International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 05 | May-2018 www.irjet.net p-ISSN: 2395-0072

Providing Secure Environment For The Storage Of Big Data In Hadoop

Kavita K. Kanyan[1], Er. Ritika Mehra[2]

[1] PG Student, Department of Computer Science & Engineering, R.P. Inderaprastha Institute of Technology, Haryana, India
[2]Assistant Professor, Department of Computer Science & Engineering, R.P. Inderaparastha Institute of Technology, Haryana, India

---

Abstract – The complex and huge amount of data is referred to as Big Data. Hadoop provides the framework for storing and analyzing such huge amount of data. Big Data comes from different resources which results in various security issues. Hadoop doesn't come with built-in security.

This paper is about providing secure environment for the storage of Big Data in Hadoop. In this paper, we used Kerberos for the authentication and authorization purpose so that only authenticated and authorized users can access the services provided by Hadoop. Data-at rest encryption is provided by using the hybridization technique. The encryption algorithms used are AES, RSA and ToCrypt.

Key Words: Big Data, Hadoop, HDFS, Encryption, Decryption, AES, RSA, ToCrypt, Kerberos

1. INTRODUCTION

1.1 BIG DATA

The data in the world is growing rapidly every year it is difficult to process this complex and huge amount of data using traditional applications/tools. Big Data is a term used to describe such data. This data could be structured or unstructured.

Big Data 3 V's are [1].

Volume: At present the data existing is in petabytes and is supposed to increase to zettabytes in nearby future. The social media, financial institution, medical institution, government, Sensors, Logs producing data in order of terabytes every day and this amount of data is definitely difficult to be handled using the existing traditional systems.

Velocity: At present data change rapidly through the archived data, legacy collections and from streamed data that comes from multiple resources sensors, traditional file records, cellular technology, social media and many more.

Variety: At present data comes in different forms including data-streams, text, picture, audio, video, structured, semi structured, unstructured. Unstructured data is difficult to handle with traditional tools and techniques. Thus our traditional systems are not capable enough on performing the analytics on the data which is constantly in motion.

1.2 HADOOP

Hadoop is a java based free framework that can effectively store large amount of data in a cluster. This framework works in parallel on a cluster and has an ability to allow us to process data across all nodes. Hadoop environments include data from multiple sources such as weblogs, online transaction and social media interactions etc. of security sensitivities providing the need for security. Hadoop was originally designed without security in mind.

To ensure user identification and control user access to sensitive data it's important to create users and groups and then map users to groups. Permissions should be assigned and locked down by groups, and the use of strong passwords should be strictly enforced. Fine grained permissions should be assigned on a need-to-know basis only and broad stroke permissions should be avoided as much as possible.

1.3 HDFS

Hadoop Distributed File System (HDFS) is the storage system of Hadoop which splits big data and distribute across many nodes in a cluster. This also replicates data in a cluster thus providing high availability.

HDFS has a master /slave architecture. An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on [2]. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients. The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode.

1.3 SECURITY ISSUES IN HADOOP

Hadoop presents brand new challenges to data risk management: the potential concentration of vast amounts

of sensitive corporate and personal data in a low-trust environment.

Hadoop doesn't authenticate users or services, and there is no data privacy. As Hadoop was designed to execute code over a distributed cluster of machines, anyone could submit code and it would be executed. Although auditing and authorization controls (HDFS file permissions) were implemented in earlier distributions, such access control can be easily circumvented because any user could impersonate any other user with a command line switch.

Some of the Hadoop security challenges that are discussed in this paper are: How data is encrypted or otherwise protected while it is in storage (at rest), and how systems and users are authenticated before they access data in the Hadoop infrastructure.

2. AUTHENTICATION AND AUTHORIZATION

I. Authentication is the process of determining whether someone is who they claim to be. If Hadoop is configured with all of its defaults, Hadoop doesn't do any authentication of users.

II. Authorization is the function of specifying access rights to resources. Authorization tells us what any given user can or cannot do within a Hadoop cluster, after the user has been successfully authenticated.

To provide Authentication and Authorization in Hadoop cluster we used Kerberos an authentication protocol.

2.1 KERBEROS

Kerberos is an Authentication protocol for trusted hosts on unsecure Network. The trusted hosts mean these hosts that are kerberized need to belong to a particular Realm. Authentication is done by using a central server.

