difficulties faced with centralized structures [2].

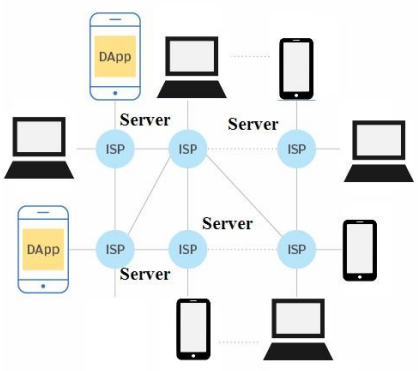

Figure 2. Decentralized Cloud Storage System

Figure 2 illustrates the structure of decentralized cloud storage. In this setup, user data is stored across multiple databases assigned by the server, creating replicas of the data. The data is broken into smaller chunks and distributed across different servers. This method offers significant storage capacity and reduces the cost of storing data on cloud servers. Unlike centralized storage, no single entity owns or controls the data, enhancing security and accessibility. Decentralized cloud storage operates on a continuously live network, allowing data access at any time.

This distributed method of storing data is particularly effective in thwarting hackers, as they cannot retrieve complete data sets. However, while data spread across multiple locations is inherently more secure against data loss, encryption is still necessary to protect data from unauthorized access. Encrypting data ensures that even if a small block of data containing sensitive information is intercepted, it remains secure.

Various algorithms are used to enhance data security in decentralized cloud storage. Notable service providers include Siacoin, Filecoin, Storj, and Maidsafe. Since decentralized cloud storage leverages blockchain technology, it offers superior data security compared to centralized storage [6]. Even if hackers access a piece of data, the encryption mechanisms used by these service providers prevent them from decrypting the information within the data block. Therefore, decentralized cloud storage is preferred over centralized cloud storage for enhanced data security.

To implement such a secure system, the Advanced Encryption Standard (AES) is used to enhance data security and keep it out of reach from attackers.

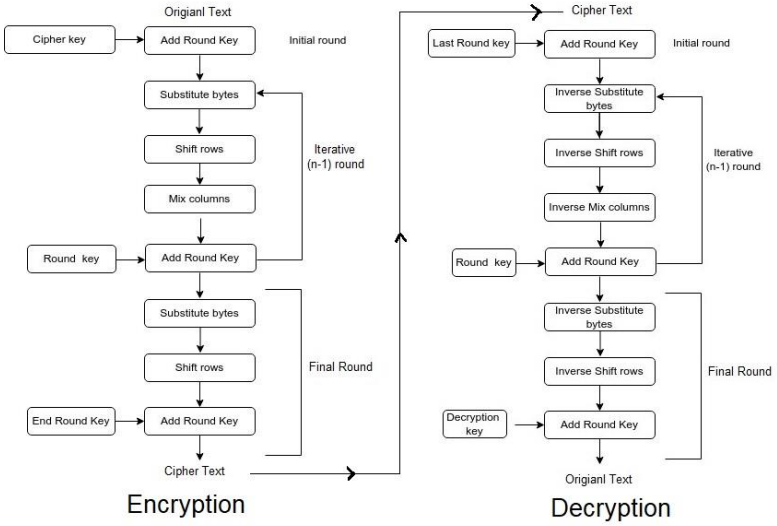

Figure 3. Encryption and Decryption of AES Algorithm

2.1. Data Block

In the initial stage, data is divided into separate blocks. As shown in Figure 3, data is divided into columns of four by four, resulting in sixteen bytes.

2.2. Key Expansion

This process involves taking the initial key words and creating an array of 44 words, which are then used as a series of keys for each subsequent round of the encryption process.

2.3. Add Round Key

Key expansion generates 10 keys using a method called key scheduling. The expanded key is XORed with the resulting data to prepare the input for the next round.

2.4. Substitute Bytes

In this stage, whole words are encoded such that each letter is replaced by the next one in the alphabet. For example, "hello" changes to "ifmmp."

