International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

NOVEL APPROACH TO SECURE DATA IN HADOOP DISTRIBUTED FILE SYSTEM

*ARUN BABU KARUMANCHI **M.MANASA *** D.RAJESH
*ASST.PROFESSOR Dept Of CSE Bapatla Engineering College,(BEC), Guntur
**BTECH Dept Of CSE Bapatla Engineering College,(BEC), Guntur
***BTECH Dept Of CSE Bapatla Engineering College,(BEC), Guntur
arunkarumanchi123@gmail.com muddana95@gmail.com dammurajesh9401@gmail.com

Abstract:- Hadoop is most popularly used distributed programming framework for processing large amount of data with Hadoop distributed file system (HDFS) but processing personal or sensitive data on distributed environment demands secure computing. Originally Hadoop was designed without any security model. In this project, security of HDFS is implemented using encryption of file which is to be stored at HDFS. For encryption a real-time encryption algorithm is used. So a user who has the key for decryption can perform decryption of data & access that data for data mining. User authentication is also done for the system. We have also compared this method with the method previously implemented i.e. encryption & decryption using AES. Encrypting using AES results into growing of file size to double of original file &hence file up load time also increases. The technique used in this project removes this drawback. We have implemented method in which O Auth does the authentication and provide unique authorization token for each user which is used in encryption technique that provide data privacy for all users of Hadoop. The RealTime encryption algorithms used for securing data in HDFS uses the key that is generated by using authorization token.

***Key Words*:** Hadoop, Bigdata, Security, HDFS, OAuth.

1.INTRODUCTION

Hadoop was developed from GFS Hadoop is a framework of tools, implemented in Java. It supports running applications on big data.

1.1 ProjectIdea:

Hadoop is designed without considering security of data. Data stored at HDFS is in plain text. This data is prone to be accessed by unauthorized user. So method for securing this data is needed. Hence we are developing this highly secure system for Hadoop Distributed FileSystem.

1.2 Need ofproject:

Hadoop is generally executing in big clusters or might be in an open cloud administration. Amazon, Yahoo, Google, and so on are such open cloud where

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

numerous clients can run their jobs utilizing Elastic Map Reduce and distributed storage provided by Hadoop. It is key to execute the security of client information in suchsystems.

Web produces expansive measure of information consistently. It incorporate the organized information rate on web is around 32% and unstructured information is 63%. Additionally the volume of advanced substance on web grows up to more than 2.7ZB in 2012 which is 48% more from2011and now so a ring to wards more than 8ZB by 2015. Each industry and business associations are has a critical information about various item, generation and its business sector review which is a major information advantageous for efficiency development.

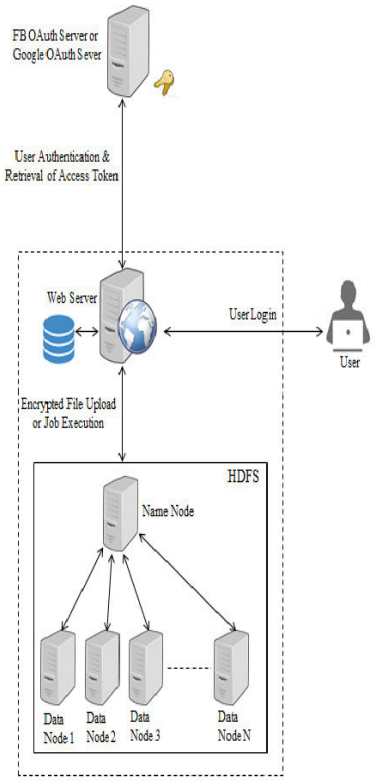


Fig-1:System Architecture

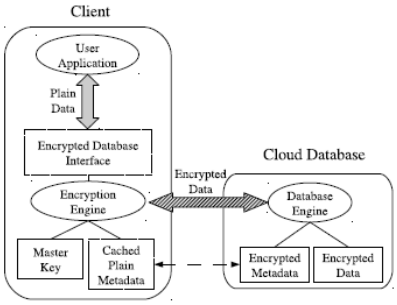
The files in Hadoop distributed file system (HDFS) are divided into multiple blocks and replicated to other Data Nodes(by default 2 nodes) to ensure high data availability and durability in case of failure of execution of job (parallel application in Hadoop environment).Originally Hadoop clusters have two types of node operating as master-slave or master-worker pattern [6]. Name Node is a master node and Data Nodes are workers nodes in HDFS. Data nodes are the nodes where actual file(part of file on a node) is stored. However Name Node contains information about where the different file blocks are located but it is not persistent, when system starts block may changes one Data Node to another Data Node but it report to Name Node or client who submit the Map Reduce job or owner of Data periodically [11]. Client gets list of data nodes where file blocks reside & then communicate with Data nodes only. Name Node contains only metadata. Our proposed system architecture is as shown infig-1.

