Design and Implementation of a Distributed Encryption System for the Cloud

Arun Mohan[#1], Sudheep Elayidom[*2]
[#1]Computer Science and Engineering, School of Engineering, CUSAT
[*2]Division of Computer Science, School of Engineering, CUSAT
[1]*arunmohan3157@gmail.com*
[2] *sudheepelayidom@gmail.com*

Abstract— "Big Data "- voluminous and variety of data from different sources. The Data can be either in the form of structure (or) unstructured Data. Privacy and security of Big Data is gaining high importance, since all the technologies are started to depend on Big Data. It is difficult to work with using most relational database management systems, desktop statistics and visualization packages since it requires massive parallel software running on tens, hundreds or even thousands of servers.
In this paper, we are going to discuss the Hadoop and the method for maintaining the privacy & security of big Data. Originally Hadoop was invented without any security model. The main goal is to propose a Hadoop system that maintains privacy and security at the client system. Advanced Encryption Standard (AES) enables protection to data at each cluster, it performs encryption/decryption before read/write respectively and we are using SHA 1 for user authorization.

Keywords— *Big Data, Hadoop, AES, HDFS, SHA-1*

*****

I. INTRODUCTION

Big data is a term for data sets that are so large or complex that traditional data processing application software is inadequate to deal with them. Data obtained from social media, software logs, cameras, microphones, cloud etc. all constitute Big Data. Privacy and security of this information is of high importance and priority. As Big Data constitutes both structured and unstructured data, implementing security mechanisms on this data is a challenge [1]. Hadoop was developed from GFS (Google File System) [6] and MapReduce papers published by Google in 2003 and 2004 respectively. It has been popular recently due to its highly scalable distributed programming or computing framework, it enables processing big data for data-intensive applications as well as many analytics. Hadoop is a framework of tools which supports running application on big data and it is implemented in java. It provides MapReduce programming architecture with a Hadoop distributed file system(HDFS), which has massive data processing capability with thousands of commodities hardware's by using simply its map and reduce functions.

A. Challenges of Big Data

Big data has three challenges i.e. the 3 V's Volume Velocity, and Variety

1) Volume: It is the data at rest. Usually, Terabytes and Exabyte's of existing data to process for accurate analysis. Distributed cloud storage is the solution for storing this information. Implementing security mechanisms for such high volumes of data becomes a challenge. Implementing Encryption and Decryption on this data stored on cloud is going to result in fall in performance.

2) Velocity: It is the data in motion [2]. Usually, streaming data has a few milliseconds to a few seconds for a response. Maintaining such response times along with ensuring privacy and security becomes difficult to achieve. Implementation of signatures, Encryption and Decryption on this fast streaming data will slow the processing speed or analysis of data.

3) Variety: It is data in many forms. Structured, Unstructured, text and multimedia, etc., various forms of data present a challenge for achieving the privacy and security of Big Data. Different implementations of security mechanisms are necessary for different forms of data. These are going to inhibit the scalability and performance of this system. In addition to the 3V's there is another challenge presented by Big Data.

B. Components

The main components of Hadoop Eco system are Hadoop Distributed File System [HDFS] and MapReduce [3]. HDFS is the cloud storage system for data and MapReduce is used for processing and analysing of data.

1) MapReduce: MapReduce [1] is the heart of Hadoop, which is a programming model meant for large clusters. It has a parallel computing framework and is obscure to responsibilities like parallelization, fault tolerance, data distribution and load

balancing. The purpose of implementing MapReduce is to process and generate large data sets. The computation of MapReduce takes a set of input key/value pairs and generates a set of output key/value pairs. The computation of generating the set of output key/value pairs is divided into two functions: Map function and Reduce function.

2) Hadoop Distributed File System (HDFS): Hadoop uses a block structured distributed file system for storing large volumes of data (tetrabytes and petabytes) called Hadoop Distributed File System [5]. All the individual files in the HDFS are divided into fixed size blocks. A cluster of machines with storage capacity are used for the storage of these blocks. A Data Node is the individual machine in the cluster. The distribution of data among the DataNode is handled by the HDFS. It is responsible for adding and removing nodes from the cluster and for DataNode recovery. To support fault tolerance HDFS implements replication of blocks. Major components of HDFS [5] include NameNode, DataNode and Secondary Name Node.

