RESEARCH ARTICLE OPEN ACCESS

Cryptographic Security technique in Hadoop for Data Security

Praveen Kumar [1], Pankaj Sharma [2]
M.Tech. Scholar [1], Assistant Professor [2] & HOD
Department of Computer Science & Engineering, MDU Rohtak
India

ABSTRACT
The data utilized by various organizations is increasing at a fast rate. In today's world, firms need to process Zettabytes of knowledge. The normal database management system fails to process such great deal of knowledge. So, we'd like to seek out an efficient approach for handling and process such large amount of knowledge which provides rise to huge data downside. Size or volume of knowledge isn't solely the only criteria to classify huge data, we've got to stay in mind the kind of knowledge that's whether or not data is structured or semi structured or unstructured. Semi structured or unstructured information is troublesome to manage through ancient direction system. To beat this huge knowledge downside Apache created a software tool named Hadoop that uses MapReduce design to process large amount of knowledge. Hadoop doesn't make sure the security and privacy of the stored files in Hadoop. During this thesis, AN uneven key cryptosystem is projected for the secret writing of files stored in Hadoop Distributed file system. We tend to use our projected uneven key rule to encipher the content of knowledge before storing it in HDFS. The results obtained from varied experiments indicate favorable results of above approach to handle security downside related to huge knowledge.

Keywords:- *Big Data, Hadoop, MapReduce, Encryption, Security*

I. INTRODUCTION

Big data, as the name indicates, is very voluminous data with other features such as velocity (streaming data) and variety (data in different formats such as structured, unstructured and semistructured). When data is in the form of relational database, it is known as structured data. When data is in the form of documents of any kind, it is known as unstructured data. When data is in the form of XML, it is known as semi-structured data. While the size used to determine whether a particular data set is considered Bigdata is not firmly defined and continues to change over time, most analysts currently refer to data sets from 30-50 terabytes to multiple Zettabytes as big data.

1.1 Need for Big Data Mining

Big data is the complete data of business with all details. When such data is processed in a distributed environment, it is possible that it produces comprehensive business intelligence. If partial data is processed, it may result in inaccurate business intelligence that cannot be used for making expert decisions. As shown in Figure 1, it is evident that people who are analyzing data were not able to produce correct output. There is elephant over there and people has seen a part of it and understood differently. For instance the leg of the animal is understood like a tree. Probably it is the act of blind person who cannot see the complete picture but can touch and say what it is. Here it is very obvious that biased conclusions are provided. These conclusions are wrong and they cannot help in making well informed decisions. Thus it is understood that when whole data (complete data) is considered, it can produce intelligence for making good decisions [2].

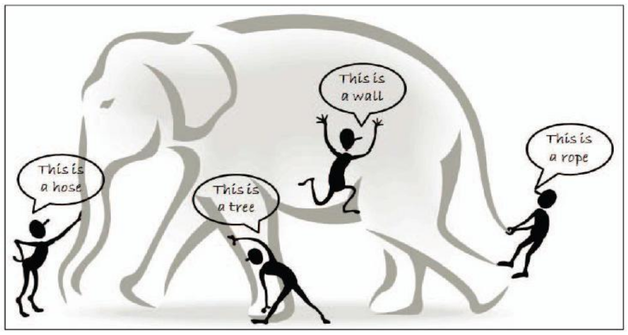


Figure 1: Limited View of Users Provided Biased Conclusions

1.2 4-Vs of Big Data

- **Volume of data:** Refers to the vast amounts of data generated every second. We are not talking Terabytes but Zettabytes or Brontobytes. If we take all the data generated in the world between the beginning of time and 2008, the same amount of data will soon be generated every minute. This makes most data sets too large to store and analyze using traditional database technology. New big data tools use distributed systems so that we can store and analyze data across databases that are dotted around anywhere in the world

- **Variety of data:** Different types of data and sources of data. Data variety exploded from structured and legacy data stored in enterprise repositories to unstructured, semi structured, audio, video, XML etc.
- **Velocity of data:** Velocity refers to the speed of data processing. For time-sensitive processes such as catching fraud, big data must be used as it streams into your enterprise in order to maximize its value.
- **Veracity of data:** refers to the messiness or trustworthiness of the data. With many forms of big data, quality and accuracy are less controllable but big data and analytics technology now allows us to work with these types of data. The volumes often make up for the lack of quality or accuracy

1.3 Hadoop

Hadoop is a Programming framework used to support the processing of large data sets in a distributed computing environment. Hadoop was developed by Google's MapReduce that is a software framework where an application breaks down into various parts.