2. RELATEDWORK

Hadoop is a distributed system which permits us to store enormous structured &un structured information (i.e. Big Data). It is also helpful to process such huge amount of data in parallel environment. Numerous associationsutilizeshuge information applications to fore see future degree,

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Hadoop group store the sensitive data about such associations (data like profitability, monetary information, client criticism and so forth.). As result Hadoop file system requires method to protect such information using very strong authentication. It also requires authorization of user. The technique described in [1] is a secure Hadoop architecturehereencryptionanddecryption functionsareappliedtotheHDFS.AESencrypt/decryptclassesareadded
for encryption and decryption of data.



The trusted computing technologies [2] combined with the Apache Hadoop Distributed File System (HDFS)in an effort to address concerns of data confidentiality and integrity. The two different types of integrations called HDFS-RSAandHDFS-Pairing[3]usedasextensionsofHDFS, these integrations provide alternatives toward achieving data confidentiality forHadoop.

Novel method used [4] to encrypt file while being uploaded. In this method, data which is to be uploaded to HDFS is first stored in a buffer. After that encryption is appliedtothebuffer'sdatabeforebeingsendingittoHDFS. This encryption is transparent to user. Thus, client needs not to stress over the information's privacy anylonger.

The homomorphic encryption technology [5] enables the encrypted data to be operable to protect the securityofthedataandtheefficiencyoftheapplication.The authentication agent technology provides various access control rules, which are defined using access control mechanisms, privilege separation and security audit mechanisms, to ensure the protection for the data that will be stored in theHDFS.

Theseaforementionedsystemsgive goodsecurityto HDFS however Hadoop is a distributed programming framework for processing huge information where the DataNodes are physically appropriated with its individual tasks furthermore the undertaking given by TaskTracker, requests for more secure processing of data. All above portrayedtechniquesdoesnotgiveDataprotectionbecause of the comparative instrument used to giveinformationsecurity to all clients at HDFS. The measure of scrambled information in the wake of utilizing AES or comparative algorithm is more noteworthy, so these are not proficient where record stockpiling becomes rapidly on account of execution overhead. In the event that we utilize the encryption procedure which give information protection furthermore does not influence size of information an excessive amount of so it support for ongoing application and conceivable to diminish overhead happens in existing framework.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

3. PROPOSEDSYSTEM

Wehaveproposednewtechniquefor securingdata atHDFSbyanalyzingalltechniquespreviouslymentioned.It is actualized by utilizing Real Time Encryption Algorithm andOAuth(calledOpenStandardforAuthorization).OAuth 2.0 is an Open Authentication Protocol that is used for authentication and authorization of client in conventional client-server model. In the traditional client-server model, the customer solicitations to an entrance secured asset on the server by verifying itself utilizing the asset proprietor's international ID. In order to give third-party applications accesstorestrictedresources,theresource ownerverifiesits authorization with the third-party[13].

In proposed system, to authenticate user we have used OAuth 2.0, which returns unique token for each user whoattemptssuccessfullogin.Thetokenreturnedby OAuth server utilized as a part of encryption strategy so it gives informationprivacyandintegritytotheuserdata.Thefiles areencryptedbeforeloadtoHDFSanddecryptedwhenjob execution is in progress [1]. The Real Time Encryption AlgorithmutilizestheOAuthtokenaskey andEncryptdata (uploaded by user) by XoRing with thekey.