It works on the principal of Single Sign on as the user is asked to enter a password only once per work session.

Kerberos is highly time dependent so all the clients Clocks must be Synchronized [3].

Terms used in Kerberos:

REALM

Realm is collection of Principal than belong to the same domain. REALM is case sensitive always written in capital letters.

Principal

Principal is any entry in Kerberos database. It could be a User, Service and Server. Example Name/Instance@REALM.

Ticket

Ticket allows you to access some particular service. Client presents the ticket to application server to demonstrate the authenticity of its identity.

KDC

It stands for Key Distribution Centre. Its components are:

A. Database
It stores the principals.

B. Authentication server (AS)
This is responsible for authenticating the users.

C. Ticket granting server (TGS)
It is responsible for providing Ticket.

Application Server

It is a server that is running a particular service that we want to access e.g. IMAP server, SSH server which is running a service which is kerberized.

Steps Involved in Kerberos Authentication Process:

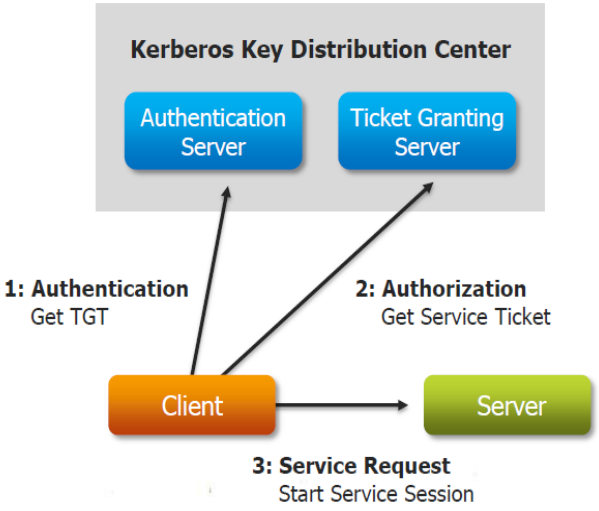

Fig-1: Kerberos Authentication Process

2.1.1 TGT Generation

The Client requests the Authenticator Server for the TGT. The KDC check the entry for the Client in its Database. If the Credentials matched, the KDC generates a TGT for the client. A Client-Krb session key is created and sent back to the Client with TGT which is encrypted by the Private Key of the KDC and Client Private Key. If the client has the credentials it decrypts the private key.

2.1.2 TGS Session Ticket Generation

The Client uses TGT and sends a request for the Ticket. The TGT provides a Ticket+ Client-Krb Session Key to the Client for Authentication.

2.1.3 Service Access

Now, if the Client wants to access the service of the Application server e.g. ssh service. It sends an Authenticator (User Name, IP and Time Stamp) + Session Key + TGT to the TGS. The TGS fetches the Authenticator + TGT and has the Client-krb session key, it decrypts the packet. The Authenticator Credentials (User Name, IP and Time Stamp) are matched by the KDC. If the credentials get matched the Ticket is sent to Client which is encrypted by the private key of the Application Server and client-krb session key and Client-app Session key. The Client can decrypt this Ticket and access the service.

To enable Hadoop security, add the following properties to the core-site.xml file on every machine in the cluster [4]:

```
<property>
      <name>hadoop.security.authentication</name>
 <value> kerberos </value> <!-- A value of "simple" would
                    disable security. -->
</property>

<property>
 <name>hadoop.security.authorization</name>
                    <value>true</value>
                        </property>
```

Fig-2: Hadoop core-site.xml file

The properties for each daemon (NameNode, Secondary NameNode, and DataNode) must specify both the HDFS and HTTP principals, as well as the path to the HDFS keytab file.

The Kerberos principals for the NameNode, Secondary NameNode, and DataNode are configured in the hdfs-site.xml file. The same hdfs-site.xml file with all three of these principals must be installed on every host machine in the cluster. That is, it is not sufficient to have the NameNode principal configured on the NameNode host machine only. This is because, for example, the DataNode must know the principal name of the NameNode in order to send heartbeats to it. Kerberos authentication is bi-directional.