2.5. Shift Rows

As the name suggests, each row is shifted. The second row is shifted to the position of the first row, the third row to the second, and so on.

2.6. Mix Columns

Each column is assigned a value based on the previous stages of the algorithm. This mixing of columns is performed accordingly.

2.7. Add Round Key (iteration)

This block takes input from the previous block and adds the round key derived at the beginning of the encryption process.

After completing these steps, the encrypted data is obtained. To retrieve the original data, the reverse operation of encryption is performed on the encrypted data, resulting in the original data.

When uploading encrypted data to the cloud server, blockchain technology is used to track the sequence of uploaded data. Blockchain technology records information in such a way that it is impossible to alter the system's data transactions. It operates using a hash function, with each data block linked to the next by a hash value. Since blockchain does not involve encryption and decryption, once an operation is performed on data, it cannot be reversed [4]. This makes blockchain technology highly trustworthy, append-only, and efficient for applications where maintaining logs is crucial. Applications include banking, online music, and the Internet of Things (IoT). This approach eliminates the need for third-party validation in peer-to-peer networks. Blockchain technology was invented by Satoshi Nakamoto and has been in use since 2008.

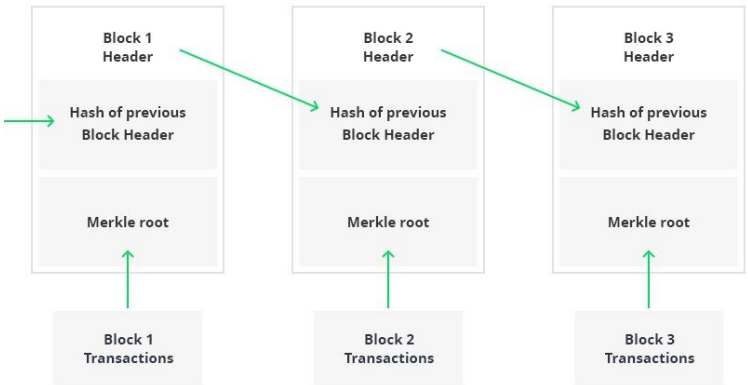

Figure 4. Working and the Structure of Blockchain

As shown in Figure 4, data blocks are linked chronologically. The header block contains information about the subsequent block to be appended. After linking to the next data block, the header of the first block is referred to as the hash block. This sequence of blocks is maintained in the Merkle root block, which securely stores all transaction-related information. This architecture stores data in a digital block format, allowing for detailed control over processing and transactions without a central point. Such a structure helps prevent cheating, data theft, and other cyber-crime attacks [3].

To implement a chain of blocks, a hashing method is required to maintain the data in sequence by their update order. This means the first transaction remains first, the second transaction remains second, and so on. This process ensures a transaction chain where alteration is impossible. Additionally, blockchain technology prevents data repetition, reducing unnecessary or duplicate data blocks. For this purpose, we use the SHA-512 hashing algorithm.

The SHA-512 algorithm, developed by Ron Rivest, is a one-way hashing algorithm. It is an evolution of previous algorithms such as SHA-0, SHA-1, SHA-256, and SHA-384 [4]. Hashing, also known as a compression or message summary function, takes a variable-length input and converts it into a fixed-length binary sequence. This hash function is designed to be irreversible, making it a one-way process. The concept of the hashing algorithm is illustrated in Figure 5.

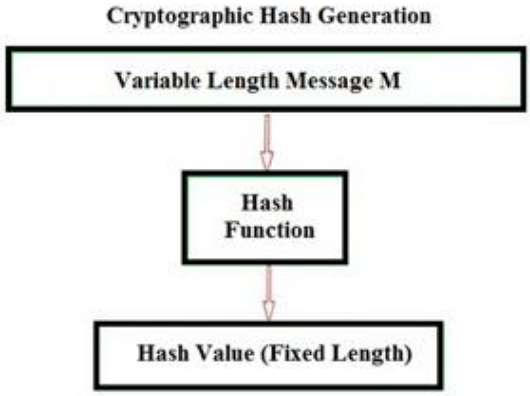

Figure 5. Working of Hash algorithm

SHA – 512 Algorithm

The first step in all the SHA-512 algorithm processes involves adding a bit according to specific algorithm rules. This initial step is crucial across various hashing algorithms. The SHA-512 algorithm then processes the message in 1024-bit blocks. Following this, an additional 128 bits are appended to the original message.

