SECURITY FOR CLOUD DATA USING COMBINED TECHNIQUE IDENTITY AUTHENTICATION AND ENCRYPTION

Kale Rohitkumar Bhausaheb
Computer Department, Amrutvahini College of Engineering Sangamner,
Savitribai Phule Pune University

Prof Paikrao R.L.
Head of Computer Department, Amrutvahini College of Engineering Sangamner,
Savitribai Phule Pune University

ABSTRACT

Hadoop has gain a lot of importance in modern world, hadoop has been extensively used for storage of cloud data. Hadoop distributed file system (HDFs) used as a architecture for storing data on cloud storage. But there are very fewer attempts have been made to verify the group membership of users, which deals interaction with HDFS. Those who have created log in before use can easily store data in hadoop, directly in browser history without any additional storage media. This promptly facilitates browsing of data for a user from any computer, tablet, mobile etc.. Major drawback of this system unauthorised user to access this confidential cloud data. Many attempt is been made to improve security. One of them is hadoop kerberose authentication is used for security of data but according to survey this is not very effective method for securing confidential data. By using two systems in combination we can ensure security of the network to additional extent.

INTRODUCTION

Hadoop uses a architecture of HDFS, a distributed file system based on google file system (GFS), as base of this shared file system. Implemented architecture helps in dividing file into large chunks say approx (~64MB) and these chunks are distributed across data servers.

But so far no attempt has been made to verify or identify, group membership of users who try to attempt an interact with hadoop architecture.

Cloud computing is a virtual technology software works with various technologies of software.The role of hadoop in cloud computing system is to store data on a different devices. The method used for experiment here is based on hadoop kerberoes authentication. Author has trying to provide security to mainly four functions

Data transmission: data transmission may be intercepted but using encryption technique we avoid such kind of interception.

Access control: In Cloud User store data without setting any access authority to data so there is problem of user lost his absolute right to monitor.

***Data Confidentiality and Data storage**:* In Cloud storage Hadoop there is not any classification of Confidential and Non-confidential data so maybe there is problem of data lost. Also data store is in distributed manner we cannot get actual location of data.
Data verification: In cloud there no any identity verification on uploading data. That means when data uploaded on cloud it comes from right user or not.

LITERATURE SURVEY

Hadoop is open source project, and open source project are always vulnerable to security. Hadoop has been developed by apache as a part of open source project [1]. The major parts of this system are architecture main comprises of HDFS and map reduce. Map reduce is been used to paralleling and dealing with large number of tasks at a time [2]. The map reduce possesses and main advantage of high fault tolerance and certain data access control. The hadoop is been used by yahoo in 2009 with safety measures of Kerberos. The users have to obtain access certification from the third party to gain access to hadoop cluster first. The advantage of doing this is, it reduces the risk of users' data fraud. Many researchers have implemented different methods to avoid fraud while using cloud storage. The hadoop implemented for private enterprise gives more security than global cloud servers. The many scheme ensured the safety data storage [5]. Symmetric encryption and asymmetric encryption features are complementary to each other in nature [9]. The mentioned methods in literature survey just works on encrypt the data or identification of the user from only one perspective for a single demand. But no extensive or comphrensive method is implemented for data protection.

IMPLEMENTATION

In our System having Two Modules:

1] **Client** :
a. Transmission Module.
b. Data Encryption Module.
2] **Server** :
a. Secret Key Production Module.
b. Data Decryption Module.
c. Data Authentication Module.
d. Data Compression Module.

Concept implemented here uses a three way handshaking between client and server. For that client and server meet each other three times as presented below.
Type: 0 means: Uploaded file want to be **not Confidential** then **no need of Encryption**.
Type: 1 means: Uploaded file want to be **Confidential** then **need of Encryption**.

