Security System For Hadoop Distributed Storage

[1]Chaitanya P. Garware,[2]Shweta A. Joshi
[1] Student ME(Computer),[2]Prof. ME(Computer)
[1]Department of Computer Engineering
[1]Flora Institute of Technology, Pune, India.

***Abstract*–In various scientific and social domains, such as Life Science, Nuclear Physics, High Energy as well as Materials and Chemistry, Data Explosion is observed. In today's era, security of distributed network is very important.It is proposed that, the existing security problems of the distributed network will be solved by this Encryption Technique for Distributed storage using Hadoop.Selective encryption scheme is given on the basis of different confidential level of user data. It gives full consideration to the some security issues like the security of user data transmission in the network, no verification over user data and the leakage of user data privacy etc. More secured, effective and stable performance of Hadoop and distributed security storage can be supplied with the combination of symmetric encryption algorithms such like identity authentication AES and Blowfish which has rapid encryption speed.Key distribution will be placed using Apache Ranger Tool for the faster as well as secure access.**

IndexTerms– Hadoop, Apache Ranger, Blowfish, AES Algorithm.

I. INTRODUCTION

A computer network which stores the information on one or more than one node i.e. often in replicated fashion, is called as a distributed data store. It is specifically used to refer two things, first is either as a Distributed Database i.e. the number of different nodes contains information stored by user, or second is the Computer Network in which users store information on a different network nodes. Apache Hadoop is the open-source software framework fordistributed storage and distributed processing of very large data sets on computer clusters and it is written in java. The modules in Hadoop are designed with a fundamental assumption that hardware failures are very common to occur and these should be automatically handled by the framework. The Apache Hadoop consists of a core storage part, known as Hadoop Distributed File System (HDFS), and a processing part of Apache Hadoop is called MapReduce[1]. The Hadoop performs important task like Splitting of files into large blocks and then distributing them across nodes in a cluster. During processing of data, the Hadoop transfers packaged code for nodesto process in parallel manner which is based on the data that needs to be processed. Hadoop is a distributed system, so it allows us to store big data and it supports for processing of the data in parallel environment

Today, one of the biggest concerns revolves around the security and protection of sensitive information [2]. So, security in this process is becoming increasingly more important. The more data you have, the more important it is that you protect it.

II. RELATED WORK

Hadoop is the Apache open source project. It consists of HDFS, MapReduce, HBase, Hive and other projects [3]. Hadoop contains two major parts i.e. HDFS and MapReduce. MapReduce deals with tasks with a large scale,it aims at paralleling. So, the MapReduce scheduler becomes particularly important [4]. HDFS is the open source project of Google distributed file system (GFS) [6].

The authors Seonyoung Park and Youngseok Lee presented secure Hadoop architecture. They added encryption and decryption functions in HDFS in "Secure Hadoop with Encrypted HDFS", J.J. Park et al. (Eds.): GPC 2013, LNCS 7861, pp. 134–141, 2013 Springer-Verlag Berlin Heidelberg. They publish secure HDFS by adding the AES encryption/decryption class in to the CompressionCodec in the Hadoop

"Encryption Schemes and Hadoop Distributed File System", 26th IEEE International Conference on Advanced Information Networking and Applications, 2012 Springer, Verlag Berlin Heidelberg. They have published two different integrations as extensions of Hadoop Distributed File System (HDFS), first is the HDFS-RSA and second is the HDFS- Pairing.

The authors Weimin Xu, Wenfeng Shen, Jiwei Jiang and Thanh Cuong Nguyen have presented the scheme to encrypt user's data before transferring to HDFS if he requires high privacy. "A Novel Data Encryption in HDFS", IEEE International Conference on Green Computing and Communications and IEEE Cyber, Physical and Social Computing, 2013. While file is being uploaded, they provided novel method for providing the encryption.

