Sanjai gm, 2025, 13:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

A Blockchain-Based Efficient Data Integrity Verification Scheme in Multi- Cloud Storage

Sanjai Gm, Assistant Professor K Kumutha
Department of computer applications
VISTAS, Chennai-india

Abstract- A Blockchain-Based Effective Data Integrity Verification System for Multi-Cloud Storage Systems is presented in this study. Ensuring data integrity across dispersed storage platforms becomes essential as more and more businesses use multi-cloud setups to manage their data. Conventional data verification techniques have drawbacks such centralized control, lack of transparency, and attack vulnerability. Using blockchain technology, which guarantees immutability, transparency, and decentralization, our suggested approach provides a more effective and safe way to confirm data integrity in multi-cloud systems. Real-time auditing and tamper detection are made possible by the blockchainledger, which lowers the possibility of data alteration or illegal access. This method is a promising answer to the changing requirements of cloud-based data management since it increases data trustworthiness, reduces security flaws, and facilitates scalability.

Keywords- Blockchain, Data Integrity, Multi-Cloud Storage, Data Auditing, Cloud Security, Smart Contracts, Decentralized Verification, Proof of Storage, Cryptographic Techniques

I. INTRODUCTION

Cloud computing's virtualized infrastructure has grown in popularity as a target for sophisticated cyberattacks. In order to identify sophisticated threats in virtualized infrastructures, this study suggests a revolutionary big data-based security analytics method. The Hadoop Distributed File System (HDFS) houses the network and user application logs that are periodically gathered from the guest virtual machines (VMs). Then, using graph-based event correlation and Map Reduce parser-based identification of possible attack paths, attack features are extracted.

Next, two-step machine learning is used to determine the presence of an attack. Specifically, logistic regression is used to determine the conditional probabilities of an attack in relation to the attributes, and belief propagation is used to determine the belief that an attack exists based on those attributes. Experiments are carried out to compare the suggested method with current security methods for virtualized infrastructure and to assess it using well-known malware. The outcomes demonstrate the effectiveness of our suggested method in identifying assaults with the least amount of performance overhead.

Existing Research

Big data processing with on-hand database administration becomes challenging due to its large volume and complexity. Big data storage in the cloud is a good choice since it can store large amounts of data and efficiently handle a large number of user access requests. Since data owners cannot completely trust cloud servers, data security becomes a key worry when big data is hosted in the cloud.

A disadvantage is that allowing data owners to retrieve the information under the new access policy and then returning it to the cloud is a simple implementation.

Data owners must bear a significant computational load and a high communication overhead with this approach.

II. PROPOSED SYSTEM

A promising method for guaranteeing end-to-end data security in cloud storage systems is attribute-

based encryption, or ABE. Ensuring the following requirements is the biggest problem when outsourcing policy updates to the cloud:

- Correctness: By using the old decryption technique, users with the necessary qualities should still be able to decrypt the material encrypted under the new access policy.
- Completeness: Any kind of access policy should be able to be updated using the policy updating approach.
- Security: Changing the policy shouldn't compromise the access control system's security or cause any new issues.

Advantages

Benefit: We define the policy updating problem in ABE systems and create a novel approach for server-side policy updating outsourcing.

- To enable effective dynamic policy updating, we suggest an expressive and effective data access control approach for big data.

III. MODULE DESCRIPTION

Module for Authentication
All of the verified user's information is contained in this module. If a user is authenticated, he can access his login; otherwise, he cannot access it without his username and password. The process of confirming a user's identification via acquiring credentials and utilizing them to confirm the user's identity is known as authentication. The authorization process begins if the credentials are legitimate. The authorization process is always the next step after authentication. Administrators take on these duties as volunteers after completing a community review procedure. They're not behaving like consumers. Their tools are never necessary, and they must never be used to obtain an edge in a dispute involving who needs access to their data.
2. Security Guidelines In cloud storage systems, Module Attribute-Based Encryption (ABE) has become a viable method for guaranteeing end-to-end data security. It enables data owners to specify access policies and encrypt data in accordance with them, limiting data decryption to users whose attributes meet access policy requirements. Since data owners may alter data access restrictions often and dynamically, policy updating becomes a major concern as more and more businesses and organizations outsource their data to the cloud.

3. An auditor of data

Each authority is in charge of overseeing the characteristics of users within its jurisdiction and is independent of the others. Additionally, it creates a secret key for every user based on their attributes and a secret/public key pair for every attribute in its domain.

4. Module for Server Control

Data owners' data is stored on the cloud server, which also gives users access to their data. Updating cipher texts from outdated access policies to updated ones is another duty of the server.

Consumer of Data

Every user is given a global user identity and has unrestricted access to the server's ciphertexts. Only when the attributes meet the access policy specified in the cipher text can the user decrypt the cipher text.

