AN EFFICIENT APPROACH FOR BIGDATA SECURITY BASED ON HADOOP SYSTEM USING CRYPTOGRAPHIC TECHNIQUES

Saritha Gattoju[1]
[1]Research Scholar, Department of Computer Science,
Gitam Institute of Science (Deemed to be University), Visakhapatnam, India.
saritha760@gmail.com
ORCID: https://orcid.org/0000-0002-0881-2973

Dr. V. Nagalakshmi[2]
[2] Professor, Department of Computer Science,
Gitam Institute of Science (Deemed to be University), Visakhapatnam, India.
Nagalakshmi.vadlamani@gmail.com
ORCID: https://orcid.org/0000-0003-1514-3592

Abstract:

When relational database systems could no longer keep up with the huge amounts of unstructured data created by organizations, social media, and all other data-generating sources, big data came into being. The amount of data being added every day, together with Hadoop, makes for an urgent and growing need for more data processing solutions. The MapReduce programming model is one common approach for processing and handling huge amounts of data, especially when used to big data research. As HDFS, a distributed, scalable, and portable file system constructed in Java for the Hadoop architecture, is already useful, it is noteworthy that it is built using Java technology. This computing environment suffers from two issues. First, when intruders access the system, they can steal or corrupt the data stored in the system. The AES encryption mechanism has been implemented in HDFS to safeguard the security of data stored in HDFS. Some data saved in HDFS can be secured with the application of AES encryption technique. I conducted an extensive research on security challenges around large data in the context of Hadoop, along with numerous solutions and technologies utilized to secure it.

Keywords: HDFS, MapReduce, AES encryption method, Hadoop

I Introduction

There are currently a growing number of interested communities in cloud computing, which has led to the creation of substantial software resources, storage, and high-performance computing resources [1] available to users. A tremendous amount of data is created every day in the digital world, which demands a considerable quantity of storage space, processing power, and system performance. Another well-known cloud computing platform is GFS and MapReduce, which are both used by Google. Analysis of the hazards helped increase the odds of a predictive capability's analysis and Big Data features by way of data analytics. A second issue associated with cloud is its inherent variability; it can be tailored, but it may be unsecure, more expensive, slower, incompatible, less reliable, or difficult to manage. Over the past few years, data sizes have increased from tens of terabytes to multiple zettabytes (that's 100,000,000,000,000,000,000 bytes), and they're only getting bigger. To fully glean insights from different and complicated data sets, you need a collection of methodologies [2]. Concerns related to Big Data are categorized under four headings: volume, velocity, variety, and validity. When it comes to data, every problem has its own mission, which is to get it through to the end.

- Volume: The term "large" by itself defines the size of the data in Big Data. Big data is connected with the amount of data produced. In the near future, it is predicted that the data will reach petabytes in size, and this could eventually expand to zettabytes.
- Velocity: Velocity in Big data involves measuring the pace of multiple data streams. The speed at which data flows is part of the velocity characteristic.
- Variety: The diversity of data is represented by the amount of variability in the data. You can include any type of media as long as it's formatted properly.

- Value: The value of data is measured in terms of its utility in making decisions. In terms of analyzing the data, “data science” deals with data, but “analytic science” concerns predictive data analysis. Varying users can execute various querying the data and so has the ability to pull out valuable results from the filtered data and afterwards rate the results based on the dimensions they need. These studies allow users to discover the current business trends with their plans they can make changes
- Complexity: The complexity of a system is measured by the extent of interdependence within enormous data structures, and a tiny change in one or a few pieces can lead to extremely huge changes or a little change that causes a shift in only a portion of the system broad reaching or disseminating impacts on the system, or no alterations at all (Katal, Wazid, & Goudar, 2013) [3] (possibly very large).

With the growing amount of data, there is now a new security challenge.

Data sourced from sensors, weather data, social media content, retail transactions, and cell phone GPS signals are all part of big data. On a daily basis, hundreds of petabytes of unstructured data are generated online, and many of these megabytes have value in terms of business applications if they can be collected and processed. Examples, data collection is done in the cell tower and refinery sensor and seismic exploration industries, while data collection in the utility industry is done in power plants and distribution systems [4]. Personal data, such as social security numbers, is increasingly available to businesses from potential customers and customers themselves. Your name, address, and credit card details, plus information about your patterns of buying habits and usage are exposed.
Huge new targets for hackers and other cyber criminals have opened up due to the increasing availability of big data. Previously, the data had little use for companies, but thanks to privacy rules and regulations, this data is now important and must constantly be secured [5].

II.Related Work

Data Processing in Hadoop

As the World Wide Web has grown in popularity over the years, so has the amount of data it generates. This spurred on a great deal of data generation, and the sheer volume of data being generated was hard to analyses and store using the standard relational database systems. In addition, the unstructured data like movies, photos, etc. was also created. There are no relational databases capable of handling this kind of data. Hadoop was created to combat these challenges. The Apache Hadoop architecture enables for the efficient and rapid processing of massive amounts of data [6]. Storing large quantities of structured and unstructured data is feasible with data catalogues.

Building Blocks of Hadoop

1. HDFS (The storage layer)
The name "Hadoop Distributed File System" simply describes the storage layer of Hadoop, which keeps data distributed across multiple machines (master and slave configuration). Each chunk of data is broken into separate blocks, which are then distributed over different data nodes. To ensure data safety, the data blocks are replicated across several data nodes.
The two major processes involved in processing the data are at work.

a. Name Node
The main machine is running it. It records where the data is located in the cluster and remembers where each file is located in the file system [7]. To make specific operations on the data, the Name Node interacts with the client apps. The Name Node will send back a list of Data Node server URLs when it receives the request.

b. Data Node
The entire process is carried out on each slave machine. HDFS data blocks are stored in a separate file in the local file system. In other words, it is a block containing the real data. It continuously monitors the user's heartbeat and is prepared to receive a request from the Name Node to retrieve the data.