A. Hadoop Architecture

In a distributed environment, open-source framework which allows to store and process big data across clusters of computers using simple programming models is called as Hadoop[7]. It scales up from single server to thousands of machines

The following figure can be used to depict these four components available in Hadoop framework

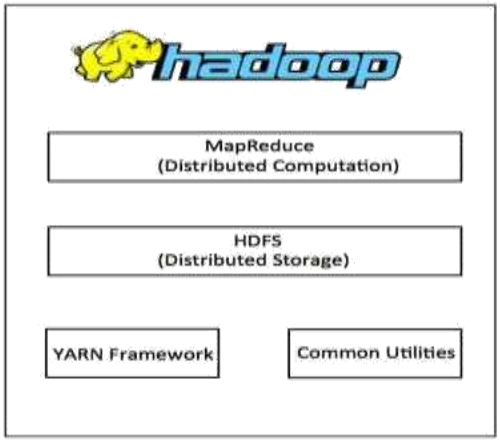

Figure 1. Overview of System Components of Hadoop Framework

Following four modules are contained in Hadoop Framework.

Hadoop Common:

It contains Java libraries and utilities required by other Hadoop modules as well as scripts to start Hadoop. The Hadoop. Libraries provide file system and OS level abstractions.

Hadoop YARN:

YARN stands for Yet Another Resource Negotiator. This framework is used for job scheduling and cluster resource management.

Hadoop Distributed File System (HDFS):

It is a distributed file system which provides high-throughput access for application data.

Hadoop MapReduce:

Hadoop MapReduce is YARN-based system which works for parallel processing of large data sets

The term ”Hadoop” also refers to the collection of additional software packages that can be installed on top of Hadoop such like Apache Pig, Apache Hive, Apache HBase, Apache Spark etc

For a wide variety of applications, nearly across all the industries, MongoDB is used. MongoDB is the open-source database used by different companies with variable sizes. Nearly 40,000 organizations running MongoDB are found unprotected and vulnerable to hackers. MongoDB is built for increasing scalability, performance and to achieve high availability. It is also useful for scaling from single server deployments to large, complex multi-site architectures. MongoDB provides high performance for both reads and writes by leveraging in-memory computing. The German researchers were able to get ”**read and write access” to the unsecured MongoDB databases** without using any special hacking tools. They found **39,890 MongoDB databases** openly available over the Internet

This incident seriously pushes Hadoop to recommend more strong security mechanism to avoid such unauthorized and disastrous vulnerable attacks on Big Data. To provide more strong security mechanism on Hadoop, authentication method is provided as follows

Kerberos: The secure Hadoop is nothing but a deployment of Hadoop in the environments where Kerberos has already been enabled for providing strong authentication. After configuring the Kerberos, client-side credentials are validated by using Kerberos authentication. Service Ticket requested by client should be valid for the Hadoop environment. The Service Ticket is submitted as a part of the client connection. Kerberos provides authentication through exchange of tickets between client and server. Here, validation is provided by a trusted third party in the form of the Kerberos KDC i.e. Key Distribution Center

III. PROPOSED MODEL

Proposed model is given as in the form of architecture as well as flow chart. In proposed model, the data transfer from or to Hadoop system will be in encrypted manner using Blowfish algorithm and AES algorithm and Key distribution will be placed using Apache Ranger Tool for the faster as well as secure access. So, more enhanced security can be provided over the Hadoop.

A. Architectute

The architecture explains the behaviour of client request processing. It includes the launching of the request from client and different encryption choices provided to client as per its requirement or convenience

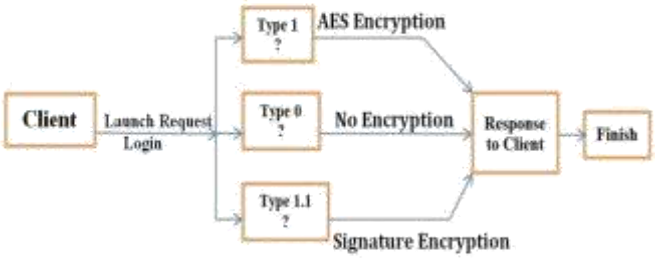

Figure 2. Architecture Model

Architecture model contains the request for access in three different manners:

Type 0: It directly calls to Hadoop function and client gets the response. It has no security. If the user doesn't want any encryption, then Type 0 option is selected with no encryption.