3. PROPOSED APPROACH FOR ENCRYPTION

The best way to ensure data security is "cryptography". Cryptography uses encryption method to keep the data secure from intruders. As, Hadoop is the biggest vendor of processing and storing data at large scale on cloud, the security of data is of major concern. Hence, hadoop uses some encryption techniques to ensure security. No encryption algorithm provides complete security and they all have their own limitations and loopholes which cannot be ignored. To provide security the encryption and decryption of data is done using the hybridization technique [5].

The three algorithms used for the encryption and decryption purpose are AES, RSA and ToCrypt.

3.1 AES

The Advanced Encryption Standard (AES) is a symmetric-key block cipher algorithm for data encryption and decryption. Block cipher algorithms encrypt data on a per-block basis [6].

It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs and others involve shuffling bits around, each executed on data blocks of 16 bytes. Those operations are repeated several times, called rounds.

The number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys.

AES involves following steps:-

3.1.1 Initial Round

- AddRoundKey: each byte of the state is combined with a block of the round key using bitwise xor.

Rounds

- SubBytes: a non-linear substitution step where each byte is replaced with another.
- ShiftRows: a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.
- MixColumns: a mixing operation which operates on the columns of the state, combining the four bytes in each column.

Final Round (no MixColumns)

- SubBytes
- ShiftRows
- AddRoundKey.

3.1.2 Encryption

Mix column (Shift Rows (SubBytes (Text)))

3.1.3 Decryption

SubBytes (Shift Rows (Mix Column (Cipher)

3.2 RSA

RSA is an asymmetric cryptographic algorithm for the encryption and decryption of the data. Asymmetric means there are two different keys, public and private, one for encryption and one for decryption [7].

The RSA algorithm involves following three steps key generation, encryption and decryption:-

3.2.1 Key Generation:

Step 1: select random prime numbers p and q, such that p!=q

Step 2: calculate n=pq

Step 3: compute phi Ø=(p-1)(q-1)

Step 4: select public exponent e,1<e<Ø Such that gcd (e,Ø)=1

Step 5: calculate d $d=e^{-1} \bmod Ø$ Public key is {e,n} and Private key is d.

3.2.2 Encryption:

$c=m^e \bmod n$

3.2.3 Decryption:

$m= c^d \bmod n$

3.3 ToCrypt

ToCrypt is an algorithm for encryption and decryption. It uses the same key for both encryption and decryption of the data.

The steps involved in the ToCrypt algorithm are as follows:-

Step 1: A key for encryption and decryption is generated

By using a PRNG.

Step 2: Seed is generated using the mersenne twister.

Step 3: To encrypt the data the random number is XORed With the plain text, and the seed is appended to that.

Step 4: To decrypt the data first the seed is extracted from the ciphertext, after that the key is XORed with the ciphertext to generate the plain text.

3.4 HYBRID ENCRYPTION

The hybrid encryption of the data using AES, RSA, and ToCrypt is done using following steps :-

Step 1: Hash function is applied on the plain text, which in Turn generates the random session key.

HASH (PlainText)=SK

Step 2: AES uses the key generated in first step to encrypt the plain text.

AES (PlainText)=ciphertext1

Step 3: The random session key is encrypted using asymmetric algorithm RSA.

RSA (SK)=Encrypted Key

4: The encrypted key is appended to the ciphertext1 and the appended package is encrypted using ToCrypt.

ToCrypt(Ciphertext1+Encrypted Key)=ciphertext2

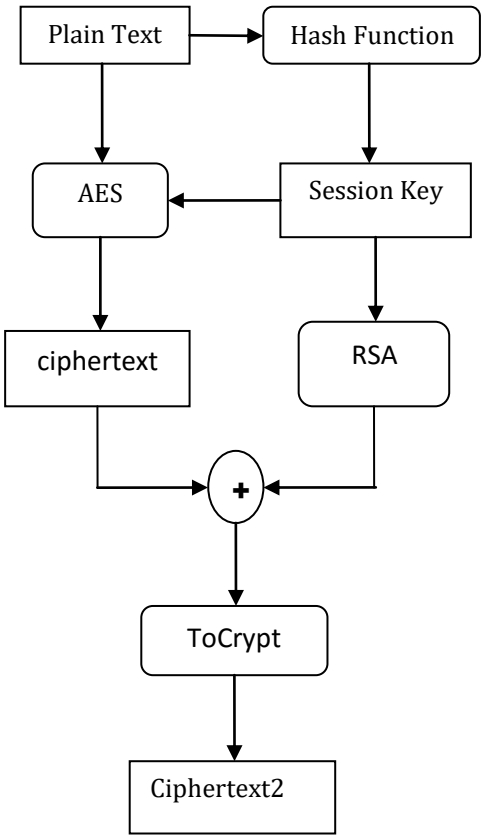

Fig 3: Encryption Process

3.5 HYBRID DECRYPTION

The decryption of the ciphertext involves following steps:-

Step 1: The ciphertext2 is decrypted using ToCrypt, this step will give us the encrypted key and ciphertext1