2. MapReduce (The processing layer)

It is a programming approach based on Java that is used on top of the Hadoop framework for faster processing of massive quantities of data. It processes this vast data in a distributed environment utilizing numerous Data Nodes which enables parallel processing and speedier execution of operations in a fault-tolerant method. In order to be processed by the mappers, data are divided into many chunks, after which each chunk is assigned a new key-value pair [8]. The data may not be ready for processing in its raw form. Thus, Input Split and Record Reader are used to generate input data that is compatible with the map phase.
The data that is to be handled by an individual mapper is mapped to the logical form known as input Split. This division of notes into records produces key-value pairs for the Notes list. The record form of the output is created by encoding the byte-oriented representation of the input.

This data is subsequently sent to the mappers, who do the additional processing on the information. There are three main phases to every MapReduce task. The first of these is the Map phase, the second is the Shuffle phase, and the third is the Reduce phase.

a. Map Phase

The data has entered the first phase of processing at this point. Once every input from the Record Reader has been processed, these must be turned into intermediate tuples (key-value pairs). Local disc mappers store this intermediate output. Because the values of these key-value pairs can differ from the ones used as input from the Record Reader, the values of these key-value pairs should be saved separately. Local reducers (combiners) are also referred to as combiners [9]. They use aggregations inside the context of one mapper, but they do not calculate data outside of this context. Ensuring that the values that have the same key are all combined in a single reducer is important. The partitioned does this process. The mechanism first applies a hash function on the key-value pairs and then merges the resulting sets. It makes sure that the overall workload distribution is divided equally throughout reducers. Typically, when working with many reducers, practitioners come into play.

b. Shuffle and Sort Phase

The mapper to reducer step is achieved during this phase. Shuffling is a term used to describe this process. Additionally, the output is sorted prior to being distributed to the reducers. Key-value pairs are sorted according to the keys. Reducing the time required for computations also helps the reducers do their work. This sorting of the keys enables the reducer to conduct a new operation each time it receives a different key.

c. Reduce Phase

The results of the map step are fed into the reduce step. The reduced key-value pairs are produced by calling the reduce function on each of these input key-value pairs. Key values are supplied to the reduction function, which performs specific actions based on them. Combining and filtering the data will yield the aggregated output. To do the reduction operation, just post the code. It can result in zero or more key-value pairs.

3. YARN (The management layer)

Hadoop's resource handling component is Yet Another Resource Navigator. Allocating resources requires cooperation among the background programmes operating on all the devices (a Node Manager on the slave computers and a Resource Manager on the master node). The Resource Manager is the crucially important component of the YARN layer, which coordinates resource allocations between all of the application instances and forwards requests to the Node Manager. The Node Manager measures resource usage and informs the Resource Manager on these measurements [10]. The tasks are executed on every Data Node, and each Data Node has one instance of the task.

The full data processing workflow on Hadoop is summed up as follows

- To the best of our knowledge, this input Split causes the Hadoop distributed file system to split the data residing on HDFS into numerous blocks of data. The data is separated in one of two ways based on the Input format.
- Record Reader transforms the data into key-value pairs. When the data is byte-oriented, the Record Reader turns it to record-oriented data. This data is being used to generate the mapper's output.
- To create the intermediate key-value pairs, the mapper, which is nothing but a user-defined function, processes these key-value pairs.
- In order to limit the quantity of data passed from the mapper to the reducer, the combiners locally reduce the set of pairings within the scope of one mapper.
- By ensuring that all values with the same key are merged together into a single reducer, as well as applying an even distribution of duties among reducers, the practitioner is helpful.
- For every intermediate key-value pair, the shuffler organizes it into reducers, sorts on the basis of keys, and redistributes to the reducers. This input is passed to the reducers, who then produce the result.
- Reduce concatenates all of the values for each key, which are stored in HDFS via Record Writer. The output format must be defined before writing the data back to HDFS.

III THREATS TO SECURITY IN HADOOP SYSTEM

The specific dangers of working with data in the Hadoop ecosystem include the following:

- Unauthorized client: The use of a Data-transfer protocol, such as pipeline streaming, by an unauthorized client allows for the writing and reading of a data block of a file. And if access privileges are granted, and jobs can be submitted to a queue, deleted, or changed in priority. By setting up and using task trackers within Map tasks, intermediate data from the Map operation may be accessed.
- Task: Regardless of how the task may execute, it always interacts with the host operating system to utilize any tasks (either upstream or downstream), as well as any data in-between, such as intermediate Map outputs or the local storage of the Data Node that runs on the same physical node. In this alternative explanation, it's possible that a task or node is pretending to be a Hadoop service component, such as a

Name Node, job tracker, Data Node, task tracker, or in fact any of the other services, including the MapReduce Framework services, such as the job tracker.

- Unauthorized user: A malicious actor who is not a member of the authenticated group could execute arbitrary code, gain unauthorized access, or perpetrate other attacks by accessing an HDFS file using the RPC or HTTP protocols. Additionally, he might be collecting extra information by sniffing data packets exchanged between Data nodes and Oozie client applications and submitting a workflow to Oozie as another user. Data Nodes does not impose access control; therefore, he could overcome any security protections and obtain access to data blocks, including those that hold sensitive data.

3.1 SECURITY ISSUES IN HADOOP SYSTEM

Due to Hadoop's present security difficulties, data center managers and security specialists are encountering several challenges. These are the security issues:

- Fragmented Data: When using Big Data clusters, you should think about and design for redundancy and resiliency of the data that may be moved to and from multiple nodes. Because of the complexity that is introduced as a result of fragmentation, security becomes a concern owing to a lack of a security model.
- Distributed Computing: Due to significant levels of parallel processing, data sources are always being processed. Attackable environments are constructed that are at far higher risk of attacks than well-monitored and monolithic repositories.
- Controlling Data Access: At the schema level, big data provides access control alone. An even finer level of granularity for resolving users' responsibilities and access-related requirements is not possible.
- Node-to-node communication: In Hadoop, RPC over TCP/IP is not implemented as a secure communication mechanism; instead, it is used for RPC communication [12].
- Client Interaction: The communications with the resource manager, data nodes take place with the client. A compromised client will be likely to circulate malicious material or connections to either of the services.
- Virtually no security: Security was completely left out of the architecture of the big data stacks. There is no protection for the web against common dangers, either.

