



Decentralized Data Protection: Leveraging Blockchain And Dynamic AES For Cloud Security

M.Swathi
Dept. of IT
Assistant Professor

Dokku Siva Harini
Dept. of IT
Student

Gunapu Anudeep
Dept. of IT
Student

Ganesh Singh
Dept. of IT
Student

Atluri Vishnu Vardhan
Dept. of IT
Student

Abstract: Traditional cloud security systems rely on centralized key management, making them vulnerable to breaches, insider threats, and single points of failure. If an attacker gains access to the central repository, they can compromise the entire system, leading to massive data breaches. To address this issue, a decentralized approach using blockchain for key management is introduced, eliminating centralized storage by distributing encryption keys across multiple nodes. This method enhances security by generating unique AES encryption keys derived from file and block hash codes, ensuring that even if a key is compromised, only a single file is affected rather than the entire system. Additionally, blockchain's immutable ledger provides transparency, auditability, and resistance to unauthorized modifications, strengthening data integrity and confidentiality. By leveraging blockchain's decentralized nature and AES encryption's dynamic key generation, this approach significantly reduces security risks, making cloud storage highly resistant to hacking, unauthorized access, and tampering. The combination of these technologies creates a robust security framework that enhances trust, reliability, and resilience in cloud environments. Ultimately, integrating blockchain-based key management with dynamic AES encryption not only mitigates traditional security challenges but also provides a scalable and future-proof solution for secure cloud data storage and transmission.

Keywords: Decentralized Cloud Security, Blockchain-Based Key Management, Dynamic AES Encryption, Distributed Key Storage, Immutable Ledger, Data Integrity, Confidentiality, Cybersecurity, Cloud Data Protection

I. INTRODUCTION

Cloud computing has transformed content to data and retention, providing both users and companies with flexible and affordable options. However, the increasing reliance on cloud infrastructure has also introduced significant security challenges, particularly in encryption key

management. Traditional cloud security models rely on centralized key storage systems, where encryption keys are stored in a single repository managed by a trusted third party. This centralized approach makes cloud systems highly susceptible to breaches, insider threats, and single points of failure [2]. If an attacker gains access to the key management server, they can decrypt and compromise vast amounts of sensitive data, leading to severe financial and reputational damage. Additionally, internal threats from malicious insiders or human errors further increase the risk of unauthorized access. These security vulnerabilities call for an advanced and decentralized approach to encryption key management that eliminates single points of failure and enhances data protection.

By facilitating global and impenetrable encryption handling, the use of blockchain offers a strong answer to these security issues. Blockchain eliminates corporate key sources and drastically lowers the risk of a single-point attack by distributing key distribution among numerous nodes, in contrast to older approaches [11]. Blockchain makes it nearly hard for hackers to infiltrate the entire system by utilizing an autonomous network to guarantee that no one party has total control over encryption keys. Additionally, each purchase and access code is recorded in the transparent and unchangeable ledger of blockchain, which guards against unwanted changes and guarantees complete auditability. Because of this degree of security and openness, cloud environments are more trusted, and data protection is more resistant to online attacks.

To further strengthen cloud security, integrating Dynamic AES Encryption with blockchain-based key management provides an additional layer of protection. Traditional AES encryption uses static keys that remain unchanged for an extended period, increasing the risk of brute-force attacks if the key is exposed. However, Dynamic AES enhances security by generating unique encryption keys for each file or

data block based on hash codes derived from file properties and blockchain transactions. This method ensures that even if a key is compromised, only a single file is affected, rather than the entire system. The dynamic nature of the encryption process significantly reduces the attack surface, preventing unauthorized access and making it nearly impossible for hackers to decrypt large volumes of data. Additionally, since the encryption keys are never stored in a central location, their distribution across blockchain nodes makes them highly resistant to hacking and tampering.

By combining blockchain's decentralized security with the adaptability of Dynamic AES encryption, this approach creates a highly secure and scalable cloud data protection framework. Organizations can leverage this model to enhance data confidentiality, integrity, and availability while ensuring compliance with stringent security regulations. As cyber threats continue to evolve, the integration of blockchain-based key management and dynamic encryption will play a crucial role in securing sensitive cloud data against unauthorized access, tampering, and breaches. This innovative security model not only mitigates existing vulnerabilities but also paves the way for a more resilient and trustworthy cloud computing environment.