1] : ***<u>3 way Hand Shake Technique</u>***:

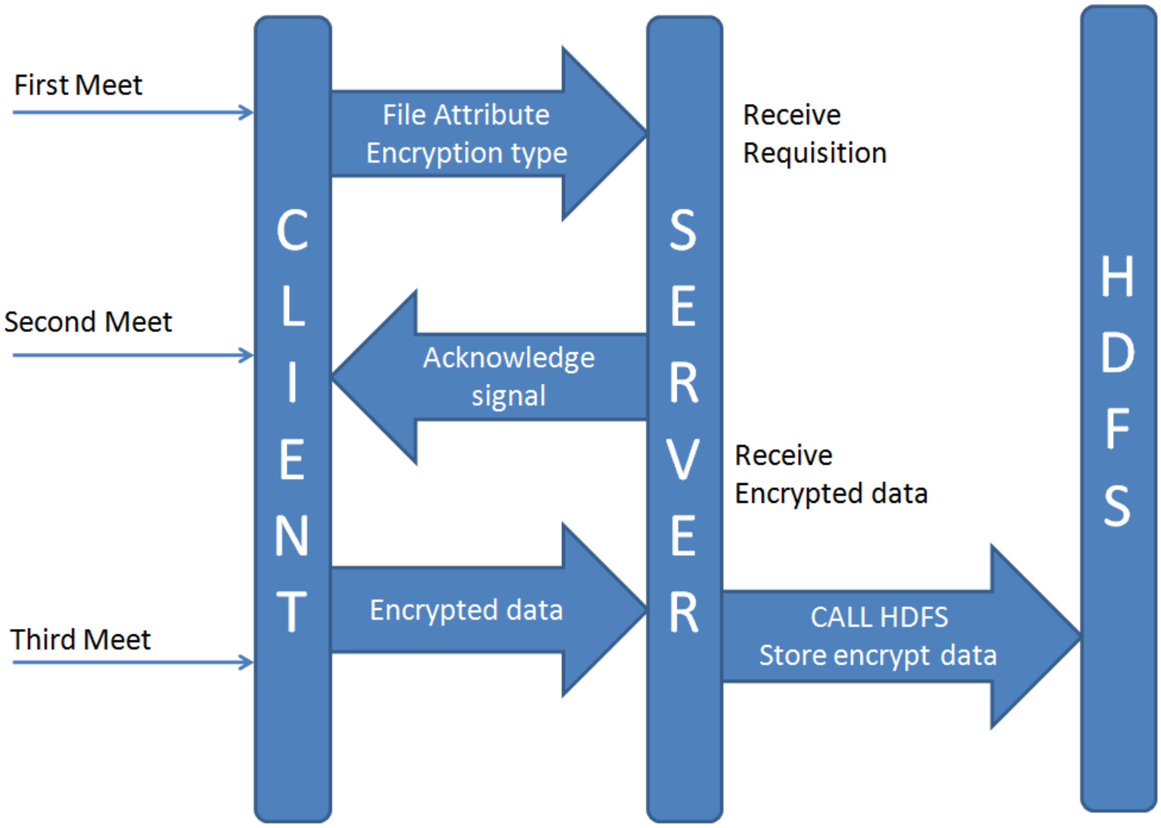

Fig 1: Three Meet between Client and Server during data transmission.

1.1] <u>FIRST MEET / FIRST HANDSHAKE</u>:

1) Check confidential of file.
2) Choose the encryption method (0 or 1)
3) After that Generate data structure according to the size of the file and the value of type. Type's value stands for the confidentiality level of the file. If the value of type is 0, the attribution of file is unsecured and the file can be uploaded to the cloud server directly.

   Structure: {TYPE: **0 or 1**, SIZE: **file Size**, FILENAME: **filename**};

4) Send the data generated in step3 to the server and wait for response.

1.2] SECOND MEET / SECOND HAND SHAKE :

1) Receive request sent from the client.
2) Analyses the request and get the confidentiality level.

3) According to the confidentiality level call the secret key production and distribution module to generate symmetric key and signature key.
4) Generate the signature key in step3; this step is utilising a signature. Firstly, generation of random cumber ‘rand’ and get the current system time ‘RESPONSE_CURRENT_TIME’, then we can combine two signals ‘rand’ in addition with ‘RESPONSE_CURRENT_TIME’ to get the signature number ‘rand’. Integration with time factor ‘T’ which is used to check encryption and transmission of the data is completed within a valid period. The implemented system calculate T is given below

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

   Where S denoted the size of the file, c corresponds to complexity of the algorithm, TS is abbreviation of the transmission time and D represents the acceptable delay time.

5) After step d, we could begin to form a data structure named ACK which is shown below.

   {{Rand, symmetric key, signature key, filename}
   Encrypted using user’s master key};
6) Send Acknowledge to client and wait for response.