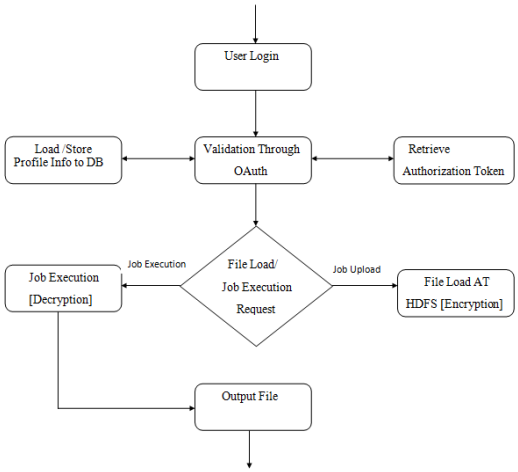


Fig-2:Flow chart

Flow chart is shown in Fig-2. User does log in to system using OAuth 2.0 and then uploads 'n' number ofdocuments(either file or job) as an input to the HDFS. But before writing to HDFS it will be passed to Real Time encryption model. In this model data will be encrypted. Similarlydecryptionwillbeperformedwhenmapreducejob read data from HDFS after job execution request. Authentication token and authorization token provided by OAuth are used for user verification and encryption/decryption algorithmsrespectively.

MODULE DESCRIPTION:

Adaptive encryption:

The recommend system supports adaptive encryption methods for public cloud database service, where distributed and concurrent clients can issue direct SQL operations. By avoiding an architecture based on one [or] multiple intermediate

servers between the clients and the cloud database, the proposed solution guarantees the same level of scalability and availability of the cloud service. A scheme of the recommend architecture where each client executes an encryption engine that manages encryption operations. This software module is accessed by external user applications through the encrypted database interface. The recommend architecture manages five types of information.

• Plain data is the tenant information;
• Encrypted data is stored in the cloud database;
• Plain metadata represent the additional information that is necessary to execute SQL operations on encrypted data;
• Encrypted metadata is the encrypted version of the metadata that are stored in the cloud database;
• Master key is the encryption key of the encrypted metadata that is distributed to legitimate clients.

Metadata structure:

Metadata include all information that allows a legitimate client knowing the master key to execute SQL operations over an encrypted database. They are organized and stored at a table-level granularity to reduce communication overhead for retrieval, and to improve management of concurrent SQL operations. We define all metadata information associated to a table as table metadata. Let us describe the structure of a table metadata .Table metadata includes the correspondence between the plain table name and the encrypted table name because each encrypted table name is randomly generated. Moreover, for each column of the original plain table it also includes a column metadata parameter containing the name and the data type of the corresponding plain column (e.g., integer, string, timestamp). Each column metadata is associated to one or more onion metadata, as many as the number of onions related to the column.

Encrypted database management:

The database administrator generates a master key, and uses it to initialize the architecture metadata. The master key is then distributed to legitimate clients. Each table creation requires the insertion of a new row in the metadata table. For each table creation, the administrator adds a column by specifying the column name, data type and confidentiality parameters. These last are the most important for this paper because they include the set of onions to be associated with the column, the starting layer (denoting the actual layer at creation time) and the field confidentiality of each onion. If the administrator does not specify the confidentiality parameters of a column, then they are automatically chosen by the client with respect to a tenant's policy. Typically, the default policy assumes that the starting layer of each onion is set to its strongest encryption algorithm.

4. TEST SETUP ANDRESULTS

TodotheexperimentwehaveinstalledUbuntuLinux12.04 on our machine. After that we installed Openjdk1.7 and

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

ApacheTomcat1.7andenabledSSH.WeconfiguredHadoop 1.2.1 as a Single-Node Cluster to use the HDFS and MapReduce capabilities. For OAuth server setup we deployed and configured OAuth app [17] for login with Google and also deployed another app [18] for login with Facebook.

TheNameNodeisfocusbitofHadoopinlightofthe waythatitcontrolsthewholeDataNodesexhibitinacluster. It is a Single-Point-of-Failure yet late structures (0.21+) go with Backup NameNode [2] to make it outstandingly available. The DataNodes in HDFS contain all the data on which we be input to our MapReduce jobs. JobTracker at NameNode controls all the tasks which are running on TaskTrackers.

We have implemented two different encryption techniquesofwhichfirstdoestheencryptionusingAESand secondalgorithmperformtheencryptionusingOAuthtoken. We named the second algorithm as Real-time encryption algorithm. The MapReduce programs (Hadoop job) which take the encrypted data as input and execute job, we observed that it took 23.0490 seconds to execute a WordCountMapReduce job for the unencrypted HDFS(normal execution) for size of 10MB test file, while it took83.2780secondsfortheencryptedHDFSusingAESand 54.2360secondstakenforencryptedHDFSusingReal-time encryptionalgorithm(RTEA).

