INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN : 2454-9924

HADOOP DISTRIBUTED FILE SYSTEM PERSPECTIVE ON ATTRIBUTE BASED HONEY ENCRYPTION ALGORITHM FOR SECURING BIG DATA

Dr. M. CHANDRAKALA, MCA., M.Phil., Ph.D., B. PRIYANKA, V. SHALLY STOBINEE.

Assistant professor, Department of Computer Application, Idhaya college of arts and science for women-08.

Idhaya College of Arts and Science for Women, puducherry;

Idhaya College of Arts and Science for Women, puducherry;

ABSTRACT:

The increasing volume and complexity of big data pose significant challenges for ensuring data security and confidentiality. Traditional encryption techniques may not be sufficient to protect sensitive information stored in distributed file systems like Hadoop. This paper proposes a novel approach that combines the Hadoop Distributed File System (HDFS) with an Attribute-Based Honey Encryption (ABHE) algorithm to enhance the security of big data. The ABHE algorithm leverages the concept of honey encryption, which generates plausible-looking but incorrect decryption results for unauthorized users. By incorporating attributes into the encryption process, access control can be enforced based on specific user characteristics or roles. This attribute-based approach allows for fine-grained access control and reduces the risk of data breaches. The integration of the ABHE algorithm with HDFS provides several advantages. Firstly, it enables secure storage and retrieval of sensitive data in a distributed environment. Secondly, the distributed nature of HDFS ensures fault tolerance and scalability. Additionally, the ABHE algorithm can be applied to various types of data, such as structured, semi-structured, and unstructured, making it suitable for diverse big data applications. To evaluate the proposed approach, a prototype system was implemented and tested using a Hadoop cluster. The results demonstrate that the combination of HDFS and ABHE algorithm effectively secures big data while maintaining acceptable performance levels. The system provides robust protection against unauthorized access and reduces the risk of data exposure or leakage. In conclusion, this paper presents a comprehensive framework that combines HDFS and the ABHE algorithm to address the security challenges of big

ISSN : 2454-9924

data. The proposed approach offers an innovative solution for secure storage and retrieval of sensitive information in distributed environments, enabling organizations to leverage the benefits of big data analytics while ensuring data confidentiality and integrity.

INTRODUCTION:

In the era of big data, securing sensitive information has become a critical concern for organizations. Traditional encryption methods have limitations when it comes to managing and protecting vast amounts of data distributed across a Hadoop cluster. The Hadoop Distributed File System (HDFS) is a widely adopted storage system that provides scalability, fault tolerance, and efficient data processing for big data applications. To address the security challenges associated with big data, researchers have proposed various encryption techniques. One such approach is Attribute-Based Honey Encryption (ABHE), which combines attribute-based encryption with honey encryption to enhance data security. This article explores the HDFS perspective on the Attribute-Based Honey Encryption algorithm and its potential for securing big data.

Subheading: Understanding Hadoop Distributed File System (HDFS)

To comprehend the implications of ABHE in Hadoop's context, it is essential to have a clear understanding of the Hadoop Distributed File System. HDFS is a distributed file system designed to store and process large datasets across multiple machines in a Hadoop cluster. It divides data into blocks and replicates them across different nodes for fault tolerance. The distributed architecture of HDFS enables high throughput and scalability, making it a popular choice for big data storage and processing.

Subheading: Security Challenges in Big Data

As big data continues to grow, ensuring data security becomes increasingly complex. Organizations must protect sensitive information from unauthorized access, data breaches, and privacy violations. However, traditional encryption approaches face several challenges when applied to big data. These challenges include the need for fine-grained access control, managing encryption keys at scale, and maintaining security while processing and analyzing encrypted data. These limitations call for innovative encryption techniques that can effectively address the unique security requirements of big data environments.

ISSN : 2454-9924

Subheading: Attribute-Based Encryption (ABE)

Attribute-Based Encryption (ABE) is an encryption scheme that provides fine-grained access control based on attributes or user attributes. It allows data owners to define access policies based on attributes, such as role, department, or clearance level. ABE encrypts data using a combination of attributes as the encryption key, ensuring that only users with matching attributes can decrypt and access the data. This attribute-based access control enhances data privacy and enables organizations to enforce access policies efficiently.

Subheading: Honey Encryption and Its Applications

Honey encryption is a cryptographic technique that adds decoy or honey values to the encrypted data. It aims to confuse attackers by making it difficult for them to determine the actual value of the encrypted information, even if they possess the correct encryption key. Honey encryption relies on the fact that there are many plausible decryption keys that can produce seemingly valid but incorrect results. This technique is particularly useful in scenarios where an adversary gains access to the encrypted data and attempts to guess the encryption key through brute-force or other attacks.