Type 1: The AES key is generated and the more strong security is provided in this Type 1.It creates the key based on encryption algorithm and allows data access in secure manner. If the user wants data integrity at high level, then this option can be preferred for the encryption.

Type 1.1: Signature is matched to get access to authorized data. Here, Signature encryption technique is used. Matching signature allows the access for only authorized users

With the help of flow chart, we can give proposed model as explained in the following figure 3. It gives the detailed description of different ways of providing security to avoid the unauthorized access to the data. It describes the flow of the different security manners for providing encryption

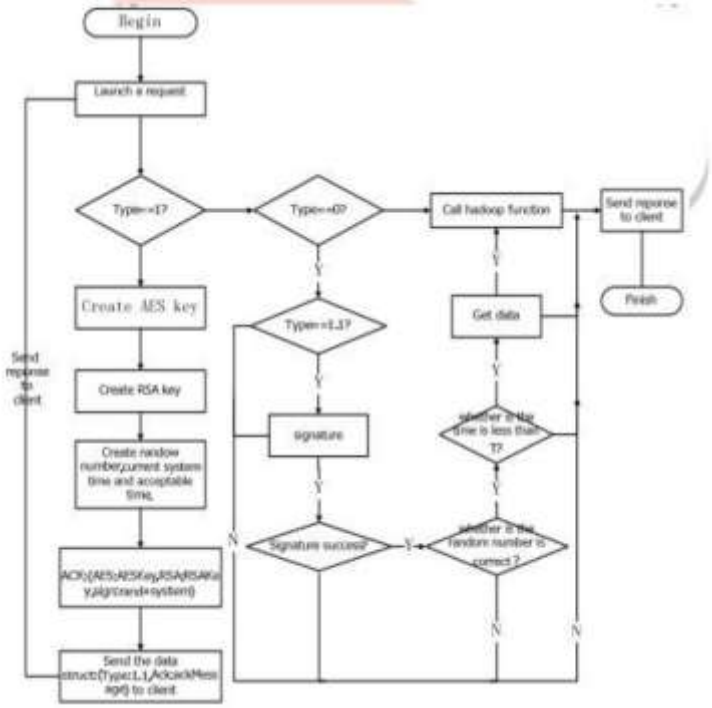

Figure 3: Proposed Model

IV. METHODOLOGY AND ALGORITHM

1. MATHEMATICAL MODEL

The mathematical model implemented in the proposed system is as follow:

1) Set K is the "Key space" and its elements are calledas the keys.
2) The rule by which each k€K is associated with a trap-door one-way function i.e. ek with domain (plaintext space) Mk and range (ciphertext space) Cipk.
3) The procedure for generating a random key k€K together with a trap-door d for ek and the inverse map Dk: Cipk → Mk such that:

$$Dk\ (ek(m)) = m;\ \text{for all } m€M$$

4) Key space i.e. K is also known as public-key space and set of trap-doors d is called as private key space. Relative to (3), it is also required that random keys k€K and their corresponding trapdoors d be easy to generate

The description of all the components (1)-(3) of a cryptosystem is public knowledge. A user (person) who wants to become a part of communication network can proceed as follows:

- To generate a random key k 2 K and the corresponding trap-door d, use (3).
- Now place the encryption function ek or equivalently the key k in a public directory (i.e. in the user's directory or home page). And keep d and decryption function Dk as secret.

Now consider that, the Bob wants to send a message m to user Alice. To perform this, he simply looks up her public enciphering function ekA and computes c = ekA(m) which he sends to Alice.
On receiving c, Alice Computes

$$DkA(c) = DkA(ekA(m)) = m;$$

Thereby recovering the message. An eavesdropper might intercept c and can obtain ekA from public files, but cannot find m from c without knowledge of dA (or equivalently DkA).