NameNode acts as the master of the system and is responsible for the management of blocks on the DataNodes. It runs on high quality software, as it's responsible for the storage of meta-data. It is the single-entry point for a failure happening in Hadoop cluster. The replication factor, specified by the application is stored in the NameNode. A Block Report and Heartbeat are received by the NameNode from each DataNode in the cluster. Block Report contains all the blocks on a DataNode. NameNode is aware of the racks and it determines the rack id for each DataNode.

DataNode acts as slaves and it is deployed on each machine in the cluster. DataNode stores the actual data. Its responsibility is to process the requests of client for read and write on the blocks. DataNodes hold only part of the overall data and responsible for processing the part of data it holds. The creation, deletion and replication of the blocks are handled by the DataNode when instructed by the NameNode. *Secondary NameNode* whole purpose is to have a checkpoint in HDFS. It's just a helper node for NameNode. That's why it also known as checkpoint node inside the community.

3) Working of HDFS: Client machine loads the file into the cluster. The DataNode breaks the file into fixed blocks and stores them throughout the cluster. The Name Node has the Block Report and is aware of the available racks. It decides which DataNode it should allocate. Replicas are placed in unique racks. Three Replicas are created by default and one replica is stored on a DataNode the second is stored on a DataNode on the same rack and the third is stored on the DataNode on a different rack. The replication factor is used to manage how many Replicas should be created and additional Replicas are randomly placed.

The client machine writes the data block to one DataNode and automatically the DataNode will replicate itself to the other DataNodes. Fig. 1. clearly shows the working of the HDFS system.

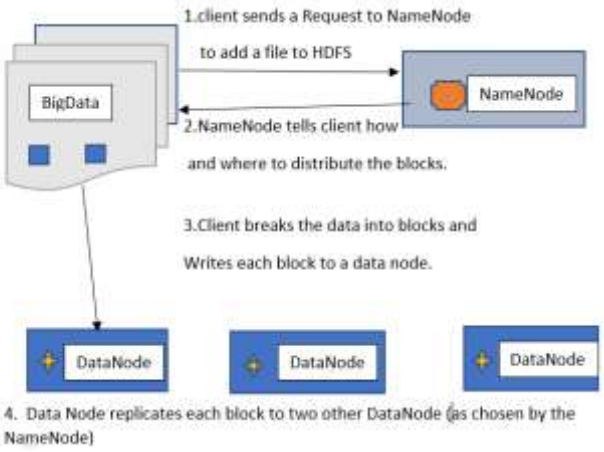

Fig. 1 HDFS Architecture

II. SECURITY RISKS IN HDFS

There are three kinds of security violations in HDFS, unauthorized access, unauthorized modification of data and denial of service or resource. Following are the areas where threat identify in Hadoop.

- *Hadoop does enforce authenticate any user or service*: unauthorized users may any HDFS cluster like owner via RPC of HTTP protocol.
- *DataNode can't have any access control mechanism to protect data block*: it is possible to write or modify existing data blocks to DataNode.
- *An attacker can presence as Hadoop service*: For example, code submitted by user register itself on MapReduce cluster as a new TaskTracker.
- *Super-user of system does anything without checking*: User who takes control of NameNode is a super-user; it means somebody started the NameNode which have fully access on HDFS data.
- *An executing MapReduce may use the host operating system interfaces* - Sometime execution of MapReduce demands access other tasks on the host OS, access local storage for instant Map output, but both executing on the same physical node.

III. LITERATURE REVIEW

Many organizations use big data applications to predict future scope, Hadoop cluster store the sensitive information about such organizations. As result Hadoop cluster require strong authentication and authorization with data protection such as encryption.