3.2 SOLUTION FOR BIG DATA SECURITY IN HADOOP

Performing detailed analysis on the security implications of big data Hadoop. In this paper, I've discussed several strategies for maintaining data security.

A. AUTHENTICATION

In Hadoop, Kerberos is the default authentication system. Kerberos was initially implemented with SASL/GSSAPI, which permits mutually authentic user, application, and Hadoop service RPC connections. Pluggable Authentication for HTTP Web Consoles is supported by Hadoop, meaning application and console implementers can authenticate HTTP connections by implementing their own authentication method. In the case of Hadoop Distributed File System (HDFS), HDFS Name Nodes and Data Nodes directly communicate using native Hadoop File System (HDFS) protocol, and also connect to the network using Remote Procedure Call (RPC) and mutual Kerberos authentication. When you utilize the Delegation Token, which is a two-party authentication mechanism, rather than utilizing Kerberos with three parties, you will find that it is significantly simpler and more effective [13]. One of MapReduce's unique features is the use of delegated tokens.

B. ACLs AND AUTHORIZATION

To enforce access control in Hadoop, UNIX-style permissions are used that match the file permissions concept. In HDFS, individuals and groups can enforce access control to files using permissions and ACLs in the Name Node. As long as work queues are subject to Access Control Lists (ACLs), you can define which users or groups can submit jobs to the queue or change the attributes of the queue. Hadoop uses fine-grained authorization and allows users to perform fine-grained actions, such as read or write permissions in HDFS and have access to services at the resource level utilizing ACLs for MapReduce.

C. ENCRYPTION

In order to protect the data while it's moving between the Hadoop system and the front-end system, it must be encrypted. To enable data in motion encryption in the Hadoop ecosystem, the SASL authentication mechanism is employed. SASL security is able to ensure that all data transferred between clients and servers is protected from a "man-in-middle" attack. Hadoop also offers encryptable channels including RPC, HTTP, and DTTP, which is used for transmitting data while it is in motion [14].

D. AUDIT TRAILS

It is required to audit the Hadoop ecosystem on a periodic basis in order to meet security compliance standards. The base audit features of HDFS and MapReduce are now available. Audit information is kept for Hive interactions, and it is done through the Apache Hive megastore [15]. Apache Oozie, the workflow engine, supports audit logs. While Hadoop does not have built-in audit logging, you can use audit logging monitoring solutions for the components that don't have them already.

IV Encryption of Data-at-Rest for Hadoop

The National Institute of Standards and Technology (NIST) selected the Data Encryption Standard (DES) as a symmetric key block cypher in 1977, and it is the most fundamental data improved standard. The AES symmetric key block cypher, often known as the Advanced Encryption Standard, is a symmetric key block cipher that uses symmetric keys (AES). The AES (Advanced Encryption Standard) was developed by the National Institute of Standards and Technology (NIST) and first released in 2001. (NIST). The Advanced Encryption Standard (AES) was developed to address the shortcomings of the Data Encryption Standard (DES), which employed a short cipher key and was therefore slower. Main key difference between the DES and AES are as follows

In DES, the block is divided into two halves and processed before being ciphered, but in AES, the block is treated as a whole before being ciphered

The DES method implements the Feistel Cipher approach, whereas the AES technique utilizes substitution and permutation.

The 56-bit key size of DES is less in comparison to the 128-bit, 192-bit, or 256-bit key size of AES. Each round of DES includes a different permutation (XOR, P-box, expansion permutation, S-box, swap and XOR), as well as XOR and S-box. Conversely, Subbytes, Shiftrows, Mix columns, and AddRoundKey all feature in AES rounds.

AES is more secure than DES since the key size is less.

In terms of performance, AES is relatively faster than DES.

The development and testing methods utilize an evolutionary/iterative approach in order to meet the objectives of this research. Changes to project requirements reduce the overall cost of the model. This approach facilitates easier handling of testing and debugging when minimal iterations are required.

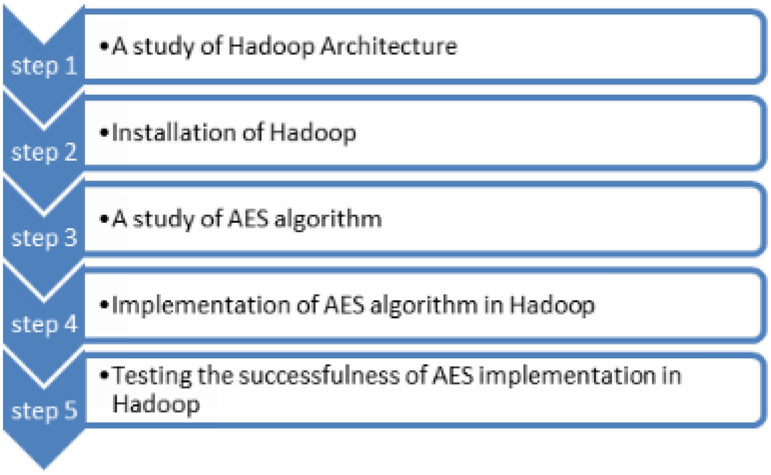

step 1
•A study of Hadoop Architecture
step 2
•Installation of Hadoop
step 3
•A study of AES algorithm
step 4
•Implementation of AES algorithm in Hadoop
step 5
•Testing the successfulness of AES implementation in Hadoop

Figure 1: Flowchart for data at rest process

This graphic depicts the essential procedures needed to accomplish the purpose of this article. Hadoop was the first research in the first stages of this project. In the second stage, Hadoop was installed in the server. In implementing the encryption technique, AES, and before deploying it in the server, researchers research and implement the algorithm, as well as test it to see whether it works in Hadoop. A virtual machine known as a "Hadoop server" is installed, which performs the task of running Hadoop. In establishing the virtual machine where Hadoop architecture is explored, Oracle VM VirtualBox is in use. studying the framework of Hadoop may be done by performing this The Hadoop platform consists of a Hadoop kernel, MapReduce, and HDFS (HDFS). The new features it introduces also include tools like Apache Hive, HBase, Oozie, Pig, and Zookeeper.

