2024, 21(5 Special Issue): 1798-1810
https://doi.org/10.21123/bsj.2024.10550
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

A Column Encryption-Based Privacy-Preserving Framework for Hadoop Big Data Sets

Hidayath Ali Baig

Department of Information Technology, College of Computing and Information Sciences, University of Technology and Applied Sciences, Sur, Oman.
ICCDA2023: International Conference on Computing and Data Analytics 2023.

Received 28/12/2023, Revised 03/05/2024, Accepted 05/05/2024, Published 25/05/2024

Abstract

The exponential growth of the Internet, the Internet of Things, and Cloud Computing in recent times has led to a significant rise of data across various sectors of business and industry. Big data has become a growing trend in recent years, attracting the attention of academics, corporate leaders, and government officials worldwide. Hadoop is a commonly adopted framework for processing big data. This data expansion has the potential to provide substantial and beneficial advantages, and some early success has been achieved from a technical standpoint in dealing with such a large quantity of data. Along with its many benefits, it also has a slew of disadvantages. These include, but are not limited to, data storage, exchange, curation, transit, analysis, visualization, security and privacy. In this research, the privacy implications of Big Data analytics are being investigated. Several publications suggest methods to secure big data. Each technique has advantages and disadvantages. Regardless of privacy laws, application developers must protect sensitive data. Therefore, there is need for innovative methods to guarantee the protection of individuals' privacy in the context of big data. This paper presents a framework for preserving privacy in data-at-rest within the Hadoop architecture. The framework employs columnar data storage, data masking, and encryption techniques to address these challenges efficiently.

Keywords: Big Data, Columnar Storage, Data Analytics, Encryption, Hadoop, HDFS, Privacy.

Introduction

Overview of Big Data

In modern society, which has been characterized by a growing reliance on technology, there has been a noticeable rise in the volume of data produced by embedded devices such as smartphones, automobiles, and data from various sensors. The rapid growth of individual interactions with the Internet has resulted in a significant increase in data volume, primarily driven by the utilization of e-commerce platforms, e-governance systems, and different social media networks. The term "Big Data" refers to the enormous volume of data. Data growth reached Zeta or Exabyte. According to the IDC (International Data Corporation) Reinsel et al.[1] document entitled "Data Age 2025", there will be ten times more global data by 2025. It foresees a time when the generation, utilization, and administration of "important" data will become increasingly crucial for the smooth running of daily life. This growth in data creates the "Big Data" phenomenon. "Big data

2024, 21(5 Special Issue): 1798-1810
https://doi.org/10.21123/bsj.2024.10550
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

is a broad term for data sets that are so large or complex that traditional data processing applications are insufficient"[2]. This data growth brings significant and valuable benefits, and some initial success is achieved from a technological perspective to handle such enormous data.

Different organizations, such as hospitals, businesses, e-commerce, stores and supply chains, etc., use digital technology to generate vast amounts of data. People and computers provide data through closed-circuit TV streaming, website recordings, etc. Social networks and smartphones create tons of data every second.

This global trend also increased awareness of data processing, interpretation and simulation for companies trying to optimize the value of accessible information resources. From making better business decisions to enhancing results, it delivers accurate results from industry to industry.

Big data is generally heterogeneous, meaning that all objects are multimodal in big data. This involves various interconnected things, including audio files, documents and images. It results in a high heterogeneity because of the structured and unstructured data. It contains individual data and is dependent on each other at the same time.

Big Data Tools and Technology

There is an increase in the number of tools and technologies to capture, manipulate, store, analyze and aggregate big data. This section describes the tools used in big data from a technological point of view.

Apache Hadoop ["High-availability distributed object-oriented platform"]: It is an open-source software platform that makes it easy to distribute massive data sets utilizing basic programming models through clusters of computers. Several ideas for server sizes range from a single server to hundreds of computers, each of which provides local processing and storage. The library is built to identify and manage application layer failures, providing highly open service across a cluster of computers rather than focusing on high-availability hardware. These may be vulnerable to failures (Apache Software Foundation)[3]. Fig. 1 below depicts the architecture of Apache Hadoop Ecosystem.