The Authors Pradeep Adluru, Srikari Sindhoori Datla, they present secure Hadoop architecture by adding encryption and decryption functions in HDFS in "Hadoop Eco System for Big Data Security and Privacy" [1], S. Ghemawat, H. Gobioff, and S. T. Leung: "The Google File System. Proc". of 2003 ACM Symposium [6] gives a brief description about the two different types of integrations called HDFS-RSA and HDFS-Pairing used as extensions of HDFS, these integrations provide alternatives toward achieving data confidentiality for Hadoop, The authors G. Asharov, Y. Lindell, T. Schneider and M. Zohner publish optimizations and efficient implementations of OT and OT extensions in the semi-honest model and propose novel OT protocol with security in the standard model and improve OT extensions with respect to communication complexity, computation complexity, and scalability in "More efficient oblivious transfer and extensions for faster secure computation" [2]. The Authors J. Dean and S. Ghemawat. MapReduce: They presents trusted computing technologies combined with the Apache Hadoop Distributed File System (HDFS) to address concerns of data confidentiality and integrity in "Simplified Data Processing on Large Clusters" [8].

IV. ENCRYPRION TECHNIQUES

The Advanced Encryption Standard (AES) [11] is formal encryption method adopted by the National Institute of Standards and Technology of the US Government, and is accepted worldwide. The AES encryption algorithm is a block cipher that uses an encryption key and a several rounds of encryption. A block cipher is an encryption algorithm that works on a single block of data at a time. In the case of standard AES encryption, the clock is 128 bits, 16 bytes in length. The term "rounds" refers to the way in which the encryption algorithm mixes the data re-encrypting it to ten to fourteen times depending on the length of the key. The AES algorithm itself is not a computer program or computer source code. It is a mathematical description of a process of obscuring data.

V. SECURITY MECHANISMS

Privacy and security of Big Data stored on distributed cloud storage became the critical concern of the current industry. To overcome the delays in storing Big Data on distributed cloud storage, we use public storage as a solution. However, using public storage will make the data vulnerable to transmission and storage. Therefore, there is a need for a security algorithm provides tradeoffs during time delay, security strength and storage risks with flexible key based encryption techniques.

VI. PROPOSED SYSTEM

We have proposed new technique for securing data at HDFS. There are privacy and security risks for the Hadoop Eco-system as the NameNode and the DataNode have the entire control over the data. The user has no control over the data and so, there is a need for establishing a trust [7] between the user and the NameNode, i.e. we are authenticating the user, so that not everybody can access the data. To make the system more secure, we need to implement encryption techniques on the data. Map Reduce does the processing and implementation of the random encryption techniques. Map Reduce is going to break the encryption/decryption process and speed up the process to ensure that the performance or scalability of the system is not affected.

A. Authenticate User

Here in Hadoop system, the NameNode and the DataNodes handle the data for its storage and access. The user has no control over handling the data and so, authentication system is established. Not every User should be given the privilege of accessing and storing the data. The User should authenticate himself to the NameNode before he is granted access. Hashing techniques are implemented to achieve the authentication. The hashing technique used in this system is SHA 1.

C. Encryption Technique

In Hadoop system, the entire data is broken down into chunks of 64MB and distributed on the cloud. The secure encryption technique implemented in this system is AES [10]. The encryption and decryption of the data are handled by the Map Reduce functions of the Hadoop system. Multiple Map and reduce functions are assigned to speed up the encryption and decryption by breaking the independent responsibilities for different Map functions and then the Reduce function brings all the encrypted/decrypted data together. Fig. 2. shows flow chart of proposed system.

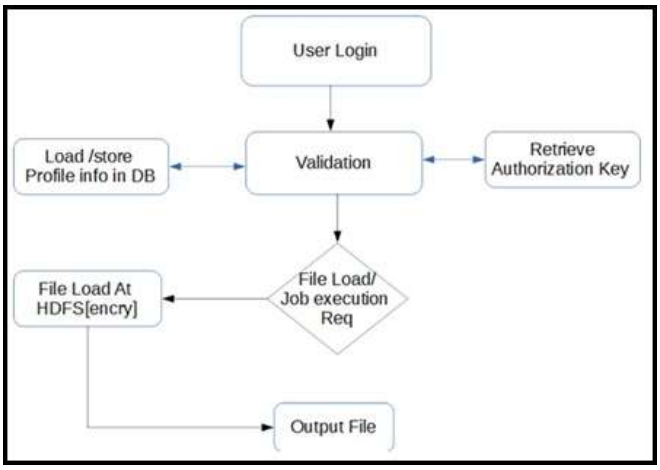

Fig. 2 Flow Chart

VII. EXPERIMENT SETUP

To do the experiment we have installed windows 10 OS on our machine [10]. After that we installed Openjdk1.8 and Apache Tomcat. We configured Hadoop 2.0.3 as a Single-Node Cluster to use the HDFS and MapReduce capabilities. A database table