1.2 Motivation:

The growing dependence on cloud computing has heightened concerns about data security, particularly in encryption key management. Traditional centralized key storage systems are vulnerable to cyberattacks, insider threats, and single points of failure, putting sensitive data at risk. Blockchain technology offers a decentralized approach, eliminating centralized key repositories and enhancing security through distributed encryption key management. Integrating Dynamic AES encryption further strengthens protection by generating unique keys per file, minimizing damage from compromised keys. This combination ensures data integrity, confidentiality, and resilience against hacking. The motivation behind this approach is to create a secure, transparent, and tamper-resistant cloud security framework.

1.3 Objectives:

- Blockchain-based key management eliminates centralized storage, reducing risks from cyberattacks, insider threats, and unauthorized access while ensuring data integrity and confidentiality.
- Dynamic AES encryption generates unique keys for each file or data block, limiting exposure in case of key compromise and preventing large-scale data breaches.
- Smart contracts automate key distribution and retrieval, reducing administrative overhead, enhancing security, and ensuring seamless access control in cloud environments.
- Blockchain's immutable ledger and decentralized access control mechanisms enable secure, transparent, and verifiable data sharing among authorized users.

1.4 Literature Survey:

Zhang et al. approach enables fully outsourced policy hiding, ensuring that access control policies remain undisclosed to unauthorized entities while allowing secure data retrieval based on user location. By leveraging cryptographic techniques, their framework strengthens data

privacy and reduces exposure to unauthorized access attempts[1]. However, their method primarily focuses on access control rather than comprehensive encryption key management, leaving room for potential improvements in handling encryption keys dynamically to enhance cloud security further.

Sharma and Singh system utilizes blockchain's decentralized nature to distribute encryption keys securely, reducing risks from single points of failure and insider threats[2]. By integrating smart contracts, the framework automates key distribution, ensuring efficient and tamper-proof management. While their approach significantly improves key security and access control, it does not explicitly implement dynamic encryption techniques such as Dynamic AES, which could further enhance security by ensuring unique encryption keys for each file or data block.

Zyskind et al. explored the use of framework leverages blockchain's immutable ledger and cryptographic mechanisms to give users full control over their data, reducing reliance on centralized entities[3]. By enabling decentralized identity management and access control, their approach minimizes risks from unauthorized access and data breaches. However, while their model enhances privacy, it does not specifically address encryption key management for cloud environments, leaving a gap in securing large-scale cloud storage systems against advanced cyber threats.

Zhang et al. conducted a comprehensive survey on blockchain security and privacy, analyzing key challenges and potential solutions. They highlighted how blockchain's transparency, decentralization, and cryptographic security mechanisms enhance data integrity and trust[4]. The study also discussed vulnerabilities, such as key exposure risks and smart contract exploits, which could be mitigated with dynamic encryption techniques. While their research provides a strong foundation for understanding blockchain security, it does not focus specifically on integrating encryption algorithms like Dynamic AES for enhanced cloud data protection. This presents an opportunity to combine blockchain-based key management with advanced encryption strategies for robust cloud security solutions.

Huang et al. proposed a decentralized key management framework for 5G mobile devices using blockchain technology. Their approach eliminates traditional centralized key storage, reducing security vulnerabilities associated with single points of failure and insider threats[5]. By leveraging blockchain's distributed ledger, they ensure secure key authentication, efficient key revocation, and seamless device handovers in 5G networks. However, while their framework enhances security in mobile communications, it does not focus on cloud data storage and encryption, leaving room for further improvements in integrating blockchain-based key management with dynamic encryption techniques for cloud security applications.

Liu et al. introduced a blockchain-based privacy-preserving scheme for decentralized cloud storage, addressing data confidentiality and integrity issues in cloud environments. Their model utilizes blockchain's transparency and decentralization to manage access control and prevent unauthorized modifications[6]. Additionally, they incorporate cryptographic mechanisms to enhance privacy while ensuring scalability and efficiency. However, their study does not explore the implementation of dynamic encryption techniques, such as Dynamic AES, which could

further improve data security by generating unique encryption keys per file or data block. Integrating blockchain with adaptive encryption methods could create a more robust cloud security framework.