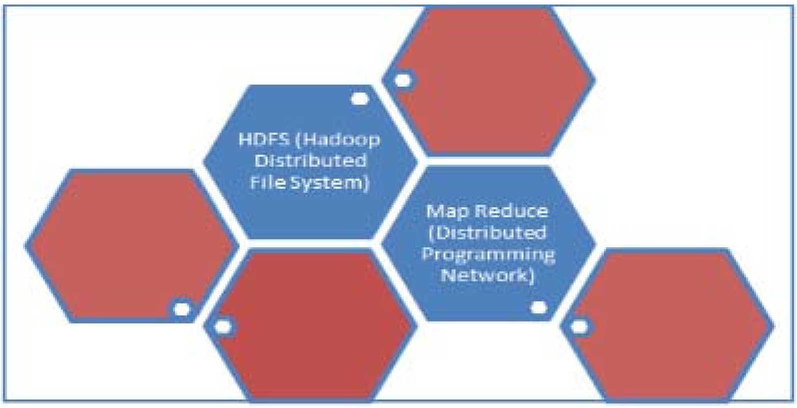

HDFS (Hadoop Distributed File System)
Map Reduce (Distributed Programming Network)

Figure 2: Hadoop Main Components

This distributed file system is failure resistant, and it is tasked with keeping data on the cluster. HBase is also a random read/write distributed NoSQL database. Hadoop computation analysis, meanwhile, is a high-level programming language for evaluating high-level Hadoop computation data. Another advantage of Hive is that it offers SQL-like access while also providing a relational model similar to SQL. In addition, Hive can be used to move or import data between relational databases and Hadoop. It is worth noting that Oozie is a tool to manage workflow and orchestration for dependent Hadoop jobs.

To keep data on Hadoop's HDFS encrypted, the use of AES encryption algorithm is required. Before the use of AES encryption can be implemented in HDFS, it is necessary to do an AES algorithm analysis. The data encrypted by the invaders will be decrypted if they obtain the necessary decryption key.

AES encryption was also successfully implemented. To identify any defects that can only be identified in the operating environment, framework testing must first be completed. This project has adopted advanced encryption standard (AES) encryption. The technique of encoding data in such a way that only authorized users may decode and utilize the data ensures that the data is protected while also bolstering defense capabilities.. The process of deciphering is simply the opposite of encrypting. Receiving an encrypted message turns the sender's encrypted text into plaintext, which is known as cleartext.

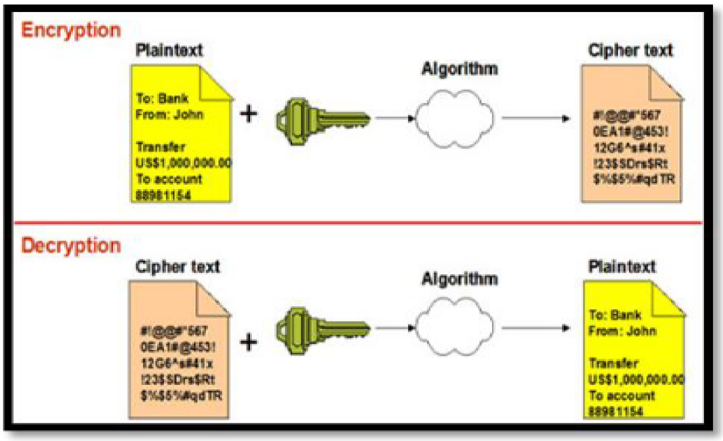


Figure 3: Decryption and Encryption Techniques

Because the data we are protecting has more than a minimal amount of it, symmetric encryption is used. Encryption and decryption are both performed throughout the process of symmetric key use. Decryption of a particular piece of cypher text requires the usage of the key to encrypt the data. The symmetric crypto scheme, known as AES (Advanced Encryption Standard), employs the same key to perform both encryption and decryption. AES is more than just for security, as it's extremely fast. to make the cypher text as difficult to decrypt as possible without needing a key

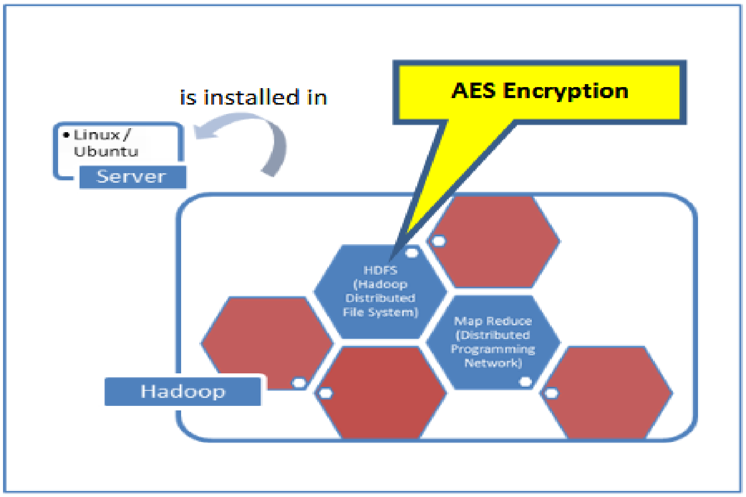

is installed in
AES Encryption
• Linux / Ubuntu
Server
HDFS (Hadoop Distributed File System)
Map Reduce (Distributed Programming Network)
Hadoop

Figure 4: Framework of project

Figure 4 illustrates the server, which is controlled and accessed by the admin. After installing open source software such as Hadoop, Apache, and a few customizations, the next step is to install the application. The only other benefit of the encryption mechanism is that it has been implemented into Hadoop's HDFS component, which is in the data node. Hadoop will use AES encryption technique to encrypt the data in Hadoop Data Lake at the data node. Only if the invaders have the correct key will the encrypted data be decoded into plaintext.