ISSN: 2321-8169
32 – 36

is created in the MySQL 5.0(Wampserver). For User login, a home page(HES) is made for user authentication in the IDE(java). Here two button for starting dfs and yarn (for this the bashrc file link is connected) are there in home screen. Not only that there is signup page for new user. This information of user is stored in database table (using JDBC-ODBC connectivity from IDE). The hash value for the user is created when user register to it. At the same time user account information is stored in the NameNode of HDFS. As the user login, the user individual account opens. here the following options can be accessed by the user:

Store file: store the file to database for encryption.
List file : shows the files in user database.
Load file: upload the encrypted file to the HDFS.
Share file: here the user can share the file with other user by giving the private key access.

The MapReduce programs (Hadoop job) which take the encrypted data as input and execute job, we observed that it took 54.0490 seconds to execute a Wordcount MapReduce job for the unencrypted HDFS (normal execution) for size of 10MB test file, while it took 101.2780 seconds for the encrypted HDFS using AES.

VIII. TEST RESULT

The implemented system is successfully establishing trust between user and NameNode. Only authenticated users are given access to the information/data. Even if the security of the system is breached the access of the system for the information needs authentication of the user and so not everyone has the privilege of system access. The Fig. 3. depicts the use of encrypted keys to authenticate the user. Only the authorized user will have a unique private key generated by the user by using the hashing SHA-1 algorithm technique.

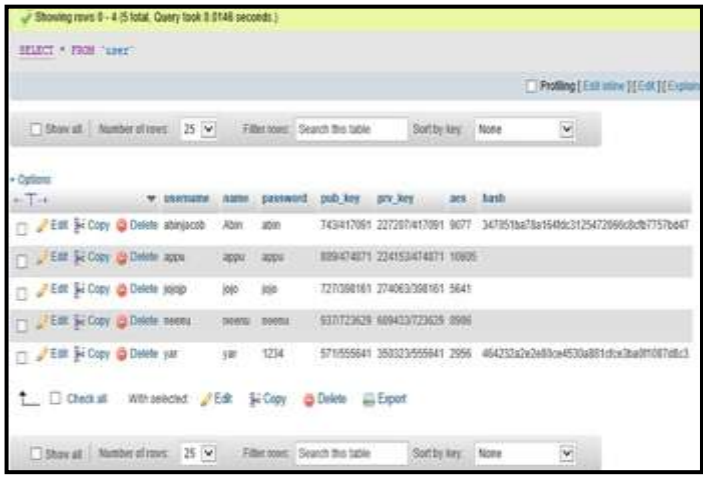

Fig. 3. Database Table

Table 1 shows the Comparison of time between AES and Normal file.

TABLE I
COMPARISON OF TIME BETWEEN NORMAL FILE AND AES ENCRYPTED FILE

Data (MB)		Data size (MB)	Time taken	Time taken to execute job(sec)
1	Normal file	1.055	12.15	22.05
	AES	1.23	46 (after encryption)	54
10	Normal file	10.72	131.55	54.23
	AES	12.1	298.09 (after encryption)	101.05

The results of data uploads of plain file and encrypted file is shown in graphs. The job execution time Comparison between AES encryption and the Plain file is also shown in table. The results of the tests are shown in graphs (Fig. 4-5).