1.3] <u>THIRD MEET / THIRD HAND SHAKE</u>:

1) Receive request sent from the client.
2) Decrypt ACK using mast key and then get the symmetric key, signature key and ‘Rand’.
3) We need to encrypt the data needed to be uploaded.
   Firstly, we would call the data encryption module to encrypt the uploading data using symmetric key, then data signature module also would be called to encrypt 'Rand' using signature key. Now it is time for forming the data structure which would be sent to the server. Details are as follows:

   {{user data} encrypt using symmetric key,{Rand}
   encrypt using signature key , filename: filename}
4) Send the data structure to client and wait for the response.
5) When data transfer occurs at the end of the server, verification and validation of data being stored in the HDFs. Data authentication module is verifies the signed data. However, if it is successful the server can make sure that the data received is packaged by right user at a time. While signature verification is being done, it also needs to check that the whole time, encryption time plus the transmission time, is carried out in a valid period. Its calculated by below method.

   (System Time-(Rand-rand))-T

   If the value of this expression is greater than 0, the entire process is completed within a suitable period, we believe that the security of the data is reliable.
6) If the entire authentication are passed as per given in step 5, we will have to ensure the entire encryption and transmission process is sage, and then only user data could be stored in HDFS.

EXPERIMENT

In implemented model, distributed encryption for checking performance, there are many cloud technologies which is not encrypt for checking performance. Two symmetric encryption techniques like blowfish and AES algorithm. Our experiment compares the distributed encryption with data encryption on the clouds, and the comparative parameter is encryption spent time.

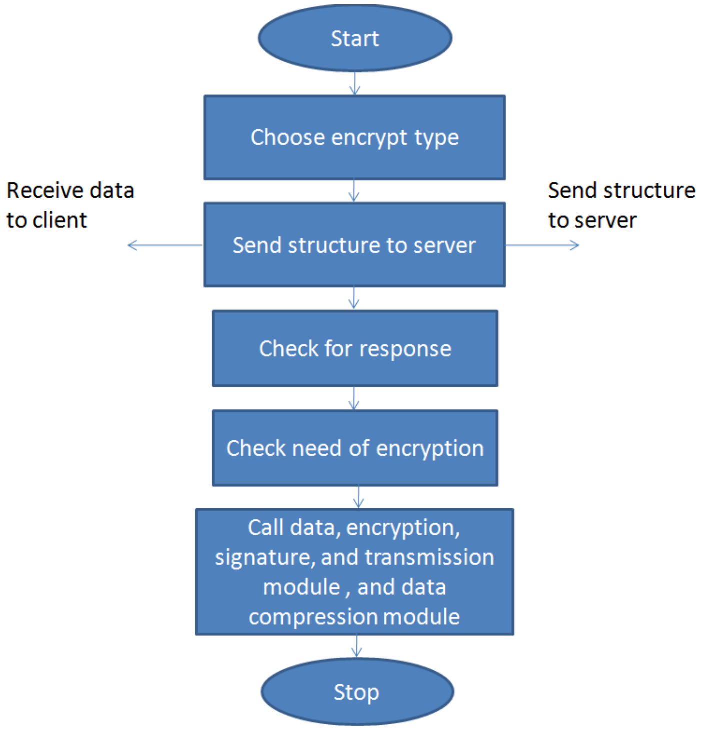

Fig.2.: Flowchart for client module

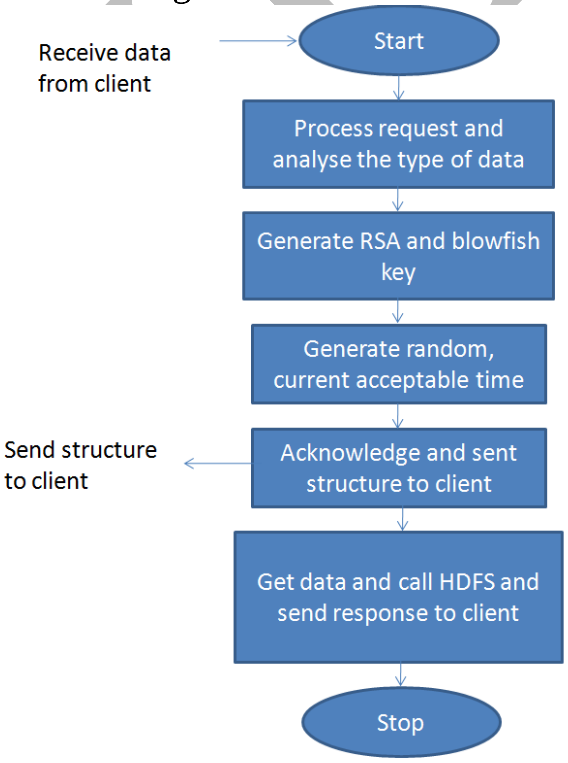
Receive data from client
Start
Process request and analyse the type of data
Generate RSA and blowfish key
Generate random, current acceptable time
Send structure to client
Acknowledge and sent structure to client
Get data and call HDFS and send response to client
Stop