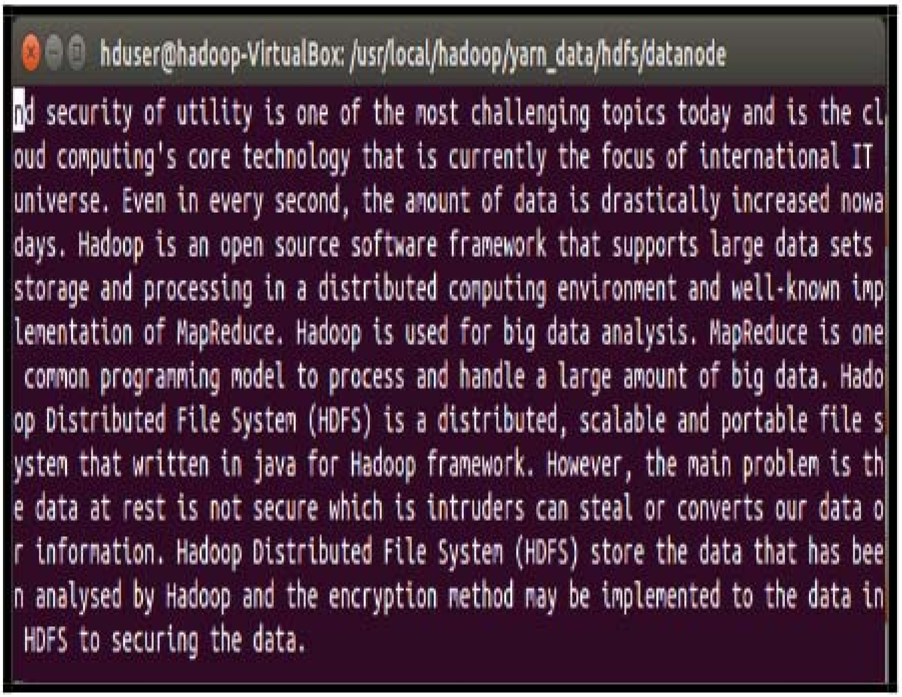
The main advantages of AES System are as follows

- Hardware and software both employ it, which means it's a secure protocol.
- Keys that are larger than what we usually use, such as 256, 192, and 128 bits, are employed.
- AES algorithm is thus more robust to hacking because of it. Comparable to Secure Socket Layer (SSL), There are various types of applications in which this form of security is frequently employed, including e-business, encrypted data storage, and wireless communication.
- One of the most widely distributed commercial and open source software solutions exists. Your personal information cannot be hacked.
- For 128-bit security, approximately 2128 attempts are required to succeed. Due to the complexity of the design, it is difficult to hack, and as a result, it is secure.

V. Results and discussions

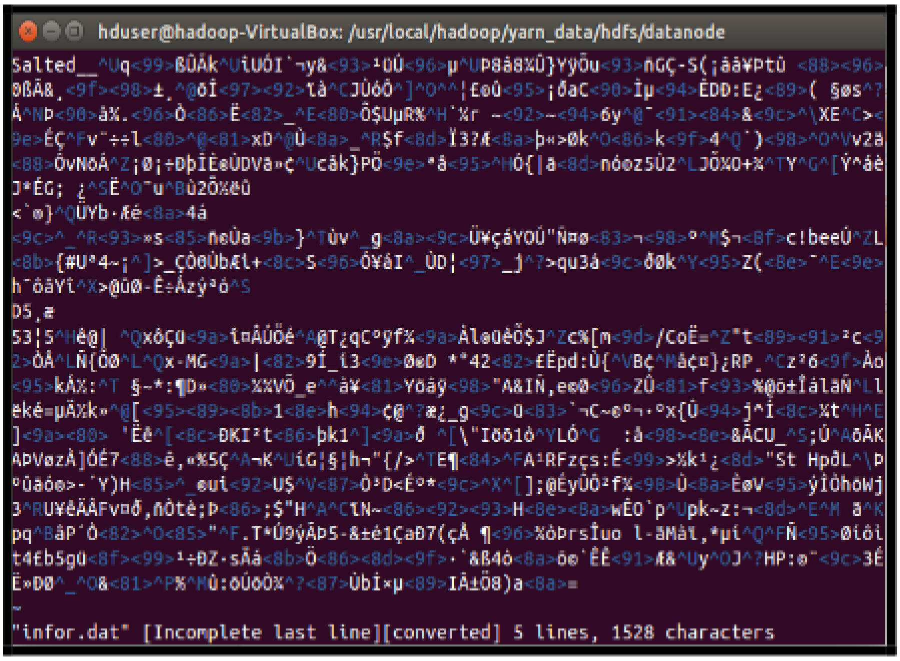
Data is generated and saved in the data node file in Hadoop HDFS when you complete the testing and reporting procedure. Fig. 5 illustrates that the original data content is stored in HDFS.

Once the encryption process has been created and labelled as encrypt.sh, it is ready to be used. In order to use the command, you must also provide the file name and where the file is going to be placed. The encrypted content of the file is AES-encrypted. Using the same key to encrypt and decrypt is known as symmetric cryptography. Since it is both secure and speedy, AES encryption is commonly used to protect information, but in addition, it has the additional benefit of using a 256-bit key rather than a 128-bit key, making it more difficult to crack. A greater key size means a higher level of security. Additionally, we've implemented the command to completely overwrite the original data.


hduser@hadoop-VirtualBox: /usr/local/hadoop/yarn_data/hdfs/datanode

```
nd security of utility is one of the most challenging topics today and is the cl
oud computing's core technology that is currently the focus of international IT
universe. Even in every second, the amount of data is drastically increased nowa
days. Hadoop is an open source software framework that supports large data sets
storage and processing in a distributed computing environment and well-known imp
lementation of MapReduce. Hadoop is used for big data analysis. MapReduce is one
 common programming model to process and handle a large amount of big data. Hado
op Distributed File System (HDFS) is a distributed, scalable and portable file s
ystem that written in java for Hadoop framework. However, the main problem is th
e data at rest is not secure which is intruders can steal or converts our data o
r information. Hadoop Distributed File System (HDFS) store the data that has bee
n analysed by Hadoop and the encryption method may be implemented to the data in
 HDFS to securing the data.
```