Algorithm

Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S National Institute of Standards and Technology (NIST) in 2001. AES is widely used today as it is a much stronger than DES and triple DES despite being harder to implement.

Points to remember

- AES is a block cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each.

That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text as output. AES relies on substitution-permutation network principle which means it is performed using a series of linked operations which involves replacing and shuffling of the input data.

Working of the cipher:

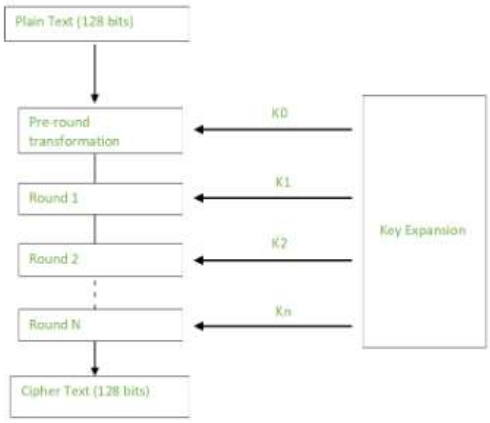
AES performs operations on bytes of data rather than in bits. Since the block size is 128 bits, the cipher processes 128 bits (or 16 bytes) of the input data at a time.

The number of rounds depends on the key length as follows:

- 128 bit key – 10 rounds
- 192 bit key – 12 rounds
- 256 bit key – 14 rounds

Creation of Round keys:

A Key Schedule algorithm is used to calculate all the round keys from the key. So the initial key is used to create many different round keys which will be used in the corresponding round of the encryption.



Encryption:

AES considers each block as a 16 byte (4 byte x 4 byte = 128) grid in a column major arrangement.

```
[ b0 | b4 | b8 | b12 |
| b1 | b5 | b9 | b13 |
| b2 | b6 | b10| b14 |
| b3 | b7 | b11| b15 ]
```

Each round comprises of 4 steps:

- SubBytes
- ShiftRows
- MixColumns
- Add Round Key

The last round doesn't have the MixColumns round.

The SubBytes does the substitution and ShiftRows and MixColumns performs the permutation in the algorithm.

SubBytes:

This step implements the substitution.

In this step each byte is substituted by another byte. Its performed using a lookup table also called the S-box. This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a compliment of the current byte. The result of this step is a 16 byte (4 x 4) matrix like before.

The next two steps implement the permutation.

ShiftRows:

This step is just as it sounds. Each row is shifted a particular number of times.

- The first row is not shifted
- The second row is shifted once to the left.
- The third row is shifted twice to the left.
- The fourth row is shifted thrice to the left.

(A left circular shift is performed.)

MixColumns:

This step is basically a matrix multiplication. Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.

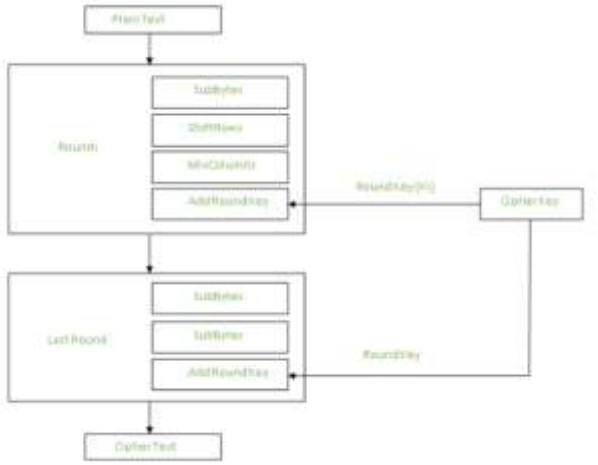
This step is skipped in the last round.

```
[ c0 ]       [ 2  3  1  1 ] [ b0 ]
| c1 |  =    | 1  2  3  1 |    | b1 |
| c2 |     | 1  1  2  3 |    | b2 |
[ c3 ]     [ 3  1  1  2 ]    [ b3 ]
```

Add Round Keys:

Now the resultant output of the previous stage is XOR-ed with the corresponding round key. Here, the 16 bytes is not considered as a grid but just as 128 bits of data.

```
[ b0 | b1 | b2 | b3 ]        [ b0 | b1 | b2 | b3 ]
| b4 | b5 | b6 | b7 |   ->   | b5 | b6 | b7 | b4 |
| b8 | b9 | b10| b11|        | b10| b11| b8 | b9 |
[ b12| b13| b14| b15]        [ b15| b12| b13| b14]
```



After all these rounds 128 bits of encrypted data is given back as output. This process is repeated until all the data to be encrypted undergoes this process.

Decryption:

The stages in the rounds can be easily undone as these stages have an opposite to it which when performed reverts the changes.Each 128 blocks goes through the 10,12 or 14 rounds depending on the key size.