SHA-512 is based on a construction where data is absorbed into a sponge structure, and the output is derived by squeezing the input. During the absorbing phase, the data is XORed, and in the squeezing phase, the data undergoes state transformation.

While SHA-512 shares structural similarities with SHA-256, there are several key differences:

- *The message is divided into 1024-bit chunks.*
- *The initial hash values and rounds extend to 64 bits.*
- *Instead of 64 rounds, SHA-512 uses 80 rounds.*
- *The round constants are based on the first 80 prime numbers.*
- *The word size used for computation is 64 bits long.*
- *The fixed length of the message is 128 bits.*
- *vii) The shifting and rotation rounds are different.*

These points illustrate the structure of SHA-512, as depicted in Figure 6.

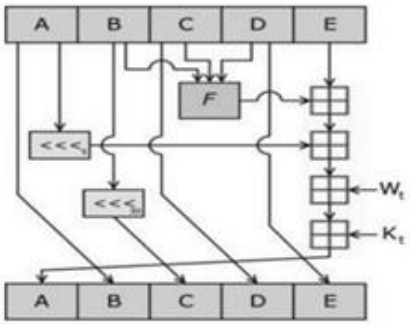

Figure 6. Structure of SHA-512

Using AES and a hashing algorithm, we have designed an architecture where a user can upload data to a cloud server. The receiver can then retrieve the data using a private key.

3. Proposed Method

In the proposed model, the data owner stores data on a cloud server, which is a prime target for intruders seeking to steal valuable information. To enhance security, a key is shared by the owner with the data user to access the data. The AES algorithm is employed to encrypt the data, rendering it unreadable to intruders and preventing data theft. Additionally, the SHA-512 hashing algorithm is used to link data blocks together. If any data blocks are found to be missing, our system allows the data user to identify which specific data block was attacked by hackers.

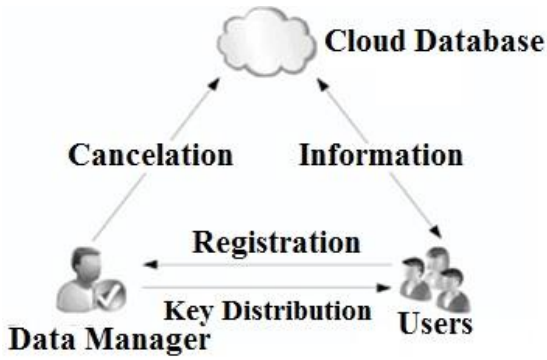

Figure 7. Process flow

As shown in the figure above, the plain data is first encrypted using the AES algorithm and then stored on the cloud as a hash value using the SHA-512 algorithm. Only authenticated users can decrypt the data. The figure illustrates four main interdependent modules, along with sub-modules that depend on both the main and sub-modules.