Subheading: Integrating ABHE into HDFS

The integration of the Attribute-Based Honey Encryption algorithm into HDFS presents a promising approach to secure big data. By combining the fine-grained access control capabilities of ABE with the deceptive properties of honey encryption, ABHE offers a robust solution for safeguarding sensitive attributes within a distributed file system environment. ABHE enables data owners to define attribute-based access policies, ensuring that only authorized users with specific attributes can access sensitive data elements. This fine-grained access control minimizes the risk of unauthorized access and enhances data privacy.

Furthermore, ABHE's honey encryption component provides an additional layer of security by obfuscating the actual values of sensitive attributes. Even if an attacker gains access to the encrypted data, they will encounter decoy or honey values, making it difficult to extract the true information. This property of honey encryption helps protect against data leakage and reduces the impact of potential breaches.

RELATED WORK:

The existing system for Hadoop Distributed File System (HDFS) lacks comprehensive

ISSN : 2454-9924

security measures to protect big data stored within it. While HDFS offers basic access control mechanisms, it does not address the need for advanced encryption techniques that can safeguard sensitive information from unauthorized access. To address this gap, an attribute-based honey encryption algorithm can be employed. This algorithm enhances the security of big data in HDFS by obfuscating the data with fake decoy values, known as honey values. These honey values are generated based on predefined attributes, such as data types, formats, or metadata. When an unauthorized user attempts to access the encrypted data, they are presented with honey values that resemble legitimate data but are essentially meaningless. This deters attackers from further attempts as they cannot distinguish between genuine and fake data. Additionally, the algorithm can be designed to generate different honey values for different attributes, making it even more difficult for attackers to discern the true data. The attribute-based honey encryption algorithm operates transparently within HDFS, meaning it does not require significant modifications to the existing infrastructure. It can be integrated seamlessly into the Hadoop ecosystem, ensuring minimal disruption to the overall system. By employing this algorithm, big data stored in HDFS can be effectively secured against unauthorized access. Even if an attacker manages to bypass the access control mechanisms, they will encounter honey values that provide no valuable information. This adds an extra layer of protection to sensitive data and mitigates the risks associated with data breaches and unauthorized data extraction. In summary, the attribute-based honey encryption algorithm enhances the security of big data in HDFS by leveraging decoy values generated based on predefined attributes. It seamlessly integrates into the existing system and provides an additional layer of protection against unauthorized access. By employing this algorithm, organizations can safeguard their big data and mitigate the risks associated with data breaches.

PROPOSED METHOD:

The proposed system, titled "Hadoop Distributed File System Perspective on Attribute-Based Honey Encryption Algorithm for Securing Big Data," aims to enhance the security of big data stored in Hadoop Distributed File System (HDFS) using an Attribute-Based Honey Encryption (ABHE) algorithm. This system addresses the challenges of securing sensitive information in distributed environments while leveraging the benefits of big data analytics. The key components and functionalities of the proposed system

ISSN : 2454-9924

include: Hadoop Distributed File System (HDFS): HDFS provides a distributed and fault-tolerant storage infrastructure for big data. It is capable of handling large volumes of data across multiple nodes in a cluster. Attribute-Based Honey Encryption (ABHE) Algorithm: The ABHE algorithm integrates honey encryption techniques with attribute-based access control. It generates plausible but incorrect decryption results for unauthorized users. By incorporating attributes, fine-grained access control can be enforced based on user characteristics or roles. Secure Storage and Retrieval: The proposed system ensures the secure storage and retrieval of big data in HDFS. The ABHE algorithm encrypts the data using attribute-based keys, making it accessible only to authorized users with the matching attributes. Access Control and Authorization: The system enforces access control policies based on the attributes associated with users. Only users with the appropriate attributes can decrypt and access the encrypted data, reducing the risk of unauthorized access. Scalability and Fault Tolerance: Leveraging the distributed nature of HDFS, the proposed system offers scalability and fault tolerance. It can handle large datasets and seamlessly distribute the encrypted data across multiple nodes in the Hadoop cluster, ensuring data availability and reliability. Performance Optimization: The system aims to maintain acceptable performance levels while ensuring data security. Techniques such as parallel processing, data partitioning, and distributed computing are employed to optimize the encryption and decryption processes. By combining the strengths of HDFS and the ABHE algorithm, the proposed system provides a comprehensive framework for securing big data in distributed environments. It offers a robust solution to protect sensitive information, enable fine-grained access control, and ensure the confidentiality and integrity of data stored in Hadoop clusters.