The stages of each round in decryption is as follows:

- Add round key
- Inverse MixColumns
- ShiftRows
- Inverse SubByte

The decryption process is the encryption process done in reverse so i will explain the steps with notable differences.

Inverse Mix Columns:

This step is similar to the Mix Columns step in encryption, but differs in the matrix used to carry out the operation.

$$\begin{bmatrix} b0 \\ b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix} \begin{bmatrix} c0 \\ c1 \\ c2 \\ c3 \end{bmatrix}$$

Inverse SubBytes:

Inverse S-box is used as a lookup table and using which the bytes are substituted during decryption.

Summary:

AES instruction set is now integrated into the CPU (offers throughput of several GB/s)to improve the speed and security of applications that use AES for encryption and decryption. Even though its been 20 years since its introduction we have failed to break the AES algorithm as it is infeasible even with the current technology. Till date the only vulnerability remains in the implementation of the algorithm.

IV.CONCLUSION

The project "Big Data Based Security Analytics for Protecting Virtualized Infrastructures in Cloud Computing" has shown encouraging outcomes after the successful development, implementation, and testing stages. Using example datasets, the system was rigorously assessed, and the results verified that the main goals of the project were successfully met. By utilizing big data analytics, the developed framework effectively improves the security of virtualized cloud infrastructures by enabling prompt threat identification and mitigation.

The solution was especially designed to cut down on the amount of manual intervention and time waste that come with using more conventional security monitoring methods. Additionally, the system architecture was created with flexibility and scalability in mind, enabling the smooth integration of upgrades or adjustments in the future as security and technological needs change.

All things considered, the project satisfies the end-user needs as well as the technical standards, providing a solid, safe, dependable, and effective solution. It establishes a strong basis for future study and development, especially in the fields of automated incident response systems in multi-cloud and hybrid cloud environments, real-time threat intelligence, and advanced predictive analytics.

REFERENCES

1. R. S. Sahu and S. K. Sahay, "Security Analytics Framework for Cloud Using Big Data Analytics: A Survey," *IEEE Access*, vol. 8, pp. 134460–

134491, 2020, doi: 10.1109/ACCESS.2020.3009506.

2. A. Abbasi, R. Shams, and A. Gani, "A Survey on Big Data Analytics in the Cloud for Cybersecurity," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–37, June 2021, doi: 10.1145/3372134.
3. M. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1020–1051, 2020, doi: 10.1109/COMST.2020.2976866.
4. Y. Zhang, R. H. Deng, and J. Weng, "Secure and Efficient Big Data Storage and Sharing Scheme for Cloud Computing," *IEEE Transactions on Cloud Computing*, vol. 9, no. 1, pp. 116–129, Jan.–March 2021, doi: 10.1109/TCC.2020.2983047.
5. S. Singh and N. Singh, "Security in Cloud Computing Based on Big Data Analytics: A Review," in *Proceedings of the 2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, Mar. 2022, pp. 1–6, doi: 10.1109/ICCMC53470.2022.9754015.
6. D. Gupta and S. K. Sahu, "Big Data Analytics in Cloud Computing for Cybersecurity: A Systematic Review and Future Directions," *Future Generation Computer Systems*, vol. 137, pp. 185–200, Feb. 2024, doi: 10.1016/j.future.2022.12.011.
7. B. Varghese and R. Buyya, "Next Generation Cloud Computing: New Trends and Research Directions," *Future Generation Computer Systems*, vol. 79, pp. 849–861, Feb. 2021, doi: 10.1016/j.future.2017.09.020.
8. H. Wu, F. Li, and B. Mao, "Cloud Security Auditing Using Blockchain-Based Cryptographic Proofs," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 540–553, March-April 2023, doi: 10.1109/TDSC.2022.3140536.
9. R. Jalili and S. M. Fakhraie, "Artificial Intelligence-Based Intrusion Detection Systems in Cloud Computing: Current Challenges and Future Trends," *IEEE Access*, vol. 9, pp. 146127–146145, 2021, doi: 10.1109/ACCESS.2021.3122964.
10. L. Zha, P. Yang, and Z. Wei, "Privacy-Preserving Security Audits for Cloud Storage: A Blockchain-Based Approach," *IEEE Transactions on Services Computing*, early access, 2024, doi: 10.1109/TSC.2024.3310807.
11. V. Chang and M. Ramachandran, "Towards Achieving Data Security with the Cloud Computing Adoption Framework," *IEEE Transactions on Services Computing*, vol. 15, no. 2, pp. 760–773, Mar.-Apr. 2022, doi: 10.1109/TSC.2020.2986019.
12. M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "A First Look at Cloud-based Security Analytics," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 541–555, Mar. 2021, doi: 10.1109/TNSM.2020.3022496.