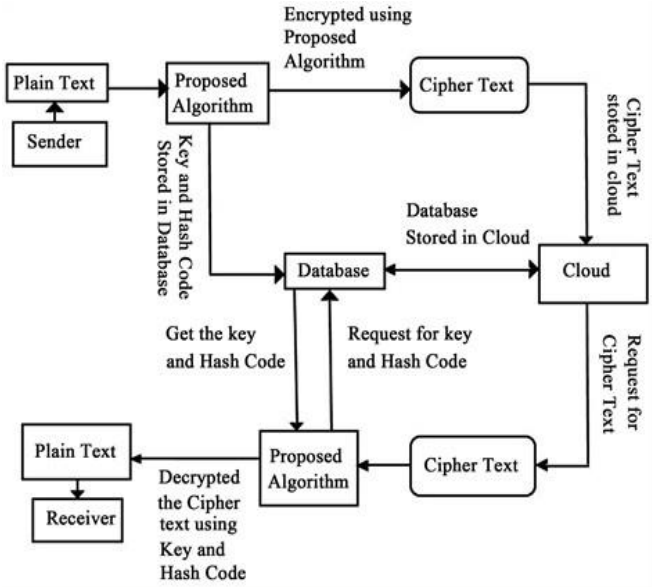

Figure 7.1 System flow

The data owner controls the storage space on the cloud server. The dealer (user) provides the necessary information to the owner, who decides whether to approve the user's request to use the services. Once approved, the user can store data on the cloud server. In some cases, the data owner may charge the user to access data stored on the cloud [3].

When data is stored on the server, a key is generated, which the user needs to access the data. To retrieve data from the cloud server, the user must send a request to the Key Manager. If the Key Manager denies the request, the user cannot access or download the file. All users wishing to access data stored on the cloud must have permission. The Key Manager has the authority to share the key with the respective user to grant access to the cloud data.

4. Results and Discussion

The encryption algorithm used in this approach renders the data unreadable, ensuring that even if an intruder manages to steal some part of the data, it cannot be decrypted. Decrypting the data is not merely a matter of reversing the encryption process; it also requires the same key that was used during encryption. This necessity for the correct key enhances the security level of the architecture.

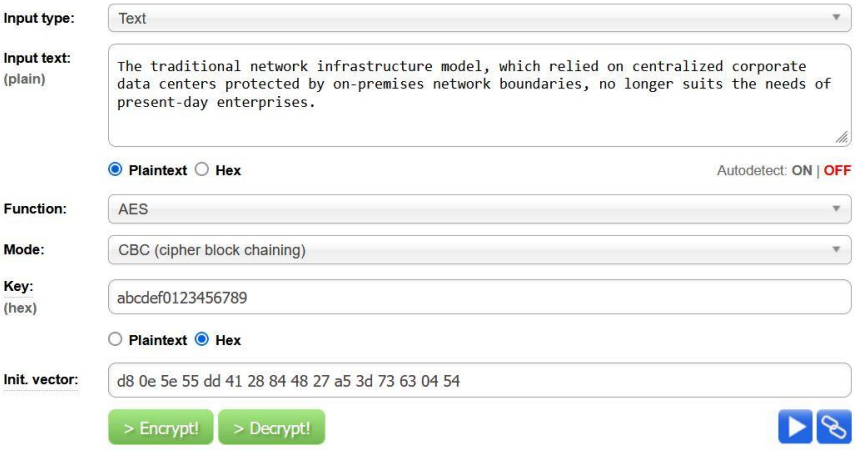

Figure 8. Providing input to AES algorithm

Steps of AES-CBC Encryption:

Divide the Plaintext into Blocks: The plaintext message is divided into fixed-size blocks (e.g., 128 bits for AES-128). If the final block is smaller than the block size, it is padded.

Initialize the IV:

The IV is generated and used for the encryption of the first block. The IV does not need to be kept secret but should be unique for each encryption session.