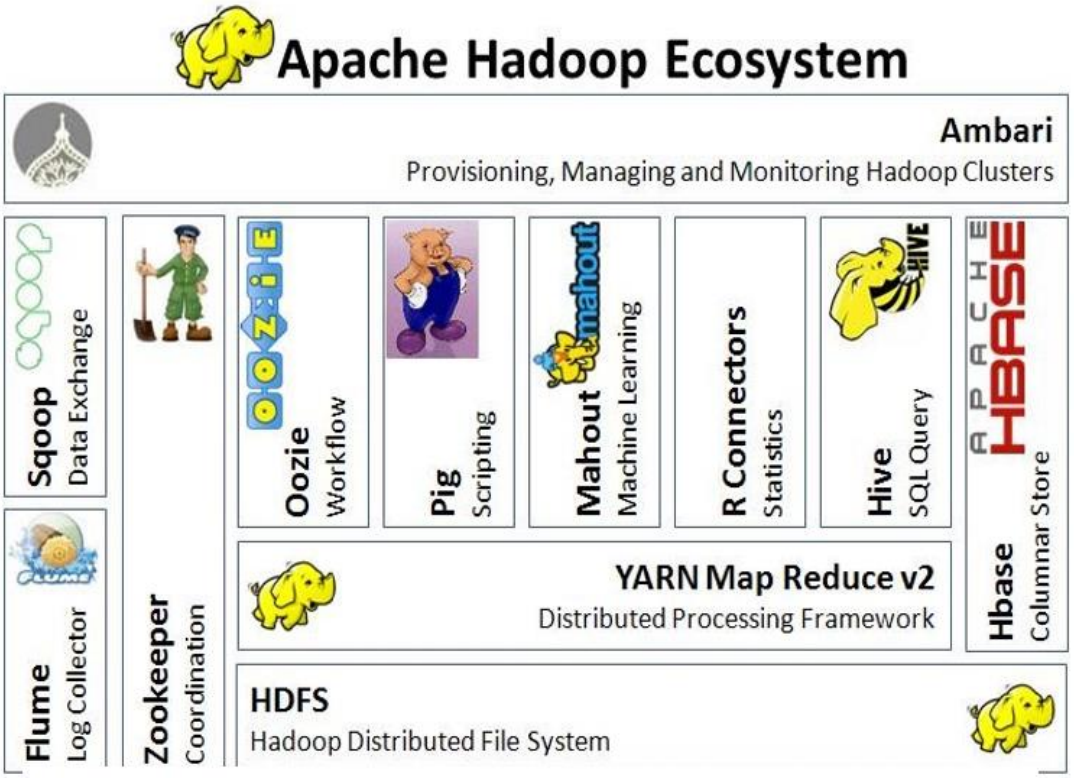

Figure 1. Hadoop Big Data Architecture

2024, 21(5 Special Issue): 1798-1810
https://doi.org/10.21123/bsj.2024.10550
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

Big Data Analytics

Big data processing uses sophisticated tools and methods to capture, process and optimize vast volumes of data. It is part of a broader approach used by organizations in many fields and uses the information available to make specific decisions and accomplish particular objectives. The objective behind this principle is that a lot of data can be helpful if it is unlocked with the right resources[4].

Various areas of business and industry have benefited from big data analytics technology, which is becoming increasingly prevalent. These areas produce an immense volume of data, requiring a large-scale collection method for reliable and successful decision-making. These fields of application are healthcare, telecommunications, network optimization, travel forecasts, retail trading, the finance industry, and energy use. The term "big data" refers to extensive data sets that contain a greater degree of heterogeneity and complex structuring[4]. These attributes typically exhibit a positive correlation with increased challenges in the areas of data storage, analysis, practical application, and result extraction. The term "big data analytics" refers to the process of conducting research on vast amounts of complex data in order to uncover hidden patterns or detect invisible correlations. However, a noticeable problem arises when considering the relationship between the security and privacy of large-scale datasets and the extensive utilization of such datasets.

Big data analysis is based on various techniques and strategies unique to the actual computing framework and experience. Predictive analysis, which is meant to forecast the behaviour of a specific client or object, indicates the features that can be contained in a particular user approach. Data mining, where data sets are analyzed to detect trends or patterns, is one of the most effective techniques to turn data into usable information, better known as intelligence. Fig. 2 below list the important concepts in Big data analytics.

Figure 2. Five Important Things in Big Data Analytics

Privacy and Security in Data Analytics

Availability, integrity and confidentiality of information are the security concerns of big data. It must be guaranteed that the data is protected from unauthorized exposure and that the information is consistent, precise, and available whenever possible[5]. On the contrary, the preservation of privacy through data utilization is a highly beneficial task. The utilization of the provided information is limited to the stakeholders for whom it was initially intended. For instance, if a person purchases a product from the ABC Company and then includes their personal data, such as location, card number, etc. The organization cannot offer this information to a third party.

The sole desire of entrusting this data to companies is to implement the digital protection strategy to protect the consumer's data privacy and personally recognizable details. No data protection strategy or policy will guarantee that the data will not be sold by the companies which they are entrusted with[5,6].

The Challenges of Privacy and Security in Big Data

Security and privacy problems are magnified by complexity, diversity, big data volume, large-scale cloud infrastructure, data source and format variability, decentralized data storage presence, and high-volume inter-domain migration. Wide-scale cloud infrastructure use, distributed data through large data networks with various computing applications, further strengthens the attacks on the framework [7].

2024, 21(5 Special Issue): 1798-1810
https://doi.org/10.21123/bsj.2024.10550
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

Concerns have been raised among a large number of individuals regarding how best to safeguard the privacy and safety of individuals in light of the advent of big data. The process of putting in place protocols that regulate the collection and utilization of massive amounts of personal data is referred to as the concept of information privacy. The primary concern with regard to privacy is the authority and control that can be exercised over the dissemination of personal information about individuals through the medium of the Internet. This is due to the inherent characteristics of the Internet.

Literature Review

Hadoop and HDFS Environment

Apache Hadoop is the most commonly accepted and regularly implemented open-source software platform for distributed storage and analysis of large-scale cluster node data. Hadoop has three key components: i) the Hadoop Distributed File System (HDFS), the general feature that allows distributed disk systems that are scalable and fault-resistant; ii) the Hadoop MapReduce framework; and iii) the Hadoop Common modules[8].

A Hadoop cluster comprises a master node and a variety of slave nodes (running a job tracker and NameNode) (running a task tracker and data node on each slave node). Fig. 3 below depicts a single node along with two slave nodes.