Table-1: ComparisonbetweenAES&RTEAforencryption

Data (MB)	Encryption Type	Encrypted Data(MB)	Time taken for Encryption(sec)	Time taken to Upload to HDFS(sec)
1	AES	1.8819	26.2190	1.7660
	RTEA	1.0659	12.1510	1.6370
10	AES	20.1015	298.0950	2.0110
	RTEA	10.7252	131.5510	1.8120

Table 1 shows the Comparison between AES andReal-time encryptionAlgorithmforfileencryption.Theresultsofdata uploads of plain file and encrypted file is shown in graphs. ThejobexecutiontimeComparisonbetweenAESencryption andtheReal-timeencryptionAlgorithmisshowninTable2. The results of the tests are shown in graphs(figures3-6).

Table -2: Comparison between AES & RTEA for job execution

Data (M	Encryption Type	Encrypted Data(M	Time taken to execute job(sec)
1	AES	1.8819	26.0420
	RTEA	1.0659	22.0510
10	AES	20.1015	83.2780
	RTEA	10.7252	54.2360

5. CONCLUSIONS

In the today's world of Big Data, where information is assembled from various sources in such case, the security is a note worthy issue, as there does not any altered wellspring of information and

HDFS not have any sort of security system. Hadoop embraced by different commercial enterprises to process such enormous and delicate information, requests solid securitysystem.

Along these lines encryption/decryption, authentication & authorization are the techniques those much supportive to secure information at Hadoop Distributed File System.

InFutureworkoursubjectpromptsproduce Hadoopwitha wide range of security techniques for securing information and additionally secure execution ofjob.

REFERENCES

[1] Seonyoung Park and Youngseok Lee, Secure Hadoopwith Encrypted HDFS, Springer-Verlag Berlin Heidelberg in 2013

[2] [2] Dean J., Ghemawat S.: MapReduce: Simplified DataProcessing on Large Cluster, In:OSDI (2004)

[3] Ghemawat S., Gobioff H., Leung, S.: The Google FileSystem. In: ACM Symposium onOperatingSystemsPrinciples (October 2003)

[4] OMalley O., Zhang K., Radia S., Marti R., Harrell C.: Hadoop Security Design,Technical Report (October2009)

[5] White T.: Hadoop: The Definitive Guide, 1st edn.OReilly Media (2009)

[6] Hadoop, http://hadoop.apache.org/

[7] Jason Cohen and Dr. SubatraAcharya Towards aTrustedHadoop Storage Platform:DesignConsiderationsof an AES Based Encryption Scheme with TPM RootedKeyProtections. IEEE 10th International Conference on Ubiquitous Intelligence & Computing in 2013

[8] Lin H., Seh S., Tzeng W., Lin B.P. Toward DataConfidentiality via Integrating Hybrid EncryptionSchemes and Hadoop Distributed FileSystem. 26th IEEE International Conference on Advanced Information Networking and Applications in 2012

[9] ThanhCuong Nguyen, WenfengShen, Jiwei Jiang andWeiminXuA Novel Data Encryption in HDFS. IEEE International Conference on Green Computing and Communicationsin 2013.

[10] Devaraj Das, Owen OeMalley, Sanjay Radia and KanZhang Adding Security to Apache Hadoop.inhortanworks

[11] Songchang Jin, Shuqiang Yang, Xiang Zhu, and Hong Yin Design of a Trusted File System Based on Hadoop
.Springer-Verlag Berlin Heidelberg in 2013

[12] [AdvancedEncryptionStandard,http://en.wikipedia.org/wiki/AdvancedEncryption Standard

[13] Sharma Y. ; Kumar S. and Pai R.M; FormalVerification of OAuth 2.0 Using Alloy Framework .International Conference on Communication Systems and Network Technologies in 2011

[14] Ke Liu and Beijing UnivOAuthBasedAuthentication and Authorization in Open Telco API .IEEE International Conference on Communication Systems and Network Technologies in 2012

[15] Big Data Security: The Evolution of Hadoops Security ModelPosted by Kevin T. Smith on Aug 14, 2013

[16] Securing Big Data Hadoop: A Review of Security Issues, Threats and Solution by Priya P. Sharma and Chandrakant P. Navdeti in 2014