1.5 Advanced Encryption Standard

AES is one of the most widely used symmetric encryption algorithms due to its strong security and efficiency. However, traditional AES implementations rely on static keys, making them vulnerable to key compromise. In our project, we introduce Dynamic AES Encryption, where encryption keys are not fixed but dynamically generated based on file and block-specific attributes. This technique enhances security by ensuring that each file or data block has a unique encryption key, significantly reducing the impact of key leakage. Dynamic AES leverages cryptographic hash functions to derive encryption keys on demand, eliminating the need for persistent key storage and reducing attack vectors such as brute-force attacks and key exposure.

The core mechanism behind Dynamic AES involves key generation using cryptographic hash functions, such as SHA-256 or SHA-512. For a given file F , we compute a unique key using a combination of file-specific attributes, such as the file hash and block index. The dynamic key K_{dyn} is derived as:

$$K_{\text{dyn}} = H(F_{\text{hash}} \parallel B_{\text{index}} \parallel S)$$

Where H is a cryptographic hash function (e.g., SHA-256), F_{hash} is the hash of the entire file, B_{index} represents the index of the data block being encrypted, S is a system-generated salt to introduce randomness and prevent dictionary attacks.

This dynamic key is then used in the AES encryption process, ensuring that even if one key is compromised, it cannot be used to decrypt other files or blocks.

One of the key optimizations in Dynamic AES is the use of on-the-fly key generation and ephemeral key storage. Instead of storing encryption keys permanently, the system generates them as needed and discards them after use. When a user requests decryption, the system re-computes the dynamic key using the same deterministic function rather than retrieving it from storage. This significantly enhances security as an attacker cannot obtain encryption keys from a compromised database. Additionally, the use of block-wise dynamic keys prevents large-scale data breaches. If an attacker manages to retrieve a key for one block, they cannot decrypt the entire file since each block has a unique key.

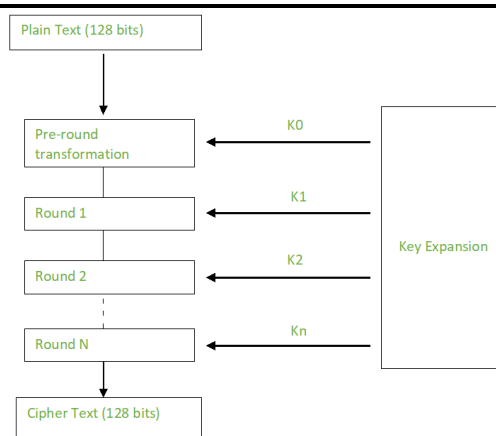


Fig1:Advanced Encryption Standard

Furthermore, Dynamic AES can be integrated with blockchain-based key management for additional security. Blockchain ensures the integrity of dynamic key generation parameters by immutably storing file hashes and block indexes. Smart contracts enforce access control rules, ensuring that only authorized users can compute the necessary encryption keys. This synergy between Dynamic AES and blockchain enhances cloud security by eliminating centralized key storage risks, making encryption resistant to common threats such as insider attacks, ransomware, and unauthorized modifications. By leveraging keyless encryption, block-specific key variation, and immutable audit trails, Dynamic AES provides a highly resilient and scalable encryption framework suitable for secure cloud environments.

II. PROPOSAL METHOD

Our proposed Decentralized Cloud Security Framework integrates Dynamic AES Encryption with Blockchain-based Key Management to provide a robust and tamper-resistant cloud storage solution. The flowchart outlines the step-by-step process of securing cloud data using encryption and blockchain validation. Initially, a user signs up and logs into the system, ensuring authenticated access. Once logged in, the user uploads a file that needs to be securely stored in the cloud. Before the file is transmitted to the storage server, it undergoes Dynamic AES Encryption, ensuring that each file or data block has a unique encryption key. This eliminates the risks associated with static key usage, making it difficult for attackers to compromise stored data.

After encryption, the secured file is stored in the cloud, while its cryptographic hash is computed and recorded on the blockchain. The hash function plays a crucial role in maintaining data integrity by generating a unique fingerprint for each file. Since blockchain is an immutable ledger, storing file hashes on the blockchain prevents unauthorized modifications and enables real-time verification of file integrity. Even if an attacker gains access to the cloud storage and modifies the data, the blockchain-stored hash remains unchanged, allowing quick detection of inconsistencies. This decentralized approach significantly enhances security compared to traditional centralized key management systems, which are prone to attacks.