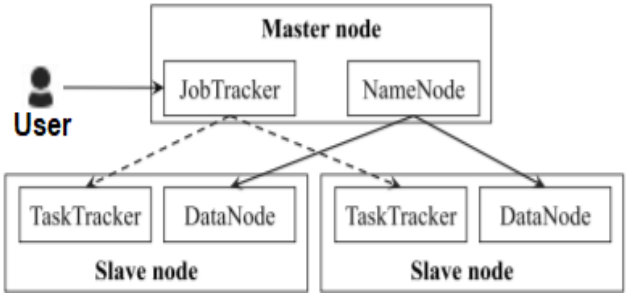

Figure 3. Hadoop Cluster Configuration of two slave nodes on a master node.

As illustrated in Fig. 3. above, an execution environment is provided for MapReduce jobs by Job Tracker and Task Trackers. A distributed file system named the Hadoop distributed file system (HDFS) includes Name Nodes and Data Nodes. It facilitates reading, writing, removing file operations, and creating and deleting directory operations. The NameNode processes the cluster metadata and data nodes, which store the data. NameNode uses inodes to carry details regarding files and folders. Inodes store many attributes for files and folders, including passwords, changes and access times, and disc space allocation. Data is broken down into small HDFS blocks (the most commonly used sizes are 64MB and 128MB). Each block is replicated separately in several data nodes, and the mapper and reducer process the block's copy[9]. The block copy is added to each data node, which uses two files to reflect each block copy in the local host file system. The first file contains the file data, and the second file contains the metadata block.

The Hadoop architecture was initially created without incorporating security measures, as its primary use cases revolved around the management of vast quantities of publicly available web data. During this development phase, considerations regarding data confidentiality and complex internal rights management were not taken into account. The original concept of Hadoop assumes that clusters exist within a secure environment, consisting of reliable and cooperative computers that authorized users exclusively access. The prominence of Hadoop security issues increases as the amount of data, consumers, and software applications accessing big-data platform clusters increases and as more businesses store personal data on the Hadoop cluster. The Yahoo team released a paper[10] in which they selected Kerberos as the primary authentication method for the Hadoop platform. This decision was made to establish a robust security management scheme for the Hadoop big-data environment. Alongside various companies, numerous network security researchers, both domestically and internationally, are engaged in researching Hadoop security. These studies encompass a range of methodologies, including the exploration of reliable equalization techniques[11–13], the development of mixed encryption algorithms tailored for Hadoop environments[14], the implementation of triple data encryption algorithms[15], the proposition of parallel encryption strategies[16], and the introduction of other related schemes. Hadoop encompasses a set of five

2024, 21(5 Special Issue): 1798-1810
https://doi.org/10.21123/bsj.2024.10550
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

security projects, namely Apache Sentry[17], Apache RecordService[18], Apache Ranger[19], Apache Knox Gateway[20], and Apache Rhino[21]. Apache Sentry is an open-source component of Hadoop that has been developed and released by Cloudera. It offers a range of features, including granularity, role-based authorization, and a multitenant management model.

On the other hand, RecordService is a recently introduced security layer for Hadoop. Its primary purpose is to facilitate secure access to data and analysis engines that are operating on Hadoop. It is worth noting that the security components of Sentry, which Cloudera previously provided, continue to support access to the control rights that have been defined. The purpose of RecordService is to enhance the functionality of Sentry by providing additional control over access at the row and column levels. Apache Ranger is an open-source component for Hadoop developed by Hortonworks. It addresses the existing issue of decentralized security management within the Hadoop platform by establishing a central and unified management interface. This interface facilitates access management, log auditing, and other related functionalities for all services on the platform. The Apache Knox Gateway is a management security scheme developed by Hortonworks that is designed to operate at the boundaries of a cluster.

The maintenance and operation team should solely focus on the internal cluster without the necessity of exposing any deployment specifications. Central access to Hadoop-related operations through boundary network administration significantly reduces the development complexity for the development team. Apache Rhino, an open-source project spearheaded by Intel, is committed to enhancing the security of Hadoop's ecosystem components and data. Encryption is widely recognized as the most direct and efficient method for safeguarding data, serving as the ultimate line of defense in data protection. The Apache Rhino project presents a solution for enhancing the security of Hadoop data through the implementation of HDFS transparent encryption technology. Transparent encryption refers to a mechanism wherein the process of encrypting and decrypting data is seamlessly integrated into users' routine operations without requiring any additional effort or disruption to their established habits. The implementation of this technology ensures that the system automatically encrypts data upon users' data-saving actions.

Upon the initiation of user activity involving the opening or modification of designated files, the system will undertake the automatic decryption of the encrypted files. The encrypted data remains within the hard disk, while the unencrypted text is stored in the computer's memory. Upon exiting the environment, the data becomes inaccessible for automatic decryption, thereby ensuring its protection. The implementation of content-level encryption in Hadoop Distributed File System (HDFS) has given rise to the notion of Encryption Zones. The encrypted area referred to in this context is a distinct directory within the Hadoop Distributed File System (HDFS). Within this directory, the written content undergoes encryption in a manner that is invisible to the user.