Fig.3.: Flowchart for server module

RESULT

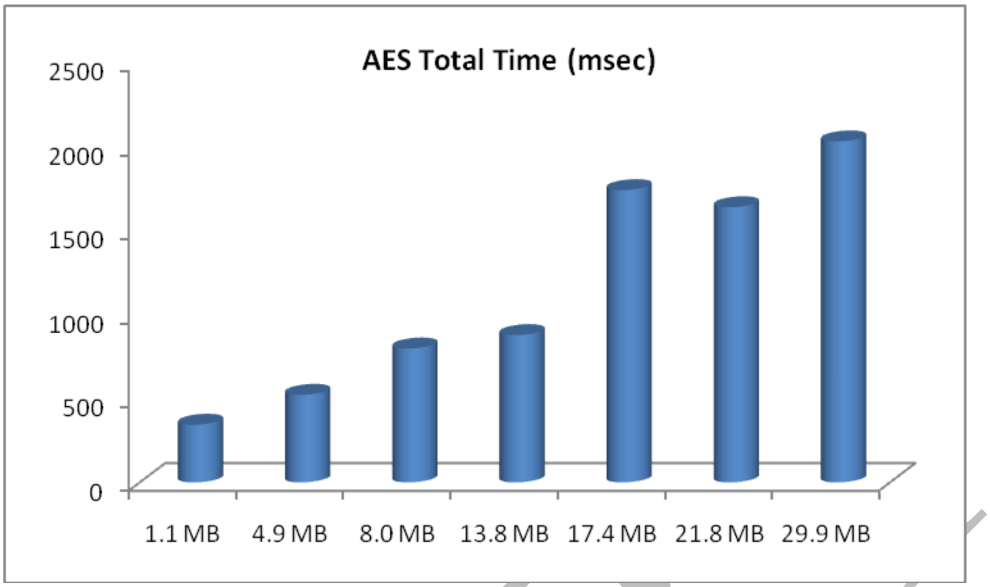

Fig 4 : chart for showing time required for encryption and uploading using AES encryption technique.(single node cluster)

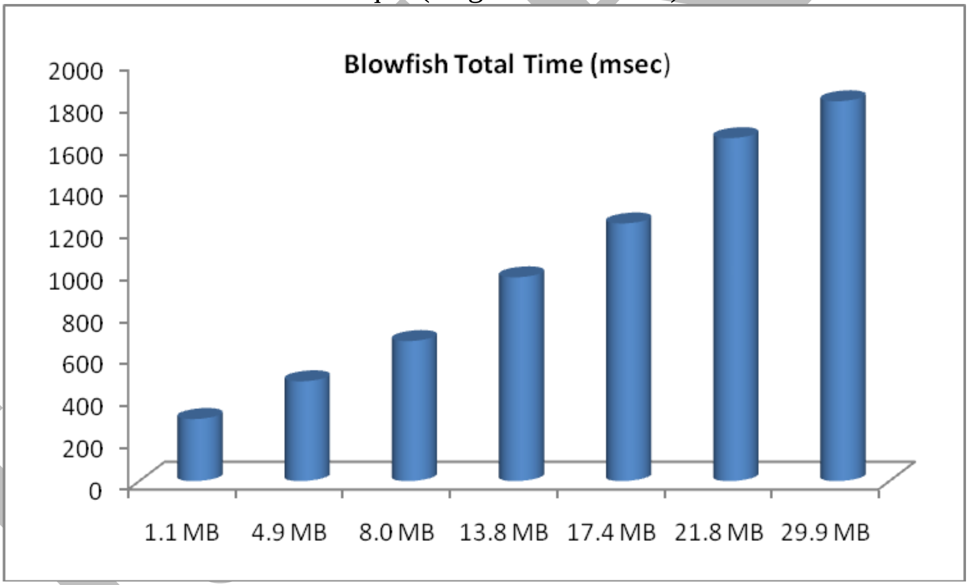

Fig 5: chart for showing time required for encryption and uploading using AES encryption technique (single node cluster)