Figure 5: Original Data


hduser@hadoop-VirtualBox: /usr/local/hadoop/yarn_data/hdfs/datanode

```
Salted__^Uq<99>ßÛÄk^UiUÓI`¬y&<93>¹ûÚ<96>µ^UP8ä8%Û}Yÿõu<93>ñGÇ-S(¡ää¥Ptû <88><96>
0ßÄ8,<9f><98>±,^@õÏ<97><92>ìà^CJÙóÔ^]^O^^¦£øû<95>¡ðaC<90>Ìµ<94>ÈDD:E¿<89>( §øs^?
Å^NÞ<90>ä%.<96>Ò<86>Ë<82>_^E<80>Õ$UµR%^H`%r –<92>–<94>6y^@¯<91><84>&<9c>^\XE^C><
9e>ÉÇ^Fv¨÷÷l<80>^@<81>xD^@Ù<8a>_^R$f<8d>Ï3?Æ<8a>þ«>Øk^O<86>k<9f>4^Q`)<98>^O^Vv2ä
<88>ÖvNöÁ^Z¡0¡÷ÐþÏÈ®ÜDVä¤ç^Ucåk}PÖ<9e>ªâ<95>^HÓ{|ä<8d>nó®z5ÜZ^LJÕ%O+%^TY^G^[Ý^áè
J*ÉG; ¿^SË^O¯u^Bù2Õ%ëÛ
<`®}^QÜYb·Æé<8a>4á
<9c>^_^R<93>»s<85>ñeÙa<9b>}^Tùv^_g<8a><9c>Ü¥çáYOÚ"Ñ¤ø<83>¬<98>º^M$¬<8f>c!beeÛ^ZL
<8b>{#Uª4~¡^]>_ÇÒ0ÙbÆl+<8c>S<96>Ó¥áI^_ÙD¦<97>_j^?>qu3á<9c>ð0k^Y<95>Z(<8e>¯^E<9e>
h¯ôäYí^X>@û0-Ê÷Ázý²ó^S
D5,æ
53¦5^Hê@| ^QxôÇU<9a>ì¤ÅÚÖé^A@T¿qCº9f%<9a>Ål®ÜêÕ$J^Zc%[m<9d>/CoË=^Z"t<89><91>²c<9
2>ÒÅ^LÑ{Ò0^L^Qx-MG<9a>|<82>9Ï_ì3<9e>0®D *°42<82>£Ëpd:Ù{^VB¢^Mâ¢¤};RP¸^Cz²6<9f>Åo
<95>kÁX:^T §–*:¶D»<80>%%VÕ_e^^à¥<81>Yöäÿ<98>"A&IÑ,e®0<96>ZÛ<81>f<93>%@ö±ÎálaÑ^Ll
ëké=µÃ%k»^@[<95><89><8b>1<8e>h<94>¢@^?æ¿_g<9c>0<83>`¬C~®º¬·ºx{Û<94>j^Ì<8c>%t^H^E
]<9a><80> 'Ëê^[<8c>ÐKI²t<86>þk1^]<9a>ð ^[\"Iðõ1ò^YLÓ^G  :å<98><8e>&ÃCU_^S;Û^AõÃK
AÞVøzÀ]ÔÉ7<88>ê,«%5Ç^A¬K^UíG¦§¦h¬"{/>^TE¶<84>^FA¹RFzçs:É<99>>%k¹¿<8d>"St HpðL^\Þ
ºûäó®>-´Y)H<85>^_®uí<92>U$^V<87>Ò³D<ɺ*<9c>^X^[];@ÉyÛÔ²f%<98>Ù<8a>ÈøV<95>ýÌÒhöWj
3^RU¥êÄÅFv¤ð,ñÒtè;Þ<86>;$"H^A^CìN~<86><92><93>H<8e><8a>wÈO`p^Upk~z:¬<8d>^E^M ã^K
pq^BâP´Ò<82>^O<85>"^F.T*Ú9ýÃÞ5-&±é1ÇaÐ7(çÅ ¶<96>%òÞrsÎuo l-ãMàl,*µí^Q^FÑ<95>0íôì
t4£b5gû<8f><99>¹÷ÐZ·sÅá<8b>Ö<86><8d><9f>·`&ß4ò<8a>õ®`ÊÈ<91>Æ&^Uy^OJ^?HP:ø¨<9c>3É
Ë»D0^_^O&<81>^P%^Mû:öÚöÒ%^?<87>ÙbÌ×µ<89>IÄ±Ö8)a<8a>=
~
"infor.dat" [Incomplete last line][converted] 5 lines, 1528 characters
```

Figure 6: Script to be encrypted

In figure 6, you can see the encrypted result. The encrypted contents of the file are represented as cypher text. The information in the encrypted file cannot be understood unless the encrypted file is decrypted by using the same key.

Table-1 Variations in cryptographic algorithms

Sno	Algorithm	Key size	Block size	Rounds	Structure
1	DES	56 bits or 64 bits	64 bits	16	Feistel network
2	AES	128, 192 or 256 bits	128	0, 12 or 14	Feistel network
3	PRESENT	80 or 128 bits	64 bits	31	SPN-Substitution–permutation network

4	IDEA	128 bits	64 bits	8.5	Lai–Massey scheme
5	CAST	40 to 128 bits	64 bits	12 or 16	Feistel network
6	Blow Fish	32–448 bits	64 bits	16	Feistel network
7	CLEFIA	128, 192 or 256 bits	128 bits	18,22, or 26	Feistel network
8	CAMELLIA	128, 192 or 256 bits	128 bits	18 or 24	Feistel network