Similarly, when accessing the contents within this encrypted area, the decryption process occurs seamlessly and without user intervention. The process of encryption and decryption does not necessitate any modifications to the code of the user's application. The encryption employed in this context is characterized by end-to-end functionality, wherein the Client possesses sole authority to both encrypt and decrypt the data. Foreign data platforms, such as Cloudera and IBM, have begun implementing encryption technology. However, there is a need for further enhancement of the HDFS transparent encryption technology. The following table illustrates different mechanisms adopted in Hadoop for encryption.

2024, 21(5 Special Issue): 1798-1810
https://doi.org/10.21123/bsj.2024.10550
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

Table1. Encryption of Data in Hadoop (Hortonworks Inc)

Volume Encryption	Application-level Encryption	HDFS data-at-rest Encryption
Protects data in the event of a hard drive failure or instances of physical theft.	Protects data in the event of a hard drive failure or instances of physical theft.	Protects data in the event of a hard drive failure or instances of physical theft.
The entire volume is encrypted: very coarse-grained security	Supports a higher level of granularity and prevents "rough admin" access	Uses specially designated HDFS directories known as "Encrypted Zones."
It does not protect against viruses or other malicious software assaults while the system operates.	Complicates the application architecture by adding a layer of complexity	Encryption of data read from and written to HDFS from end to end. HDFS is not required to contain unencrypted data or keys.

Also, with coarse grain encryption, Additional issues include data stored in database table rows next to each other that has to be protected differently. E.g., general customer data and customer identities are kept together. It is essential to secure information such as customer name and identification number. Therefore, there is a need to protect them differently and address the following issues:

- Authorization: Which user can access and view the information?
- Audit: Monitor the individuals who have accessed the document.
- Encrypt it on disc: problems of enforcement.

One of the primary concerns associated with this technique is the insufficiency of file-level granularity in HDFS to ensure security. Additionally, it is worth noting that the level of granularity at the directory level is considerably coarser in comparison.

Separate Tables for Sensitive Data

Organizations adopt this strategy to isolate sensitive information by storing it in separate tables. This methodology enables the utilization of Hadoop Distributed File System (HDFS) permissions to safeguard sensitive information. However, a limitation of this strategy is in the management of these tables. The user must perform joins to retrieve data from many tables. In the context of big data, the table may include a vast amount of information, often measured in terabytes. As a result, the execution of queries may see a performance decline while attempting to combine and extract data from these vast tables.

Problem Statement

Despite significant advancements in big data analytics, numerous challenges remain in protecting the data privacy of consumers or user details. Developers must be able to authenticate their applications' compliance with privacy agreements, ensuring that no confidential data is compromised due to alterations in the application or privacy regulations. When presented with vast amounts of data, it becomes imperative to alter data access policies, as the data proprietor may be compelled to disseminate data to many organizations. According to Yang et al.[14], it is argued that the implementation of attribute-based access control systems does not facilitate protocol modifications. The administration of policies in attribute-based access control systems is a labor-intensive and complicated task. This assertion is valid as outsourcing data to the cloud is employed instead of maintaining local storage on the controller's computer. To ensure data security, it is imperative to re-encrypt the data and store it on the cloud server. Should the data owner desire to modify the regulations governing the data, transmitting the data back to the local network becomes necessary. The current system exhibits a substantial communication overhead and a significant computational cost.

Identity-Based Encryption (IBE) and Attribute-Based Encryption (ABE) cannot allow ciphertext

2024, 21(5 Special Issue): 1798-1810
https://doi.org/10.21123/bsj.2024.10550
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

receiver alterations. Encrypting a ciphertext using a different cipher can change its recipient. This method decrypts and re-encrypts data. Decrypting and re-encrypting enormous amounts of data is time-consuming and expensive.

Access controls for vast amounts of data must be restricted to protect privacy. Implementing time and geographically-constrained access control mechanisms is another challenge. Scattered data may struggle with access control methods, increasing query complexity and data availability[22].

In order to address these challenges, it is essential to develop novel research methodologies aimed at safeguarding privacy and advancing standardized approaches.

Given the security implications pertaining to the Hadoop Distributed File System (HDFS) and its management of table-row data, there is a pressing need for a storage solution capable of efficiently accommodating growing datasets while ensuring the preservation of data privacy. The following section highlights the use of different storage formats and proposes effective strategies for safeguarding user data privacy.

Proposed Methodology for Data-at-rest Protection

To effectively achieve the objective of privacy-preserving data storage, this study presents the following recommendations:

- Proposes a novel methodology for enhancing data privacy and security in the context of big data by implementing a column-oriented storage format.
- Presents a comprehensive approach for loading data onto the Hadoop environment.
- Includes the recommendation of an encryption methodology for column-based storage.
- Set up a system for managing encryption keys, and design a flow mechanism for the encryption process. The recommended methodology can be observed in Fig. 4 below.