2. APACHE RANGER

For the Hadoop cluster, comprehensive approach to its security is given by Apache Ranger. It provides the central security policy administration including authorization, authentication, auditing and data protection. Apache Hive and Apache HBase are the centralized security frameworks provided by Apache Ranger. For individual users or groups, these policies can be set and then enforced within Hadoop.

3. THREE WAY HANDSHAKE

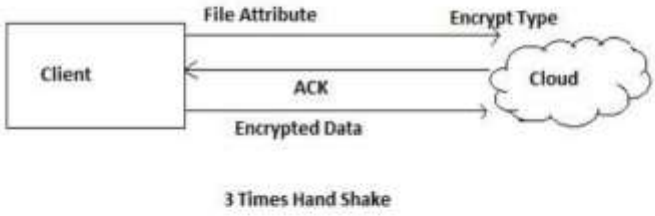

Figure 4. Three Way Handshaking

Three times handshake is generated between the client and server. It is the core part of our paper. Detail steps are as follows:

First hand shaking

If user wants to upload file on server, then first, he should initiate a request. The client needs a secret key to encrypt the file if the file is confidential. Secret key is generated by server and then it is sent to the client as it cannot be generated at client terminal.After this, client will finishsymmetric encryption of data and the data signature [9][12].

Second hand shaking

1) It receives the request which is sent from client.
2) Then the request is analyzed and it can get its confidentiality level.
3) Depending on confidentiality level,secret key production and distribution module are called to generate asignature key and symmetric key.
4) Now, firstly the generation of a random number " rand"is carried out. Now we get the current system time i.e. "Response_Current_Time". To get the signature number "Rand", we could use "rand" plus "Response Current Time" To check whether the encryption and transmission of the data is completed within a valid period or not, the Time factor 'T' is used.
The calculation of T is given as follows

$$T=S*C+TS+D$$

Where, S is Size of the file, C is Complexity of algorithm, TS represents Transmission time and D denotes acceptable Delay time. 5) After this, we begin to form a data structure named ACK which is shown below.

{{Rand, symmetric key, signature key, filename} encrypted using users master key}; Finally, ACK is sent to client and now wait for response

Third handshaking

As the core of this thesis is the three times handshake, this step turns out to be important.
Following are the detailed steps:
1) First, it receive request sent from the client.
2) Then, decrypt ACK using mast key. So now we can get the symmetric key, signature key and"Rand".

3) Now, we need to encrypt the data needed to be uploaded. So, forming the data structures which has to be sent to server.Details are as follows

{{user data} encrypt using symmetric key, {Rand} encrypt using signature key, filename: filename}

4) Here, sending the data structure to client and wait for the response.

5) It also needs to check that the whole time, encryption time and transmission time is carried out in a valid period. It is as follows:

(SystemTime-(Rand-rand))-T

If the value of above expression is greater than zero,it means thatentire process is completed in a suitable period. 6) If encryption and transmission process is safe, user's data can be stored in safe manner

4. THE BLOWFISH ENCRYPTION ALGORITHM

To replace DES algorithm, symmetric block cipher called as Blowfish can be used. It is used for both domestic and exportable use as it takes a variable-length key from 32 bits to 448 bits. Blowfish was designed by Bruce Schneier as a fast and free alternative for existing encryption algorithms. Blowfish is now gaining acceptance as a strong encryption algorithm. This is license-free and it is available free for all the users.

Blowfish Algorithm

It is designed by Bruce Schneier.
Block cipher: 64-bit block.
It is faster than DES.
Royalty-free.
Free source code available so no license required.
Variable key length: 32 bits to 448 bits.