For using encryption techniques two methods have been method. And as per the analysis on the basis of performance blowfish is faster encryption techniques is better than AES. Disadvantage of blowfish method is it possesses the block size of 64 bit, thus it becomes problematic after some GB of data encrypted with the same key. So in today's world AES is been used most extensively and accepted as standard algorithm for data transmission worldwide.

For each byte enciphered, and unless miscounted:

- Blowfish uses an average 8 table lookup and 10.25 straight 32-bit operations.
- AES-128 performs on average 10 table lookups and 10.25 straight 32-bit operations (when using a common optimization also requiring 4 Kbyte of tables).

Table lookups are much more expensive than straight 32-bit operations (most of these come almost for free in practice), and dominate the cost. Thus yes, Blowfish can be a little faster than AES implemented in software; especially if the AES implementation is not optimized to the max, or is AES-256.

Other than that, Blowfish and AES are not playing in the same league, and AES often wins without battle:

- Blowfish is a 64-bit block cipher, while AES is a 128-bit block cipher; this is a serious issue in a growing number of applications.
- Blowfish key setup is a *slow* process that produces 4 kByte of table per instance, in RAM. By contrast AES can be implemented efficiently (at least in hardware) with no RAM, and (for encryption, which is the most used) no pre-computation at all (decryption requires a tiny pre-computation).
- The 4 kByte table (or 1 kByte, 512 bytes, or even 256 bytes with speed trade-of) used for fast software AES implementation can be shared between instances (or in ROM/flash), it can't in Blowfish.

That makes Blowfish a terrible algorithm when 64-bit block is an issue (e.g. CBC mode with gigabytes of data); or when key agility matters (on occasions the relative slowness of Blowfish's key setup is an asset, see bcrypt); or for hardware implementation; or when 4 kByte is a lot of RAM (e.g. some embedded systems, or a server handling many simultaneous connections).

CONCLUSION

The implemented model is a security encryption schemes based on Hadoop, the weaknesses of the existing security system on cloud data storage is been addressed in great detail. Moreover improved method is suggested here to enhance storage security, stability, efficient and effective storage. The implemented system more sophisticated approach is required in future for data encryption and authentication.

REFERENCES

[1] A. Huang Jing. LI Renfa,C. Tang Zhuo, The Research of the *Data Security for Cloud Disk Based on the Hadoop Framework* . in 4th ICICIP Beijing, China ,2013)

[2] HongBo Zhou. Cloud computing: technology, application, standard Electronic Industry Press.2011.

[3] Hou Qinhua,Wu Yongwei,Zheng Weimin “A *Method on Protection of User Data Privacy in Cloud Storage Platform*” .in Journal of Computer Research and Development, pages 1146-1154 2011.

[4] MUNIER M.”Self-Protecting Documents for Cloud Storage Security[C].”Trust, Security and Privacy in Computing and Commu . Liverpool,2012: 1231-1238.

[5] SHAIKH F B.” Security threats in cloud computing [C]”.Internet Technology and Secured Transactions (ICIT. Abu Dhabi,2011: 214-219).

[6] V. Winkler, “Securing the Cloud Computer: Security Techniques and Tactics,” Elsevier Inc., ISBN: 978-1-59749-592-9, 2011.

[7] ZHANG Da-wei. Research on hadoop-based enterprise file cloud storage system[C]”.Awareness Science and Technology (iCAST), 2011 3rd. Dalian, 2011: 434-437

[8] BAO Rong-chang. Access Security on Cloud Comput- ing Implemented in Hadoop System[C].Genetic and Evolutionary Computing (ICGEC),2011.

[9] LIU K.The Security Analysis on Otway-Rees Protocol Based on BANLogic[C].Computational and Infor -mation Sciences (ICCIS), Chongqing, 2012: 341-344

[10] http://en.wikipedia.org/wiki/Virtualization

[11] http://www.cse.wustl.edu/jain/cse567-6/ftp/encryption-perf/.