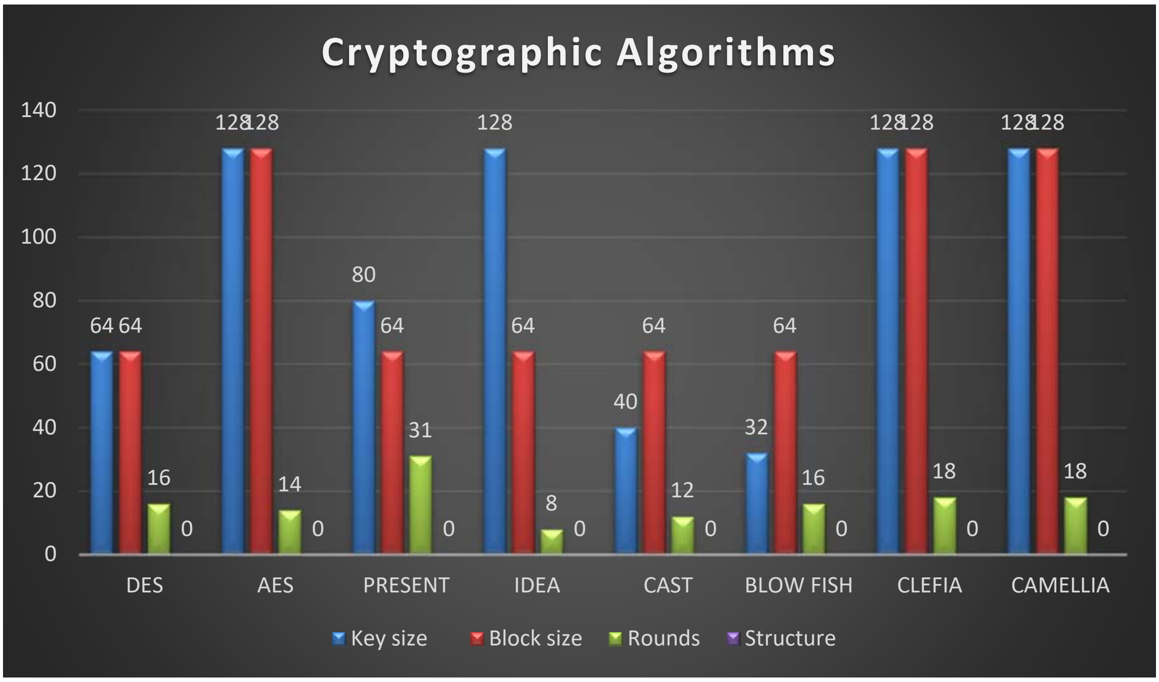


Fig- 7 Comparison of Various Algorithms

Various cryptographic algorithms and their results have presented in the above.

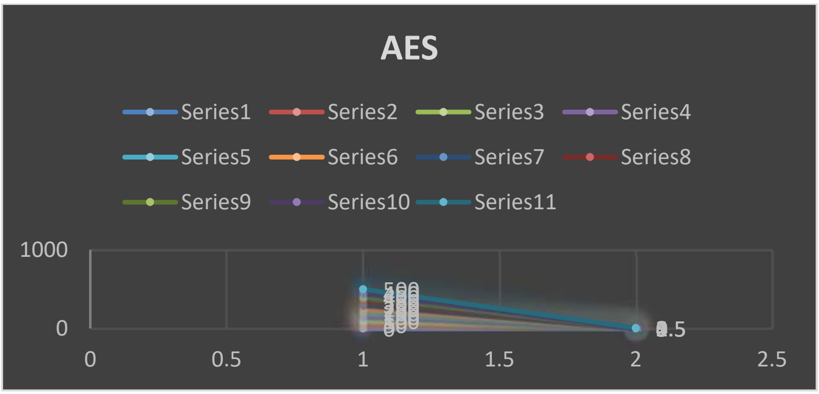


Fig-8 AES comparison with Data in KB and Time

AES itself compared with data in kilo bytes and time in seconds the results are shown in above.

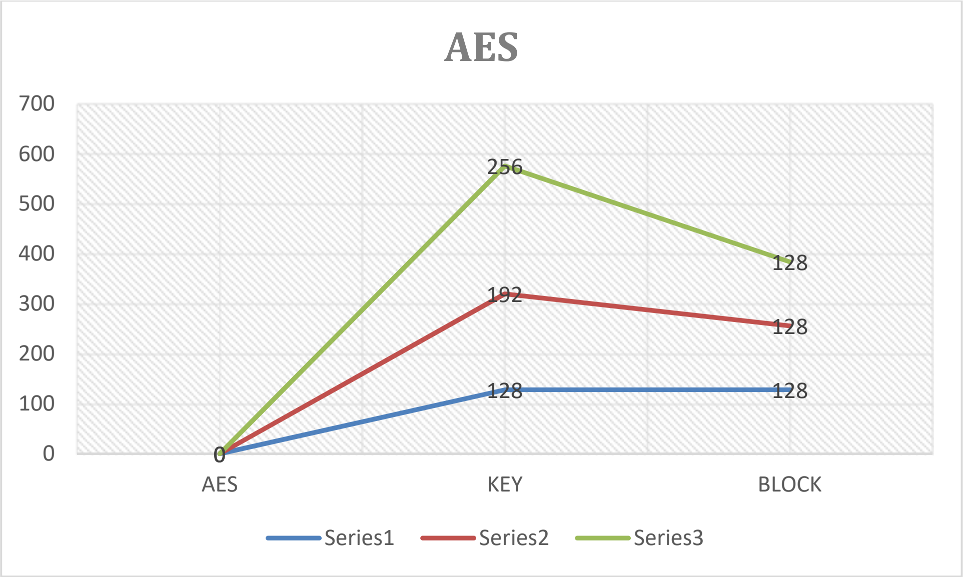
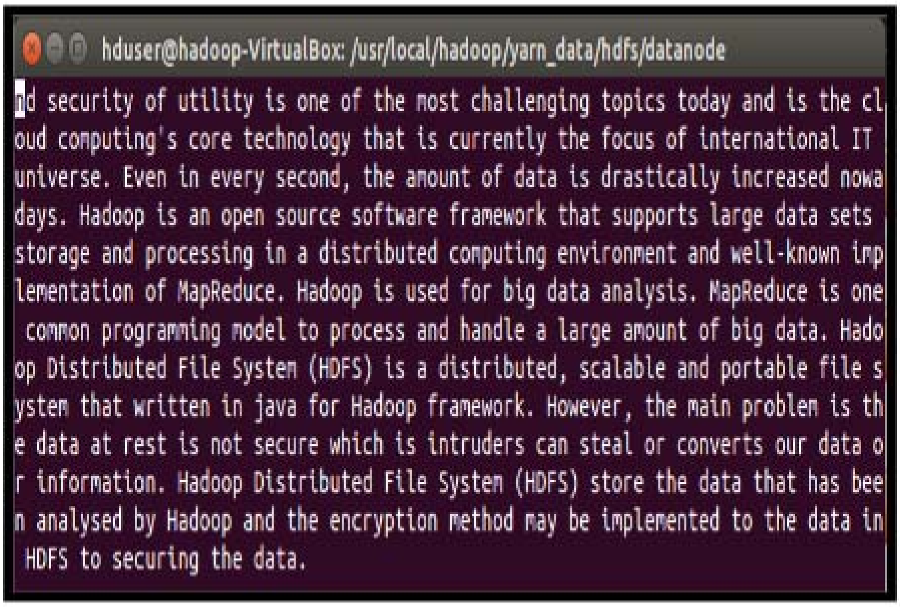