To demonstrate the robustness of our security model, the flowchart includes an attack simulation, where an external entity attempts to tamper with the stored file. When a user tries to download the file, the system first performs Cloud Attack Detection, comparing the current file state with the expected hash stored in the blockchain. If a modification is detected, the system proceeds to Blockchain Attack

Detection to verify whether the blockchain record has been altered. Since blockchain is tamper-proof, any discrepancies indicate an attack attempt on the cloud storage rather than the blockchain itself. If no modification is detected, the system allows a direct download of the file without additional verification.

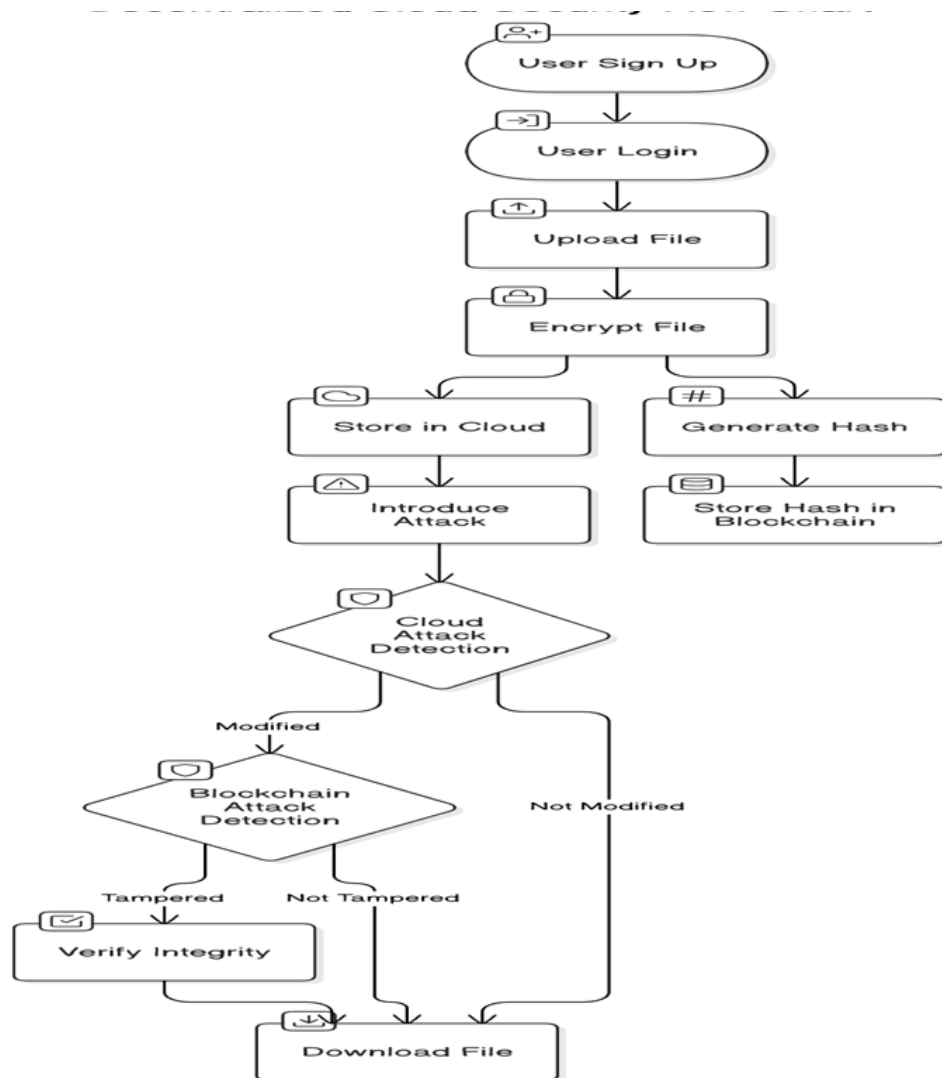


Fig2:ProposalDiagram

If an attack is confirmed, the system initiates an integrity verification process. The tampered file is cross-checked against its original blockchain hash, and security protocols are activated to prevent further damage. Since blockchain provides an immutable audit trail, all attempted modifications are logged, enabling forensic analysis and strengthening the cloud security infrastructure. By leveraging blockchain for attack detection, our method ensures that unauthorized modifications are instantly flagged and mitigated, making it highly resistant to data breaches, ransomware attacks, and unauthorized alterations.