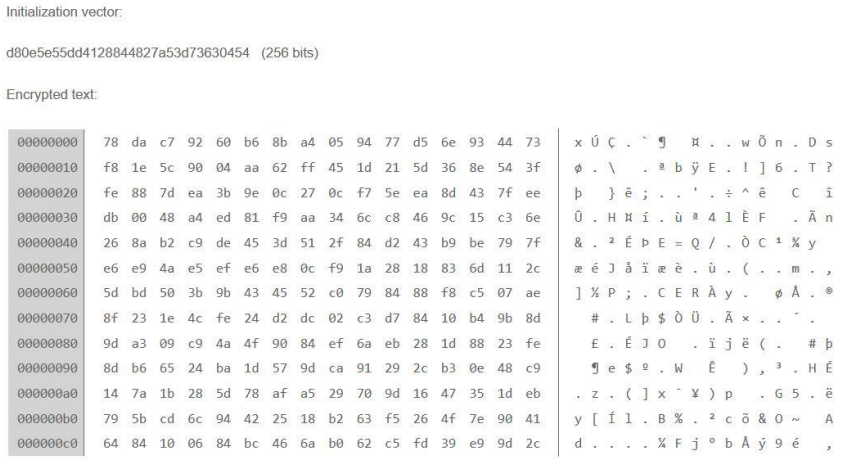

Figure 8.1 Shows the Encrypted text

Encryption Process:

First Block:

XOR the first plaintext block with the IV.

Encrypt the result using the AES algorithm and the secret key.

The output is the first ciphertext block.

Subsequent Blocks:

XOR each plaintext block with the previous ciphertext block.

Encrypt the result using the AES algorithm and the secret key.

The output is the ciphertext block for that plaintext block.

Output:

The sequence of ciphertext blocks forms the final encrypted message. The IV is typically prepended to the ciphertext to be used for decryption.

XOR the first plaintext block with the IV.

Encrypt the result using the AES algorithm and the secret key.

The output is the first ciphertext block.

Steps for Decryption

The encrypted text can be decrypted is explained in the following steps.

Initialize the IV:

Retrieve the IV from the encrypted message.

Decrypt the First Block:

Decrypt the first ciphertext block with AES and the index key.

XOR the result with the IV to retrieve the first plaintext block.

Decrypt Subsequent Blocks:

For each subsequent ciphertext block, decrypt it with AES and the index key.

XOR the result with the previous ciphertext block to retrieve the plaintext block.

Remove Padding:

If padding was added during encryption, remove it to obtain the original plaintext message.

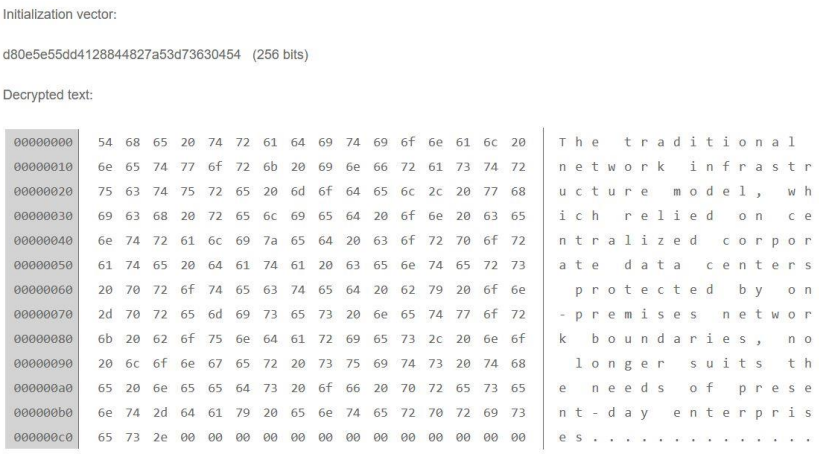

Figure 8.1 Shows the Decrypted text

Observations

We have also measured the time required for the AES algorithm to encrypt the data, as well as the time needed to perform the hashing operation on the data.

Table 1. Shows the Required time to perform operation by algorithm

File Size in Kilobytes	AES(256)	SHA-512	Time in miliseconds
12 KB	6.6 ms	20.595 ms	27.195 ms
25 KB	12.3 ms	37.370 ms	49.67 ms
5 KB	2.77 ms	9.040 ms	11.81 ms

Based on our observations, the time required to perform encryption and hashing operations on data depends on the system specifications. Our experiments were conducted on a personal computer with the following configuration: Intel(R) Core(TM) i5-8th Generation 8250U CPU @ 1.80GHz, 8GB RAM, running Windows 10 64-bit operating system version 1909. We also tested systems with lower configurations and found that these systems required more time to perform the same operations.