Fig-9 AES with Different Keys and Block size

In figure 6, you can see the encrypted result. The encrypted contents of the file are represented as cypher text. Decryption without the proper key renders the information in the encrypted file incomprehensible.


hduser@hadoop-VirtualBox: /usr/local/hadoop/yarn_data/hdfs/datanode

```
nd security of utility is one of the most challenging topics today and is the cl
oud computing's core technology that is currently the focus of international IT
universe. Even in every second, the amount of data is drastically increased nowa
days. Hadoop is an open source software framework that supports large data sets
storage and processing in a distributed computing environment and well-known imp
lementation of MapReduce. Hadoop is used for big data analysis. MapReduce is one
 common programming model to process and handle a large amount of big data. Hado
op Distributed File System (HDFS) is a distributed, scalable and portable file s
ystem that written in java for Hadoop framework. However, the main problem is th
e data at rest is not secure which is intruders can steal or converts our data o
r information. Hadoop Distributed File System (HDFS) store the data that has bee
n analysed by Hadoop and the encryption method may be implemented to the data in
 HDFS to securing the data.
```

Figure 7: Data to be decrypted.

Once that happens, the programmed which decryption uses is generated and named decrypt.sh. When you use the command in these programming, it will read the file name that needs to be decrypted, and locate the encrypted file where it will be stored. The following image depicts the final product of the decryption procedure after it is completed in order to return the original data

VI.CONCLUSON

Today, the size of data is rising rapidly, which means big data demands different data-protection strategies. Hadoop is used to process huge data sets, and in this work, I focus on security challenges when processing large data sets in Hadoop environments. In the era of Big Data, data security is a serious issue. To prevent unauthorized users from changing the data that is stored in Hadoop, the project uses Advanced Encryption Standard (AES) encryption. Only the users who have been granted access to the file will be able to open it. A comprehensive test was done on the Data Node encryption algorithm. Hadoop's encryption method has shown itself to be correct by successfully encrypting the contents of the file.

References

[1] Y. Li and D. Zhang, "Hadoop-Based University Ideological and Political Big Data Platform Design and Behavior Pattern Mining," 2020 International Conference on Advance in Ambient Computing and Intelligence (ICAACI), 2020, pp. 47-51, doi: 10.1109/ICAACI50733.2020.00014.

[2] Y. Wu, X. Li, J. Liu and L. Cui, "Hadoop-EDF: Large-scale Distributed Processing of Electrophysiological Signal Data in Hadoop MapReduce," 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2019, pp. 2265-2271, doi: 10.1109/BIBM47256.2019.8983371.

[3] V. Sontakke and R. B. Dayanand, "Optimization of Hadoop MapReduce Model in cloud Computing Environment," 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), 2019, pp. 510-515, doi: 10.1109/ICSSIT46314.2019.8987823.

[4] A. Shah and M. Padole, "Load Balancing through Block Rearrangement Policy for Hadoop Heterogeneous Cluster," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 230-236, doi: 10.1109/ICACCI.2018.8554404.

[5] G. s. Bhathal and A. S. Dhiman, "Big Data Solution: Improvised Distributions Framework of Hadoop," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018, pp. 35-38, doi: 10.1109/ICCONS.2018.8663142.

[6] D. Sharma, G. Pabby, and N. Kumar, "Challenges Involved in Big Data Processing &," vol. 5, no. Viii, pp.841–844, 2017.

[7] M. M. Shetty and D. H. Manjaiah, "Data security in Hadoop distributed file system," Proc. IEEE Int. Conf. Emerg. Technol. Trends Comput. Commun. Electr. Eng. ICETT 2016, pp. 939–944, 2017.

[8] S. Singh, P. Singh, R. Garg, and P. K. Mishra, "Big Data: Technologies, Trends and Applications," vol. 6, no. 5, pp.4633–4639, 2015.

[9] M. B. Alam, "A New HDFS Structure Model to Evaluate the Performance of Word Count Application on Different File Size," vol. 111, no. 3, pp. 1–4, 2015.

[10] Fatma A. Omara Eman, S Abead, Mohamed H. Khafagy "A Comparative Study of HDFS Replication Approaches", the International Journal of IT andEngineering,8/2015 Volume 3, Issue 8, PP4-11

[11] C. Yang, W. Lin, and M. Liu, "A novel triple encryption scheme for Hadoop-based cloud data security," Proc. - 4thInt. Conf. Emerg. Intell. Data Web Technol. EIDWT2013, pp. 437–442, 2013.

[12] J. Repschl, "Cloud Computing Framework zur Anbieterauswahl," pp. 1–35, 2013.

[13] S. Park and Y. Lee, "Secure Hadoop with Encrypted HDFS," pp. 134–141, 2013.

[14] H. Y. Lin, S. T. Shen, W. G. Tzeng, and B. S. P. Lin, "Toward data confidentiality via integrating hybrid encryption schemes and Hadoop distributed file system, "Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA, pp. 740–747, 2012.

[15] Sean-Philip Oriyano, J. M. Tanna, M. P. Sanghani, M Ayushi, and R. J. Anderson, "A Symmetric Key Cryptographic Algorithm," Int. J. Comput. Appl., vol. 1, no. 15, pp. 73–114, 2010.