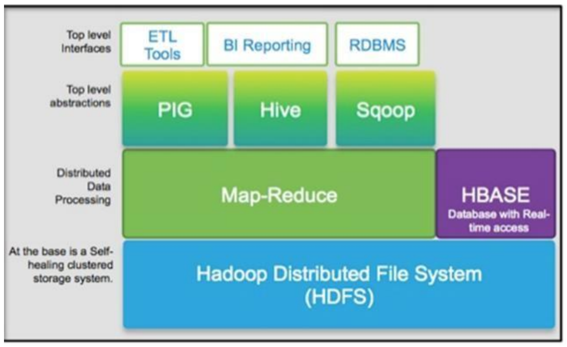
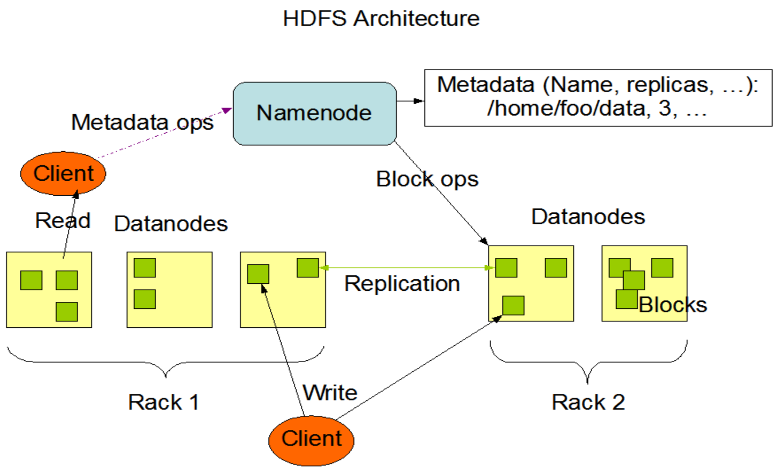

Top level Interfaces
ETL Tools
BI Reporting
RDBMS
Top level abstractions
PIG
Hive
Sqoop
Distributed Data Processing
Map-Reduce
HBASE
Database with Real-time access
At the base is a Self-healing clustered storage system.
Hadoop Distributed File System (HDFS)

Fig. 2 –Hadoop Architecture

The Current Apache Hadoop ecosystem consists of the Hadoop Kernel, MapReduce, HDFS and numbers of various components like Apache Hive, Base and Zookeeper. HDFS and MapReduce are explained in following points [1].

1.3.1 HDFS Architecture

Hadoop incorporates a fault-tolerant stockpiling framework called the Hadoop Distributed File System, or HDFS. HDFS can store gigantic measures of data, scale up incrementally and survive the disappointment of huge parts of the capacity framework without losing information. Hadoop makes bunches of machines and organizes work among them. Bunches can be worked with modest PCs. On the off chance that one comes up short, Hadoop keeps on working for the group without losing information or interfering with work, by moving work to the rest of the machines in the bunch. HDFS oversees capacity on the group by breaking approaching records into pieces, called "squares," and putting away each of the pieces needlessly over the pool of servers. In the basic case, HDFS stores three finish duplicates of each document by replicating each piece to three unique



servers [3].