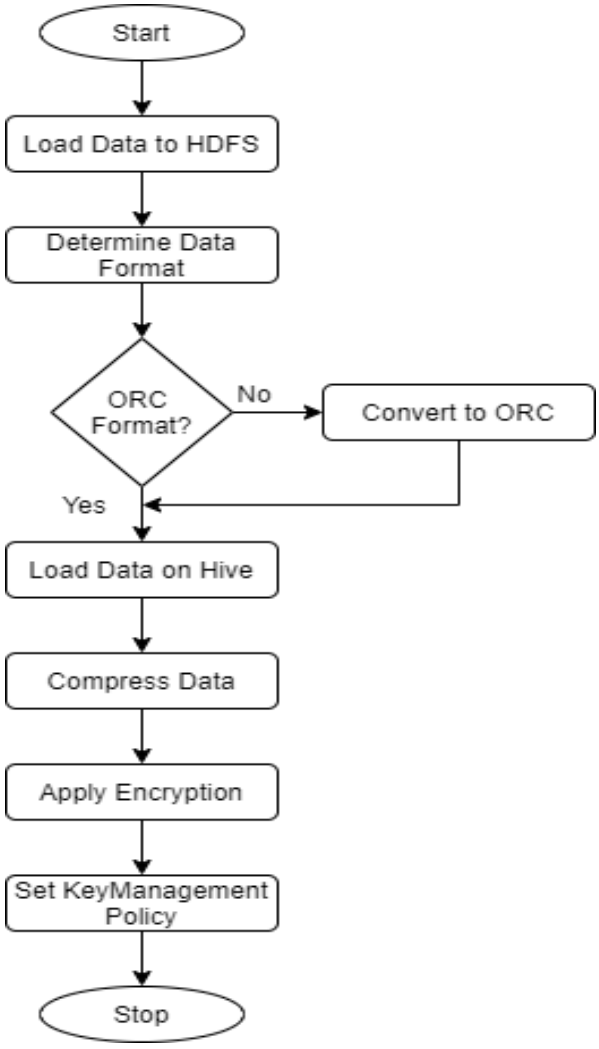

Figure 4. Proposed Data storage pipeline

2024, 21(5 Special Issue): 1798-1810
https://doi.org/10.21123/bsj.2024.10550
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

Column-Oriented Data Format

In recent times, there has been a notable increase in the adoption of column-oriented data formats, transcending the prevalence of traditional data storage formats, which store data in a structure that is organized by columns. The difference between row vs columnar storage can be clearly observed in Fig. 5 below.

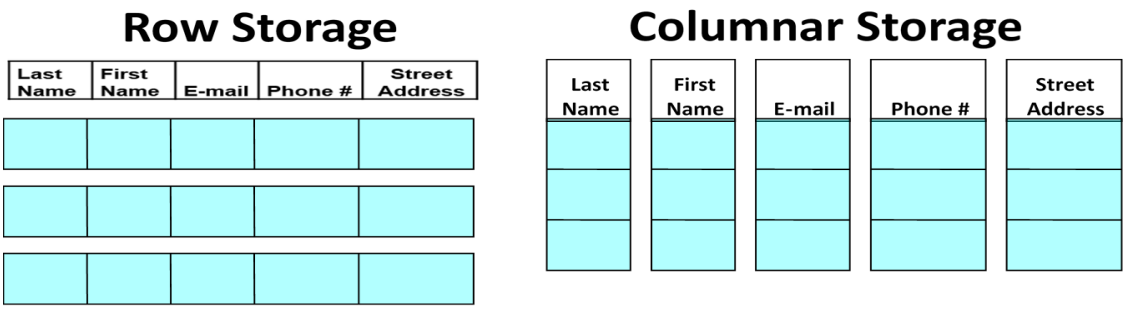

Figure 5. Row vs Columnar Storge

Optimized Row Columnar (ORC) File Format

This section describes the file formats utilized within the Hadoop environment. File formats are considered to be highly significant for data storage. Numerous research efforts have been conducted relating to this topic. Within the Hadoop context, there exists an extensive range of scholarly publications related to different file formats. The RCFile is the main file format used for column data storage.

ORC File Structure

The row data group of the ORC file contains auxiliary meta-information referred to as "rows" and "footer." The compressed footer size and compression parameters are included in a postscript appended to the file. The stripe size, by default, is 250 MB. Expanded stripe sizes enable beneficial sizable HDFS reads. The picture of the file displays a sequence of stripes, along with the data format for each column and the number of rows corresponding to each strip. The count, minimum, maximum, and total of column-level aggregates are also included. The diagram in Fig. 6. illustrates the structure of ORC files.

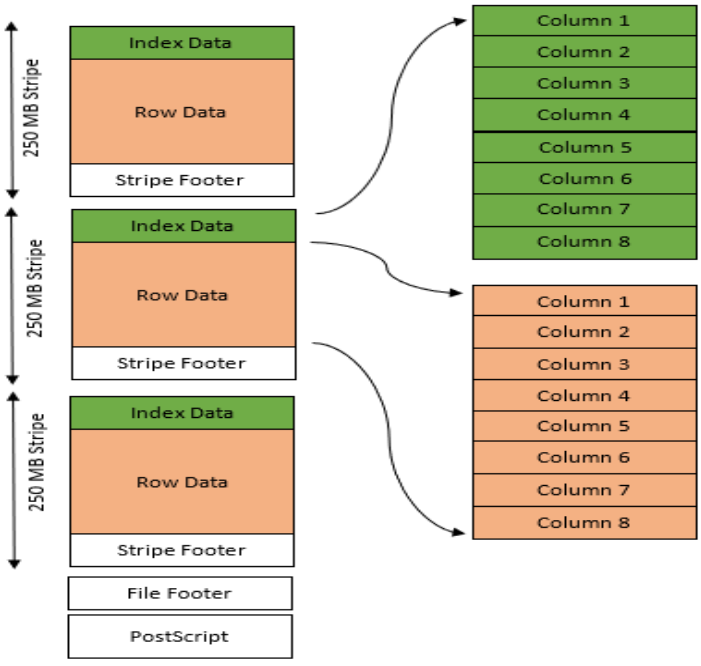

Figure 6. ORC File Structure[23]

2024, 21(5 Special Issue): 1798-1810
https://doi.org/10.21123/bsj.2024.10550
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

The benefits of columnar format over row-oriented formats are given below:

- The process of searching for a subset of columns is found to be effective. In contrast to traditional row-oriented storage, when a query is executed, only a limited number of columns from a database are accessed. It is not necessary to get the entire record. This method has the potential to reduce input/output (I/O) operations substantially.
- The data is organized into columns consisting of similar elements. These algorithms utilize data type knowledge and homogeneity to enhance performance.