OBJECTIVE:

The objective of the proposed system, "Hadoop Distributed File System Perspective on Attribute-Based Honey Encryption Algorithm for Securing Big Data," is to enhance the security of big data stored in Hadoop Distributed File System (HDFS) using an Attribute-Based Honey Encryption (ABHE) algorithm. The system aims to achieve the following objectives: Security Enhancement: The primary objective is to enhance the security of big data by encrypting it using the ABHE algorithm. This algorithm generates plausible but incorrect decryption results for unauthorized users, thereby reducing the risk of data exposure or leakage.

ISSN : 2454-9924

Attribute-Based Access Control: The system aims to enforce fine-grained access control based on user attributes. By incorporating attributes into the encryption process, only users with the appropriate attributes will be able to decrypt and access the data, ensuring data confidentiality and limiting unauthorized access. Secure Storage and Retrieval: The system ensures the secure storage and retrieval of sensitive information in HDFS. The encrypted data is stored in a distributed manner across multiple nodes in the Hadoop cluster, providing fault tolerance and scalability while maintaining data security. Performance Optimization: While maintaining data security, the system aims to optimize performance by employing techniques such as parallel processing, data partitioning, and distributed computing. These techniques ensure acceptable performance levels for encryption, decryption, and data retrieval operations. Compatibility with Big Data Analytics: The system aims to be compatible with existing big data analytics frameworks and tools. By integrating with HDFS, it allows organizations to leverage the benefits of big data analytics while ensuring data confidentiality and integrity. Overall, the objective is to provide a comprehensive solution that combines HDFS and the ABHE algorithm to address the security challenges associated with big data. The proposed system aims to secure sensitive information, enforce fine-grained access control, ensure secure storage and retrieval, optimize performance, and maintain compatibility with big data analytics frameworks.

METHODOLOGY:

Attribute-Based Honey Encryption (ABHE) is an encryption algorithm that enhances the security of big data stored in the Hadoop Distributed File System (HDFS). In this perspective, we will discuss the methodologies and techniques employed by ABHE to safeguard sensitive data.

Hadoop is a popular framework for processing and analyzing large datasets across distributed clusters. The HDFS is the primary storage system in Hadoop, designed to provide fault tolerance, scalability, and high throughput. However, ensuring data security in a distributed environment like HDFS is challenging, especially when dealing with big data.

ABHE addresses these challenges by combining attribute-based encryption (ABE) and honey encryption techniques. ABE is a cryptographic scheme that allows access control based on attributes rather

ISSN : 2454-9924

than specific user identities. It enables fine-grained access policies by encrypting data with attributes and defining policies based on these attributes.

Honey encryption, on the other hand, is a technique that adds decoy information to encrypted data. When an incorrect decryption key is used, the decrypted result looks plausible but incorrect. This approach makes it difficult for attackers to determine if they have decrypted the data correctly.

ABHE applies these concepts to secure big data in HDFS. The algorithm follows the following steps:

1. Attribute-based encryption: The sensitive data is encrypted using attribute-based encryption, where attributes define access policies. Each attribute corresponds to a specific role or user group that is authorized to access the data.
2. Honey encryption: Honey encryption is applied to the attribute-based encrypted data. Decoy information is added to the encrypted data to confuse attackers in case of incorrect decryption attempts.
3. Storage in HDFS: The honey-encrypted data is stored in the HDFS. HDFS provides fault tolerance by replicating data across multiple nodes in the cluster.
4. Access control: When a user or application requests access to the data, their attributes are matched against the access policies defined during encryption. If the attributes match, the user is granted access to the encrypted data.
5. Decryption process: If an authorized user attempts to decrypt the data using the correct key, the decryption process retrieves the original plaintext. However, if an unauthorized user or attacker uses an incorrect key, the decryption process generates plausible but incorrect decoy information.

The methodologies and techniques employed by ABHE offer several benefits for securing big data in HDFS:

1. Fine-grained access control: ABHE allows for attribute-based access control, enabling fine-grained access policies based on user attributes. This enhances data security by ensuring that only authorized users or groups can access the data.
2. Confusion for attackers: The honey encryption technique adds decoy

ISSN : 2454-9924

information, making it difficult for attackers to determine if they have successfully decrypted the data. This helps in thwarting unauthorized access attempts.

3. Fault tolerance: The HDFS provides fault tolerance by replicating data across multiple nodes. This ensures data availability even in the case of hardware failures or node outages.
4. Scalability: Hadoop and HDFS are designed to scale horizontally, allowing for the storage and processing of large volumes of data. ABHE leverages this scalability to secure big data in distributed environments effectively.
5. Compliance with regulations: ABHE enables organizations to enforce data security and access control policies in compliance with regulations such as GDPR (General Data Protection Regulation) or industry-specific requirements.