ToCrypt (ciphertext2)

Step 2: The encrypted key from the first step is decrypte using RSA private key.

RSA (Encrypted Key)=SK

Step 3: The AES decrypt the encrypted text using the session key from previous step.

AES (ciphertext2)= PlainText

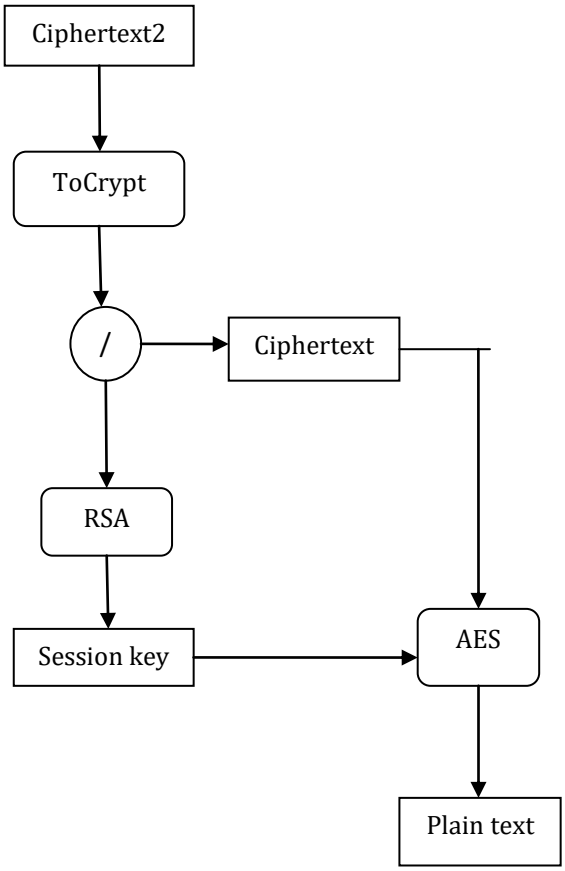

Fig 4: Decryption process

The above given flowcharts describes the various steps taken during the encryption and decryption of the data. The client will receive the ciphertext2, which contain the encrypted message and encrypted session key that would be used by the client to decrypt the data. After receiving the ciphertext2 the client will perform decryption on the text by using decryption process.

4. CONCLUSION

To secure data in Hadoop authentication of the client and encryption of the data is done. The authentication and authorization is done using authentication protocol kerberos. The encryption is done before the data will be uploaded in the Hadoop Distributed File System(HDFS). Asymmetrical and symmetrical encryption algorithms are used for the encryption and decryption of the data.

Our proposed approach includes how nodes and client applications are vetted before joining the cluster, how data at rest is protected from unwanted inspection. The proposed hybrid approach implement three level of security which makes hard for the malicious user to get access of the data.

5. REFERENCES

[1] Bermen, Jules J. "Principle of Big Data", Morgan Kaufmann, Waltham, 2013

[2] Hadoop Architecture
https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.

[3] Kerberos-The Network Authentication Protocol
https://web.mit.edu/kerberos.

[4] Authentication and Authorization in Hadoop
http://blog.cloudera.com/blog/2012/03/authorization-and-authentication-in-hadoop.

[5] Hsiao-Ying Lin, Shiuan-Tzuo Shen,Wen Guey Tzeng "Toward Data Confidentiality via Integrating Hybrid Encryption Schemes and Hadoop Distributed File System" http://ieeexplore.ieee.org/document/6184943/

[6] Asymmetric Encryption Standard
https://en.wikipedia.org/wiki/Advanced_Encryption_Standard.

[7] RSA(Cryptosystem)
https://en.wikipedia.org/wiki/RSA_(cryptosystem).

.