ORC Data Loading Strategy

It is recommended to convert data from text format in HDFS to Hive-managed format through the construction of an external Hive table.

Using the LOAD DATA command, the files must be transferred into the hive data files. Creating a transient table that is stored as text, loading data into it, and then copying the information to the ORC table is a potential workaround.

Creating a text file:

```
CREATE TABLE test_details_txt( visit_id INT, store_id SMALLINT) STORED AS TEXTFILE;
CREATE TABLE test_details_orc( visit_id INT, store_id SMALLINT) STORED AS ORC;
```

Loading data into a text table:

```
LOAD DATA LOCAL INPATH '/home/user/test_details.txt' INTO TABLE test_details_txt;
```

Copy to ORC table:

```
INSERT INTO TABLE test_details_orc SELECT * FROM test_details_txt;
```

Load data from a CSV file into the hive ORC table.

It is required to have a comma-separated file to build an ORC formatted table in hive. Fig. 7 below demonstrates the process of Data Loading Strategy.

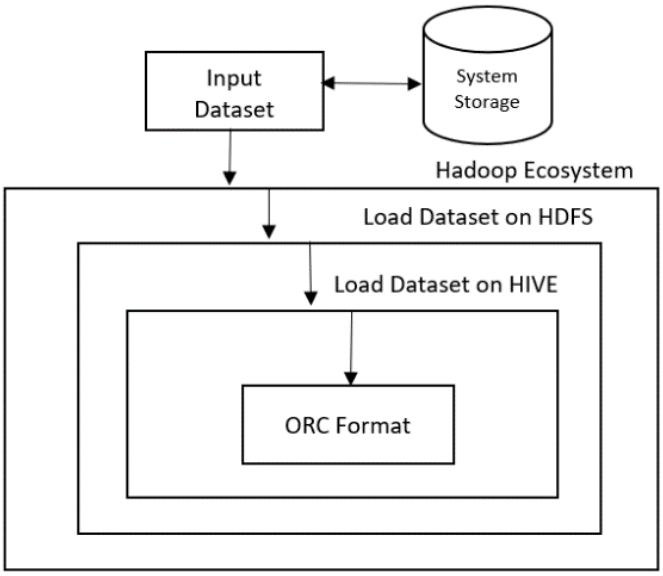

Figure 7. Data Loading Strategy[24]

2024, 21(5 Special Issue): 1798-1810
https://doi.org/10.21123/bsj.2024.10550
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

Key Management

A distinct local key is generated for each encrypted column in the provided file using a random process. Although the reader engages in local decryption, the adversarial user's access to the file is limited solely to the specific column for which they possess authorization. Although the reader engages in local decryption, the adversarial user's access to the file is limited solely to the specific column for which they possess authorization. It is essential for organizations to adopt keystores, such as Navigator Key Trustees, that adhere to the requirements of the Hadoop Distributed File System (HDFS). Data encryption at the local level is performed by either the Hadoop or Ranger platforms, which function as servers responsible for encrypting data. The file footer contains the section labelled "strip details", which contains the encrypted local keys.

The ORC can generate a locally encrypted key using random encryption using either Hadoop or Rangers KMS. For 128-bit AES, the key size is 16 bytes, while for 256-bit AES, the key size is 32 bytes. The local address can be decrypted using AES/CTR, with the first 16 bytes serving as the initialization vector (IV). Fig. 8, below depicts they key management strategy.

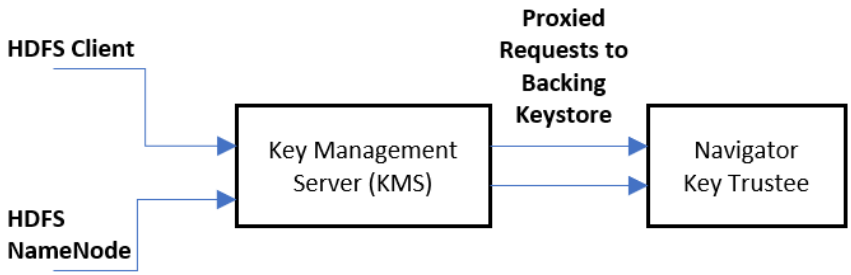

Figure 8. The high-level design of Key Management Strategy [25]

Encryption Flow

The user must access the ORC file to get the encrypted key. If the user has authorization and can access the encrypted work file data, the key management server receives the decrypted key. Data does not flow through the Key Management System (KMS). However, the user cannot see the master key used to encrypt the local key. The local key is generated randomly by the user for each file, while the initialization vector of the file remains constant. Therefore, it is possible to assign the initialization vector as the column of the form in the stripe counter. Fig. 9, blow illustrates the encryption data flow using KMS.