Finally, after successful verification, the user is allowed to download the file, ensuring that only original, unmodified data is retrieved. This structured security approach ensures confidentiality, integrity, and availability of data across distributed cloud networks. The integration of Dynamic AES

Encryption and Blockchain-based Key Management in this framework provides a next-generation cloud security solution that is resistant to key exposure, unauthorized data modifications, and insider threats. By eliminating centralized key storage and enforcing decentralized validation, our system enhances cloud security, making it highly resilient to modern cyber threats.

III.IMPLEMENTATION

3.1User Authentication & Access Control

In our decentralized cloud security framework, User Authentication & Access Control is implemented using blockchain-based identity management. Traditional authentication systems rely on centralized databases, making them vulnerable to hacking, credential theft, and insider attacks. Instead, we use blockchain-based decentralized identity verification, ensuring that user credentials are securely managed without a single point of failure. Each user

is assigned a cryptographic identity, stored immutably on the blockchain. When a user attempts to access cloud storage, their identity is verified through a smart contract, which checks their access privileges and permissions. This eliminates the need for centralized authentication servers, reducing risks related to password leaks, phishing, and unauthorized access attempts. Additionally, multi-factor authentication (MFA) mechanisms, such as biometric verification and cryptographic signatures, are integrated to further enhance security, ensuring that only legitimate users can access cloud storage and services.

3.2 Dynamic AES Encryption Module

To ensure strong data encryption, we incorporate a Dynamic AES Encryption Module with periodic key regeneration. Unlike traditional AES encryption, which relies on static keys, our model dynamically generates encryption keys based on file-specific attributes such as file hash, block index, and system-generated salt. This prevents attackers from exploiting stolen keys, as the encryption keys are constantly changing. Additionally, on-demand key generation ensures that encryption keys are never stored permanently, reducing risks associated with key exposure. By integrating this approach, our system ensures strong confidentiality and resilience against cryptographic attacks, such as brute-force and key compromise attacks.

3.3 Blockchain-Based Data Integrity & Storage

Data integrity is a critical aspect of cloud security, and our Blockchain-Based Data Integrity & Storage Module ensures tamper-proof verification of files stored in the cloud. Each time a file is uploaded, a cryptographic hash of the file is generated and stored immutably on the blockchain. This hash acts as a fingerprint, allowing users to verify whether the file has been altered. Whenever a file is accessed or downloaded, its current hash is compared with the stored blockchain hash. If there is a mismatch, the system immediately flags the modification attempt, indicating a possible attack. Since blockchain is immutable, no unauthorized entity can alter or delete these integrity proofs, making our system highly resistant to data tampering, ransomware, and insider threats.

3.4 Decentralized File Sharing & Access Management

File sharing in traditional cloud environments often involves centralized access control lists (ACLs), which are prone to unauthorized modifications and privilege escalation attacks. Our Decentralized File Sharing & Access Management Module leverages smart contracts to enforce secure and automated access control policies. When a user wants to share a file, they specify access conditions, which are recorded on the blockchain. These conditions could include time-based access, role-based permissions, or cryptographic authentication. Only users who meet these predefined conditions are granted access, eliminating the risk of unauthorized file sharing. The use of smart contracts ensures that access control policies are self-executing and tamper-proof, preventing malicious modifications by unauthorized parties. This module strengthens data privacy, secure collaboration, and controlled file distribution in cloud environments.

3.5 Decentralized Backup & Recovery

To enhance data availability and fault tolerance, our system implements a Decentralized Backup & Recovery Module. Traditional cloud backup systems rely on centralized storage providers, making them vulnerable to server failures, cyberattacks, and accidental data loss. Instead, we distribute encrypted file backups across multiple decentralized nodes in a peer-to-peer network. This ensures redundancy, meaning that even if one node is compromised or goes offline, users can still recover their data from other nodes. Our system employs erasure coding and distributed ledger technology to ensure that file recovery is efficient and tamper-resistant. Additionally, blockchain smart contracts automate the recovery process, ensuring that only authorized users can retrieve their backups. By decentralizing data storage, we eliminate reliance on a single cloud provider, making our security framework highly resilient against data breaches and service disruptions.