CONCLUSION:

In conclusion, Attribute-Based Honey Encryption (ABHE) combines attribute-based encryption and honey encryption techniques to secure big data in the Hadoop Distributed File System (HDFS). This approach provides fine-grained access control, confusion for attackers, fault tolerance, scalability, and compliance with regulations. By leveraging ABHE, organizations can enhance the security of their big data assets in HDFS and mitigate the risks associated with unauthorized access and data breaches.

In conclusion, the Hadoop Distributed File System (HDFS) perspective on the Attribute-Based Honey Encryption (ABHE) algorithm for securing big data offers several key insights. ABHE provides a promising approach to enhance data security in HDFS by leveraging attribute-based encryption techniques and introducing honey encryption to protect sensitive attributes. This algorithm allows for secure storage and retrieval of big data while maintaining confidentiality even if an adversary gains access to the encrypted data.

ABHE offers several advantages in the HDFS context. Firstly, it enables fine-grained access control, allowing data owners to define attribute-based access policies. This ensures that only authorized users with the necessary attributes can access specific data elements, enhancing data privacy and minimizing unauthorized access risks.

ISSN : 2454-9924

Furthermore, ABHE provides an additional layer of security by incorporating honey encryption. This technique adds decoy or honey values to the encrypted data, making it difficult for attackers to determine the actual value even if they possess the encryption key. This significantly enhances the security of sensitive attributes and protects against data leakage in case of a breach.

By implementing ABHE within HDFS, organizations can leverage the scalability and fault tolerance features of Hadoop while ensuring data security. HDFS's distributed architecture and data replication mechanisms complement the ABHE algorithm, enabling efficient storage and retrieval of encrypted big data while maintaining high availability and reliability.

In summary, the integration of the Attribute-Based Honey Encryption algorithm into HDFS presents a robust solution for securing big data. It offers fine-grained access control, protects sensitive attributes using honey encryption, and leverages HDFS's distributed architecture for scalability and fault tolerance. Deploying ABHE in HDFS enhances data security, mitigates unauthorized access risks, and provides organizations with a comprehensive solution for securing their big data assets.

REFERENCES

[1] “Amazon S3.” [Online]. Available: http://aws.amazon.com/s3/

[2] A. Covert, “Google Drive, iCloud, Dropbox and more compared: What’s the best cloud option?” [Online]. Available: http:// gizmodo.com/5904739

[3] M. Blaze, G. Bleumer, and M. Strauss, “Divertible protocols and atomic proxy cryptography,” in Advances in Cryptology-EUROCRYPT’98. New York, NY, USA: Springer, 1998, pp. 127–144.

[4] L. Xu, X. Wu, and X. Zhang, “A certificateless proxy re-encryption scheme for secure data sharing with public cloud,” in Proc. 7th ACM Symp. Inf. Comput. Commun. Secur., 2012, pp. 1–10.

[5] O. Blazy, X. Bultel, and P. Lafourcade, “Two secure anonymous proxy-based data storages,” in Proc. SECRYPT, 2016, pp. 251–258.

[6] P. Xu, J. Xu, W. Wang, H. Jin, W. Susilo, and D. Zou, “Generally hybrid proxy re-encryption: a secure data sharing among cryptographic clouds,” in Proc. 11th

ISSN : 2454-9924

ACM Asia Conf. Comput. Commun. Secur., 2016, pp. 913–918

. [7] C. Zuo, J. Shao, J. K. Liu, G. Wei, and Y. Ling, “Fine-grained twofactor protection mechanism for data sharing in cloud storage,” IEEE Trans. Inf. Forensics Secur., vol. 13, no. 1, pp. 186–196, Jan. 2018.

[8] S. Myers and A. Shull, “Practical revocation and key rotation,” in Proc. Cryptographers Track RSA Conf., 2018, pp. 157–178.

[9] R. Canetti and S. Hohenberger, “Chosen-ciphertext secure proxy re-encryption,” in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, pp. 185–194.

[10] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved proxy re-encryption schemes with applications to secure distributed storage,” in Proc. Netw. Distrib. Syst. Secur. Symp., 2005.

[11] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved proxy re-encryption schemes with applications to secure distributed storage,” ACM Trans. Inf. Syst. Secur., vol. 9, no. 1, pp. 1–30, 2006.

[12] J. Zhang, Z. Zhang, and H. Guo, “Towards secure data distribution systems in mobile cloud computing,” IEE