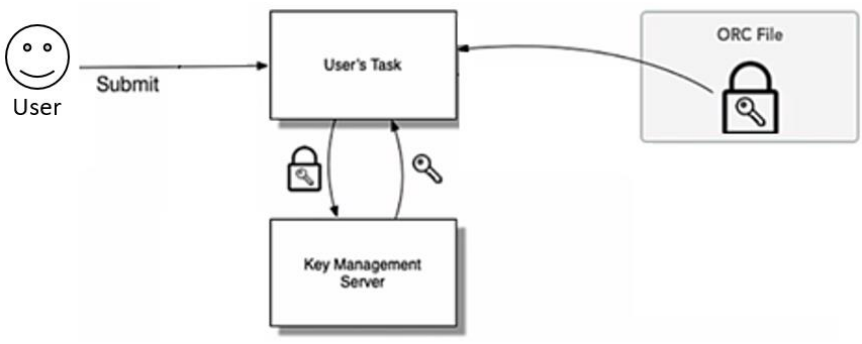

Figure 9. Encryption Data Flow

Key Disposal

In order to facilitate the preservation of data over a specified duration, it is necessary to implement anonymization techniques. One potential strategy for addressing this specific concern involves the adoption of data encryption as an alternative to the practice of creating duplicate copies or rewriting data within a specified timeframe. The keys may be subject to a daily rotation process, followed by a reversion to their initial positions at the end of the

2024, 21(5 Special Issue): 1798-1810
https://doi.org/10.21123/bsj.2024.10550
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

designated period. The preceding encryption key must be removed following the deletion of the data. Upon the removal of the keys, the data undergoes a state of inaccessibility, thereby imposing limitations on users' capacity to retrieve the local keys and efficiently gain entry to the data.

Results and Discussion

After a deeper investigation of privacy and security mechanisms in the Hadoop framework, it becomes evident that encryption methods adopted for big data environments limit data accessibility. In contrast, it has been demonstrated that access control measures are insufficient in effectively safeguarding the confidentiality of information. Therefore, this research suggested an approach for data protection using Optimized Row-Column (ORC) file formats in HDFS and defined a strategy for loading data in Hive (ORC) format. In order to achieve a significant performance boost, established ORC encryption, key management and data masking strategies.

Conclusion

The purpose of this study was to investigate privacy-preserving big data analytics by performing an in-depth analysis of the various privacy-preserving method paradigms. The primary contribution of this research is an in-depth and systematic analysis of contemporary methods for the protection of personal information, taking into account the advantages and disadvantages of these methods, which both academic and commercial organizations have acknowledged. Hadoop's privacy and security regulations indicate that big data encryption restricts the accessibility of data. Studies indicate that restricting access alone is inadequate for ensuring the confidentiality of information. This study proposed the utilization of HDFS Optimized Row-Column (ORC) file formats for data security, specifically in conjunction with Hive data loading. Performance is enhanced through the utilization of ORC encryption, key management, and data masking. This approach guarantees that only authorized users with appropriate keys can retrieve data in its original form. Further research is required to investigate the aspects of privacy and security in real-time Big Data. To effectively address these difficulties, it is essential to implement standardized encryption, data masking, and key management techniques.

Authors' Declaration

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Furthermore, any Figures and images, that are not ours, have been included with the necessary permission for re-publication, which is attached to the manuscript.
- No animal studies are present in the manuscript.
- No human studies are present in the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee at University of Technology and Applied Sciences, Oman.

Authors' Contribution Statement

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

1. Reinsel J, Rydning DR, Gantz J. Gantz JF, Reinsel D, Rydning J. The us datasphere: Consumers flocking to cloud. White Paper. International Data Corporation (IDC) 2019 Jan.
2. Anna K, Nikolay K. Survey on Big Data Analytics in Public Sector of Russian Federation. Procedia Comput Sci. 2015; 55: 905–11. https://doi.org/10.1016/j.procs.2015.07.144