5. AES ALGORITHM

In the cryptography, the Advanced Encryption Standard (AES) [14] is a block cipher algorithm. It is used as the encryption standard by the U.S. government. As it was the case with its predecessor, theData Encryption Standard (DES), AES has been lookedat a lot and it is now used all over the world.AESprovides very good defensesagainst the various attack techniques [13]

V. EXPERIMENTAL RESULT

Here, we will analyze the various security measures for distributed storage on HDFS and implement a tool that will do encryption of data using different algorithms like AES on the Hadoop Distributed File System. This algorithm will do three way handshake to avoid any tampering to the key used for encryption. Further to provide the more fine grained authorization, we will be using Apache Ranger to provide authorized access to intended user.

VI. ACKNOWLEDGEMENTS

I hereby take this opportunity to record my sincere thanks and heartily gratitude to Guide Prof. Shweta A. Joshi useful guidance and making available to me his intimate knowledge and experience in making **"Security System for Hadoop Distributed Storage"** as a project and I express my sincere gratitude towards people who have worked in this area.

VII. CONCLUSION

To overcome differentkinds of serious issues like MongoDB incident, we try to design the security framework. The data transferfor Hadoop system will be in encrypted manner using Blowfish and AES algorithmsand the key distribution will be placed using ApacheRanger Tool for the authorization. It will provide faster as well assecure access. So, the enhancement of Hadoop system can be achieved in more secure manner. In thefuture version of system, we can try to implement this security on G-Hadoop. In this way,the security framework is designed for different clusters in a distributed environment. So, itreduces the burden on the server and finally achieves secure storage of database.

REFERENCES

[1] Bingsheng He, Wenbin Fang, QiongLuo, Naga K. Govindaraju, Tuyong Wang, Mars: "MapReduce framework on graphics processors", in: Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques, PACT08, ACM, New York.

[2] Xu Tao, Fu Ge, Tan Huaiyuan, Zhang Hong and Liu Xinran, "Thump Storage: A Management and Analysis System for Structured Big Data", International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), pp. 2424 - 2427,2013

[3] XU Guang-hui, Deploying and researching Hadoop in virtual machines[C]//Automation and Logistics (ICAL), 2012 IEEE International Conference on Zhengzhou, 2012: 395-399.

[4] ZHUO Tang, ZHOU Jun-qing, LI Ken-li, et al. Contact Administrator; MTSD: A Task Scheduling Algorithm for MapReduce Base on Deadline Constraints [C]. Parallel and Distributed Processing Symposium Work.Shanghai, 2012: 2012-2018.

[5] WANG Su-li. "File Encryption and Decryption System Based on RSA Algorithm[C]". Computational and Information Sciences (ICCIS), 20, Chengdu, China, 2011: 797-800.
[6] Ghemawat S,Gobioff H,Leung S T.The Google file system[C].Proc of the 19th ACM Symp on Operating Systems Principles. NewYork: ACM,2003:29-43.
[7] Jason Cohen and Dr. Subatra Acharya "Towards a Trusted Hadoop Storage Platform: Design Considerations of an AES Based Encryption Scheme with TPM Rooted Key Protections" (2013).
[8] CAO Ying-yu."An Efficient Implementation of RSA Digital Signature Algorithm[C]". Intelligent Computation Technology and Automation Hunan 2008: 100-103
[9] Mahendra S. Patil, Jinesh K. Kamdar and Chintan B. Khatri, "Big Data – An Overview", International Journalof Engineering Research Technology (IJERT), Vol. 3 Issue 7, July 2014.
[10] Globus, "Overview of the Grid Security Infrastructure", 2014.
[11] Demchenko, Y., et al. "Architecture Framework and Components for the Big Data Ecosystem"(2013).
[12] Kan Sanjay, Ram and Christopher "Hadoop Security Design Owen", October 2009
[13] Songchang Jin, Shuqiang Yang, Xiang Zhu, and Hong Yin "Design of a Trusted File System Based on Hadoop" 2013.
[14] Seonyoung Park and Youngseok Lee "Secure Hadoop with Encrypted HDFS".