5. Conclusion

According to literature and studies, centralized storage has certain limitations. To enhance data security, decentralized cloud storage can be used. This paper proposes a secure and efficient method for storing data on the cloud. By utilizing blockchain-based cloud storage with data encryption, we achieve data security within a decentralized structure. The proposed model is well-suited for implementing a blockchain framework. The algorithms used in this system model are efficient, require less time, and provide high security for data stored on the cloud. This architecture makes the system more robust and resistant to various security attacks from unauthorized users attempting to steal or disclose user data for their own benefit.

References

11.1. Journal Article

[1] *Joan DAEMEN, Vincent RIJMEN, "On The Related-Key Attacks Against AES", Proceedings of The Romanian Academy, Series A, 2012.*

[2] *Mrs. Rohini Pise. Dr. Sonali Patil, "Enhancing Security of Data in Cloud Storage using Decentralized Blockchain", Proceedings of the Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks,* ***(2021)****, pp. 161-167.*

[3] *Edoardo Gaetani, Leonardo Aniello, Roberto Baldoni, Federico Lombardi, Andrea Margheri, and VladimiroSassone - Blockchain-based Database to Ensure Data Integrity in Cloud Computing Environments, In Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), Venice, Italy.*

[4] *H. Ako Muhamad Abdullah - Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data, Article · June 2017, Research Gate publication.*

[5] *Anita V. Mithapalli, Swati S. Joshi - A Framework for Secure Data Storage and Retrieval in Cloud Environment, ISSN: 2249 – 8958, Volume-9 Issue-2, December, 2019, International Journal of Engineering and Advanced Technology (IJEAT).*

[6] *Meiliana Sumagita and Imam Riadi - Analysis of Secure Hash Algorithm (SHA) 512 for Encryption Process on Web Based Application, International Journal of Cyber-Security and Digital Forensics (IJCSDF) 7(4): 373-381 The Society of Digital Information and Wireless Communications (SDIWC), 2018 ISSN: 2305-001.*

[7] *Akshay Babrekar, Prof. Rohini G. Pise - Public Key Encryption for Cloud Storage Attack using Blockchain, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-9 Issue-2, July 2020.*

[8] *Aradhana, Dr. S. M. Ghosh - Review Paper on Secure Hash Algorithm with Its Variants, DOI: 10.13140/RG.2.2.13855.05289, Research Gate publication.*

[9] *Avdhut Suryakant Bhise, Phursule R.N. - A Review of Role based Encryption System for Secure Cloud Storage, International Journal of Computer Applications (0975 – 8887) Volume 109 – No. 14, January 2015.*

[10] *Avdhut Suryakant Bhise, R. N. Phursule – Developing Secure Cloud Storage System by Integrating Trust and Cryptographic Algorithms with Role based Access Control, International Journal of Computer Applications (0975 – 8887) Volume 168 – No.10, June 2017.*

[11] *Peiming Xu, Shaohua Tang, Peng Xu, Qianhong Wu, Honggang Hu, Willy Susilo - Pract ical Multi-Keyword and Boolean Search Over Encrypted E-mail in Cloud Server.*

[12] *Jie Xu, Kaiping Xue, Senior Member, IEEE, Shaohua Li, Hangyu Tian, Jianan Hong, Peilin Hong, Nenghai Yu - Healthchain: A Blockchain-based Privacy Preserving Scheme for Large-scale Health Data.*

[13] *Andre Muller , Andre Ludwig and Bogdan Franczyk - Data security in decentralized cloud systems – "System Comparison, Requirements Analysis and Organizational Levels".*

[14] *V. J Tanupriya Choudhury, Vasudha Vashisht, Himanshu Srivastava – "A Secure Decentralized Cloud Computing Environment over Peer to Peer".*