3.6 Smart Contract-Based Access Logs & Auditing

Security monitoring and auditing are essential for detecting unauthorized access attempts, policy violations, and insider threats. Our Smart Contract-Based Access Logs & Auditing Module records every file access, modification, and deletion attempt on the blockchain. Since blockchain records are immutable, audit logs cannot be altered, ensuring transparent and tamper-proof activity tracking. This module enables real-time monitoring of who accessed a file, when it was accessed, and what modifications were made, providing forensic evidence in case of security breaches. Additionally, anomaly detection algorithms can analyze these logs to identify suspicious behavior, such as multiple failed access attempts or unauthorized privilege escalations. By integrating blockchain-based auditing, our system ensures regulatory compliance, enhanced visibility, and proactive threat detection in cloud environments.

IV. RESULTS

In our decentralized cloud security framework, user sign-up details are securely stored on the blockchain, ensuring tamper-proof identity management. Each registration transaction is recorded immutably, eliminating risks associated with centralized databases. When retrieving logs, the system fetches detailed blockchain records, including Block Number, Transaction ID, Hash Code, Timestamp, and User Credentials (hashed for privacy). These logs provide transparent and auditable insights into all authentication events. By displaying these blockchain-based logs, we ensure enhanced security, accountability, and real-time monitoring of user activities.

Fig3: Sign Up Page

In our system, users can select and upload files, then press a button to store file data in both Blockchain and Drive HQ Cloud while generating a unique hash code for integrity verification. The Blockchain logs maintain a tamper-proof record containing details such as Block Number, Transaction ID, and Hash Code, ensuring transparency and security. Users have the ability to upload multiple files and later retrieve a complete log of all stored files, providing an immutable and auditable trail of data transactions. This decentralized approach eliminates the risks associated with centralized key storage and ensures that every file is securely recorded with its corresponding cryptographic hash.



Fig4: HashCode

To simulate an attack, file content is deliberately modified, leading to a hash mismatch between the altered file and the original stored hash. Since Drive HQ Cloud lacks built-in verification mechanisms, it continues to list uploaded files without detecting tampering or unauthorized changes. In contrast, Blockchain actively monitors and verifies file integrity, immediately flagging any inconsistencies. This highlights the critical role of blockchain-based security in ensuring data authenticity, confidentiality, and resilience against cyber threats, making it an essential solution for secure cloud storage and data protection.

CLOUD SECURITY vs BLOCKCHAIN SECURITY				
Upload File to Cloud & Blockchain Introduce Attack Cloud Attack Detection Blockchain Attack Detection Logout				
Owner Name	File Name	Attack Detection Result	Upload Date	Download File
anudeep	conclusion ppt.txt	Not Supported	2025-03-09 12:34:53.687023	Download
anudeep	sample.txt	Not Supported	2025-03-09 12:37:26.237898	Download
anudeep	bk.txt	Not Supported	2025-03-09 12:39:16.360736	Download

Fig5:List of Uploaded Files

Our system stores and verifies uploaded files using Blockchain, which displays a list of uploaded files along with their hash codes. When a file is unaltered, its hash matches the stored value, and Blockchain marks it in green text. If a file is tampered with, the hash check fails, triggering an ‘Attack detected’ alert in red text. This ensures real-time attack detection and integrity verification. Unlike traditional cloud storage, Blockchain actively monitors and validates file integrity, preventing undetected tampering. Users can securely download files by clicking the ‘Download’ link, ensuring they retrieve only verified, untampered files.

CLOUD SECURITY vs BLOCKCHAIN SECURITY				
Upload File to Cloud & Blockchain Introduce Attack Cloud Attack Detection Blockchain Attack Detection Logout				
Owner Name	File Name	File Hashcode	Attack Detection Result	Upload Date
anudeep	conclusion ppt.txt	data70202aafba40553b117b297fd4a72ed6ecf441ab31bd266f3ce3ba4ec7b	Attack Detected! File Data Modified	2025-03-09 12:34:53.687023
anudeep	sample.txt	698f0a79133610a4f3da2e7789ec9af9183574943dfb224b72b468c3b58257	Attack Detected! File Data Modified	2025-03-09 12:37:26.237898
anudeep	bk.txt	d6c58055f7d8d60be76ed8afcc468309646af39ac81eb73c847d44c7ae8fd	Attack Detected! File Data Modified	2025-03-09 12:39:16.360736