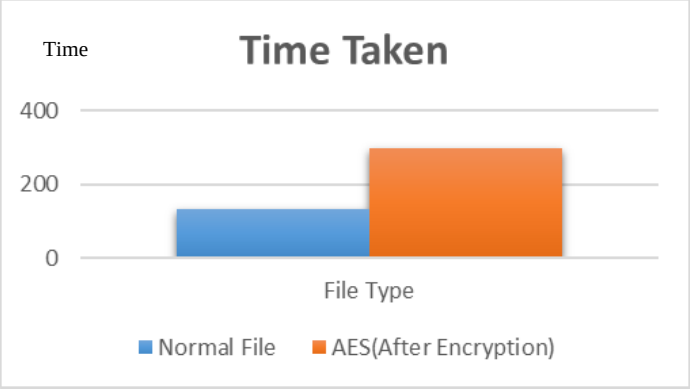

Fig. 4 Graph of time taken to Upload file to HDFS

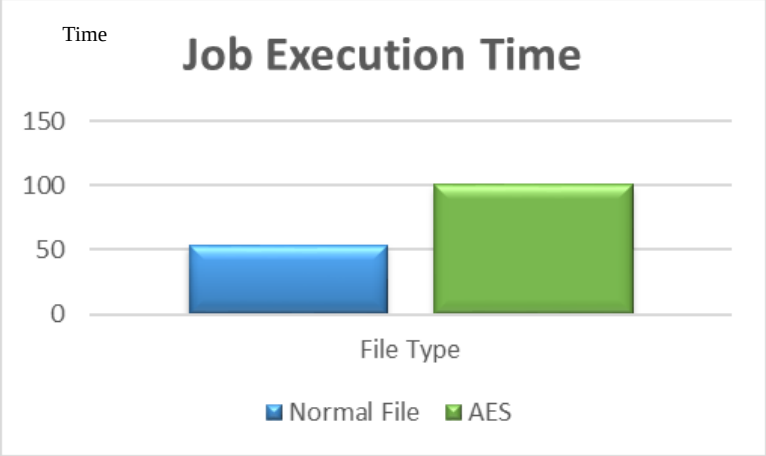

Fig. 5 Graph of comparison of job execution time

IX. CONCLUSIONS

To achieve privacy and security of Big Data the distributed data access and storage on the cloud is authenticated and secure random encryption techniques are implemented to make the system even more secure. Authentication followed by encryption techniques are implemented to make the system ironclad while maintaining the performance standards. Access to these huge volumes of private data might be very damaging when misused and so securing the system to the highest level is the priority

ACKNOWLEDGMENT

I owe my deep regards and gratitude to my project guide, M. Sudheep Elayidom for the constant encouragement and support in all forms for developing and maintaining the IEEE LaTeX style files which have been used in the preparation of this template.

REFERENCES

[1] *Pradeep Adluru, Srikari Sindhoori Datla, Xiaowen Zhang. "Hadoop Eco System for Big Data Security and Privacy ", Sponsored in part by a PSC-CUNY Research Award. 2015 IEEE.*

[2] *G. Asharov, Y. Lindell, T. Schneider and M. Zohner. More efficient oblivious transfer and extensions for faster secure computation. CCS'13, Nov 4-8, 2013.*

[3] *Apache Hadoop. http://hadoop.apache.org/, 2012.*

[4] *P. K. Mantha, A. Luckow, S. Jha. Pilot-MapReduce: An Extensible and Flexible MapReduce Implementation for Distributed Data. Proc. of 2012 Int. Conf. on MapReduce and Its Applications, pp. 17-24.*

[5] *V. G. Korat, A. P. Deshmukh, K. S. Pamu. Introduction to Hadoop Distributed File System. Int. J. of Engineering Innovations and Research, 1(2): 230-236, March 2012*

[6] *S. Ghemawat, H. Gobioff, and S. T. Leung: The Google File System. Proc. of 2003 ACM Symposium on Operating Systems Principles (SOSP), October 2003, Bolton Landing, NY, pp. 29-43.*

[7] *N. Miloslavskaya, M. Senatorov, A. Tolstoy and S. Zapechnikov. Big Data information security maintenance. SIN'14, Sept 09-11, 2014*

[8] *J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM, 51(1): 107-113*

[9] *L. Xu, X. Wu and X. Zhang. CL-PRE: a certificateless proxy re-encryption scheme for secure data sharing with public cloud. Proc. Of 2012 ACM Symposium on Information, Computer and Communications Security (ASIACCS'12), May 2-4, 2012, pp.87-88.*

[10] Apache Hadoop for Windows Platform"- https://www.codeproject.com/Articles/757934/AHadoop-for-Windows-Platform Java Tutorials. (http://www.w3schools.in/java-tutorial/)

[11] *Advanced Encryption Standard, http://en.wikipedia.org/wiki/Advanced Encryption Standard.*