3. Apache Software Foundation. Hadoop. 2020. hadoop.apache.org
4. Hashem IAT, Yaqoob I, Anuar NB, Mokhtar S, Gani A, Ullah Khan S. The rise of "big data" on cloud computing: Review and open research issues. Inf Syst. 2015 Jan; 47: 98–115. https://doi.org/10.1016/j.is.2014.07.006
5. Mutasher WG, Aljuboori AF. New and Existing Approaches Reviewing of Big Data Analysis with Hadoop Tools. Baghdad Sci J. 2022 ;19(4): 887-898. https://doi.org/10.21123/bsj.2022.19.4.0887
6. Jain P, Gyanchandani M, Khare N. Big Data Security and Privacy: New Proposed Model of Big Data with Secured MR Layer. Advanced Computing and Systems for Security. Singapore: Springer Singapore. 2019; 31–53. https://doi.org/10.1007/978-981-13-3702-4_3
7. Mayyahi MA, Seno SA. A Security and Privacy Aware Computing Approach on Data Sharing in Cloud Environment. Baghdad Sci J. 2022; 19(6(Suppl.): 1572. https://doi.org/10.21123/bsj.2022.7077
8. Merceedi KJ, Sabry NA. A Comprehensive Survey for Hadoop Distributed File System. Asian J Res Comput Sci. 2021; 46–57. https://doi.org/10.9734/ajrcos/2021/v11i230260
9. Elkawkagy M, Elbeh H. High Performance Hadoop Distributed File System: Int J Networked Distrib Comput. 2020; 8(3): 119-123. https://doi.org/10.2991/ijndc.k.200515.007
10. Tabrizchi H, Kuchaki Rafsanjani M. A survey on security challenges in cloud computing: issues, threats, and solutions. J Supercomput. 2020; 76(12): 9493–532. https://doi.org/10.1007/s11227-020-03213-1
11. Leicher A, Kuntze N, Schmidt AU. Implementation of a Trusted Ticket System. In: Gritzalis D, Lopez J, editors. Emerging Challenges for Security, Privacy and Trust. Berlin, Heidelberg: Springer Berlin Heidelberg. 2009; 152–63. https://doi.org/10.1007/978-3-642-01244-0_14
12. Khalil I, Dou Z, Khreishah A. TPM-Based Authentication Mechanism for Apache Hadoop. International Conference on Security and Privacy in Communication Networks. 2015; 105–122. https://doi.org/10.1007/978-3-319-23829-6_8
13. Shahin D, Ennab H, Saeed R, Alwidian J. Big Data Platform Privacy and Security, A Review. Int J Comp Sci Netw Secur. 2019; 19(5): 24-34.
14. Filaly Y, Mendili FE, Berros N, Idrissi YEBE. Hybrid Encryption Algorithm for Information Security in Hadoop. Int J Adv Comput Sci Appl . 2023; 14(6): 1295-302. https://dx.doi.org/10.14569/IJACSA.2023.0140613 7
15. Guan S, Zhang C, Wang Y, Liu W. Hadoop-based secure storage solution for big data in cloud computing environment. Digit Commun Netw. 2024; 10(1): 227–36. https://doi.org/10.1016/j.dcan.2023.01.014
16. Chen Y, Hao Y, Yi Z, Wu K, Zhao Q, Wang X. Searchable Encryption System for Big Data Storage. Commun Comput Inf Sci. 2021; 1452: 139–15. Springer, Singapore. https://doi.org/10.1007/978-981-16-5943-0_12
17. Anand K. Sentry to Ranger - A Concise Guide. Cloudera Blog. 2021.
18. Strata. Cloudera introduces RecordService for security, Kudu for streaming data analysis. ZDNET. 2015.
19. Cloudera. Apache Ranger. 2022.
20. Cloudera. Apache Knox Gateway Overview. 2022.
21. GoCypher. Eleven-Z/rhino. GitHub. 2020.
22. Baig HA. A Protection Layer over MapReduce Framework for Big Data Privacy. Int J Comput Inf Technol. 2022 Apr; 11(2): 68-73. https://doi.org/10.24203/ijcit.v11i2.263.
23. Baig H A, Sharma Y K, Ali S Z. Privacy-Preserving in Big Data Analytics: State of the Art (September 12, 2020). Int. Conf. on Business Management, Innovation & Sustainability (ICBMIS) 2020. http://dx.doi.org/10.2139/ssrn.3713826
24. Apache Software Foundation. ORC Specification v1. 2021.
25. Baig HA, Jummani DF, Ali SZ. A Framework for Preserving the Privacy of Data in Hadoop Clusters using Column Encryption. Int. J. Adv. Res. Eng. Technol. 2021; 8: 17894-902.

2024, 21(5 Special Issue): 1798-1810
https://doi.org/10.21123/bsj.2024.10550
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

إطار عمل للحفاظ على الخصوصية قائم على تشفير الأعمدة لمجموعات بيانات Hadoop الكبيرة

هداية علي بيك

قسم تقنية المعلومات، كلية علوم الحاسب والمعلومات، الجامعة التقنية والعلوم التطبيقية، صور، سلطنة عمان.

الخلاصة

أدى النمو الهائل للإنترنت وإنترنت الأشياء والحوسبة السحابية في الآونة الأخيرة إلى ارتفاع كبير في البيانات عبر مختلف قطاعات الأعمال والصناعة. أصبحت البيانات الضخمة اتجاها متزايدا في السنوات الأخيرة، حيث جذبت انتباه الأكاديميين وقادة الشركات والمسؤولين الحكوميين في جميع أنحاء العالم. Hadoop هو إطار عمل شائع الاستخدام لمعالجة البيانات الضخمة. إن توسيع البيانات هذا لديه القدرة على توفير مزايا كبيرة ومفيدة، وقد تم تحقيق بعض النجاح المبكر من الناحية الفنية في التعامل مع مثل هذه الكمية الكبيرة من البيانات. فإلى جانب فوائده العديدة، فإن له أيضًا عددًا كبيرًا من العيوب. وتشمل هذه، على سبيل المثال لا الحصر، تخزين البيانات وتبادلها وتنظيمها ونقلها وتحليلها وتصورها وأمنها وخصوصيتها. في هذا البحث، يتم دراسة الآثار المترتبة على الخصوصية لتحليلات البيانات الضخمة. تقترح العديد من المنشورات طرقًا لتأمين البيانات الضخمة. كل تقنية لها مزايا وعيوب. بغض النظر عن قوانين الخصوصية، يجب على مطوري التطبيقات حماية البيانات الحساسة.

الكلمات المفتاحية: البيانات الكبيرة، التخزين العمودي، تحليلات البيانات، التشفير، هادوب، HDFS، خصوصية.