Fig6:Blockchain Attack Detection

V. CONCLUSION:

In conclusion, the integration of Blockchain and Dynamic AES Encryption in decentralized cloud security enhances data integrity, confidentiality, and tamper resistance. Unlike traditional cloud storage, which lacks built-in verification mechanisms, our approach ensures real-time file integrity checks, preventing unauthorized modifications. By leveraging blockchain for key management and log transparency, we eliminate single points of failure while maintaining a secure and auditable record of all transactions. The automated attack detection mechanism further strengthens cloud security by flagging any unauthorized file alterations, ensuring users access only authentic and untampered files. Additionally, decentralized storage and smart contract-based access control enhance secure file sharing and data redundancy, making the system highly resilient to cyber threats. Our approach effectively mitigates risks such as data breaches, insider attacks, and ransomware, offering a robust security framework for cloud-based applications.

VI. FUTURE WORK:

For future work, we plan to further enhance scalability and efficiency by optimizing blockchain consensus mechanisms, ensuring faster transactions and lower computational overhead. Additionally, integrating ZKPs and Homomorphic Encryption will provide stronger privacy-preserving authentication and computation capabilities. We also aim to extend our attack detection system by incorporating AI-driven anomaly detection, enabling proactive security threat identification. Another avenue for improvement is the implementation of IPFS for decentralized storage, reducing reliance on third-party cloud providers while improving data availability and performance. As cloud-based threats continue to evolve, our decentralized security model will adapt by incorporating self-healing mechanisms and enhanced cryptographic techniques, ensuring next-generation cloud security solutions.

VII.References:

1.Zhang Y, Chen X, Li J, Wong D, Lee P, Lou W. Ensuring location-based access control for cloud-stored data with fully outsourced policy hiding. IEEE Transactions on Information Forensics and Security. 2016;11(8): 1941-1955.

2.Sharma PK, Singh S. Blockchain-based efficient key management for secure cloud data sharing. Future Generation Computer Systems. 2020;111: 620-633.

3.Zyskind G, Nathan O, Pentland A. Decentralizing privacy: Using blockchain to protect personal data. 2015 IEEE Security and Privacy Workshops (SPW). 2015: 180-184.

4.Zhang R, Xue R, Liu L. Security and privacy on blockchain. ACM Computing Surveys. 2019;52(3): 1-34.

- 5.Huang J, Kang J, Yu R, Zhang Y, Xie S, Liu Y. Decentralized key management for 5G mobile devices based on blockchain. IEEE Communications Magazine. 2019;57(7): 50-56.
- 6.Liu B, Wang H, Li L, Sun X, Zhang J. A blockchain-based privacy-preserving scheme for decentralized cloud storage. Future Generation Computer Systems. 2021;116: 364-376.
- 7.Singh PK, Kumari A, Singh AK, Gupta R, Tanwar S, Kumar N. Blockchain-based decentralized key management for secure cloud data storage. Sensors. 2020;20(16): 4506.
- 8.Wang J, Huang Y, Lin S, Wang W. Blockchain-based secure key management system for cloud computing. Journal of Network and Computer Applications. 2022;200: 103366.
- 9.Jiang Y, Wang Y, Liu W, Sun C. Blockchain-based access control for secure data sharing in cloud environments. IEEE Internet of Things Journal. 2021;8(6): 4248-4261.
- 10.Li X, Ma X, Sun X. Secure and efficient key management in cloud environments using blockchain. Information Sciences. 2022;582: 72-86.
- 11.Ming Y, Liu Y, Zhang Y, Zhou Z. Secure cloud storage using blockchain and dynamic encryption. Computers & Security. 2021;108: 102331.
- 12.He Y, Li H, Shi J, Zhou Z. A decentralized access control framework for cloud storage based on blockchain and cryptography. Future Internet. 2020;12(9): 141.
- 13.Al-Mousa R, Al-Sa'di K, Hassan R, Hossain MA. Blockchain-based access control and data security in cloud computing. IEEE Access. 2021;9: 101268-101281.
- 14.Patel V, Shah P. Decentralized key management for secure data sharing using blockchain. Security and Privacy. 2022;5(3): e211.
- 15.Wu J, Song Z, Li J, Zhao F. An enhanced blockchain-based access control system for cloud storage. IEEE Transactions on Cloud Computing. 2022;10(2): 1424-1435. 2022.
- 26.Zhao Y, Zhang J, Wang L, Wang J. Blockchain-enabled secure data storage and sharing framework for cloud environments. Information Sciences. 2022;583: 202-218.