Fig. 3 HDFS Architecture

1.3.2 MapReduce Architecture

The processing pillar in the Hadoop ecosystem is the MapReduce framework can occur on multiple dimensions. For example, a very large dataset can be reduced into a smaller subset where analytics can be applied. In a traditional data warehousing scenario, this the framework allows the specification of an operation to be applied to a huge data set, divide the problem and data, and run it in parallel. From an analyst's point of view, this might entail applying an ETL operation on the data to produce something usable by the analyst.
In Hadoop, these kinds of operations are written as MapReduce jobs in Java [4]. There are a number of higher level languages like Hive and Pig that make writing these programs easier. The outputs of these jobs can be written back to either HDFS or placed in a traditional data warehouse. There are two functions in MapReduce as follows:
Map – the function takes key/value pairs as input and generates an intermediate set of key/value pairs
Reduce – the function which merges all the intermediate values associated with the same intermediate key

1.3.3 Map Reduce Programming

Map Reduce programming is a new programming approach based on object-oriented programming using Java programming language. It is the process of writing program with two parts such as Map and Reduce. Map takes care of processing big data while reduce takes care combining Map results provided by thousands of worker nodes in the distributed environment. This kind of programming is supported by cloud computing, data centers and the presence of modern processors such as Graphical Processing Units (GPUs). The power of parallel processing is leveraged with big data in such environments.

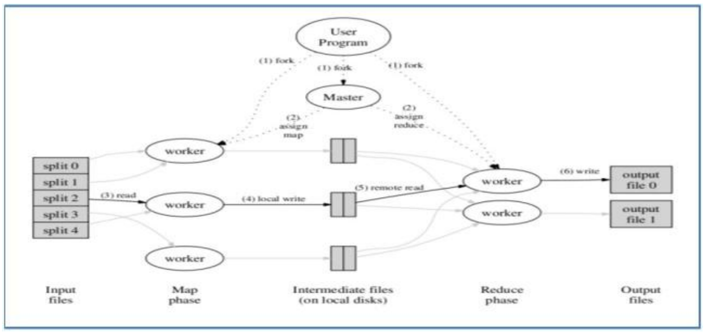


Figure 4: Map Reduce programming paradigm

The architecture shown in Figure 4 is related to Hadoop which is one of the distributed programming frameworks. The Map Reduce programming supported by Hadoop is illustrated in the figure. First of all the input files are divided into multiple parts and each part is given a mapper present in worker nodes. The mapper produces its output. Then the intermediate files are taken by other set of worker nodes for performing Reduce task. Once the reduce task is completed, it is possible to have output files generated. Both input and output files are stored in distributed file system Hadoop.

II. HADOOP SECURITY CONCERNS

2.1 Security Threats in Hadoop

Hadoop does not follow any classic interaction model as the file system is partitioned and the data resides in clusters at different points. One of the two situations can happen: job runs on another node different from the node where the user is authenticated or different set of jobs can run on a same node. The areas of security breach in Hadoop are

i. Unauthorized user can access the HDFS file
ii. Unauthorized user can read/write the data block
iii. Unauthorized user can submit a job, change the priority, or delete the job in the queue.
iv. Running task can access the data of other task through operating system interfaces. Some of the possible solutions can be
    - Access control at the file system level.
    - Access control checks at the beginning of read and write
    - Secure way of user authentication

Authorization is the process of specifying the access right to the resources that the user can access. Without proper authentication service one cannot assure proper authorization. Password authentication is ineffective against

- **Replay attack** - Invader copies the stream of communications in-between two parties and reproduces the same to one or more parties.
- **Stolen verifier attack** - Stolen verifier attack occur when the invader snips the Password verifier from the server and makes himself as a legitimate user [5].

2.2 Security in Hadoop at present

Hadoop default means consider network as trusted and Hadoop client uses local username. In default method, there is no encryption between Hadoop and client host and in HDFS, all files are stored in clear text and controlled by a central server called NameNode. So, HDFS has no security appliance against storage servers that may peep at data content. Additionally, Hadoop and HDFS have no strong security model, in particular the communication between DataNodes and between clients and DataNodes is not encrypted [6].

III. RSA CRYPTOSYSTEM

3.1 RSA Algorithm

As RSA is a public key cryptography, it uses two keys that is public key and private key [16]. Public key is used to encrypt the data and the corresponding private key is used to decrypt data. Lets say C is cipher text and P is the plain text. While encrypting, the following is done

$C=P^e \bmod n$

Where C=cipher text
P=plain text
Both e &n are public
While decrypting, the following is done

$P=C^e \bmod n$

Key Generation:

1. Lets first select two large prime's p and q such that p not equal to q.
2. $n \leftarrow p \times q$
3. $\phi(n) \leftarrow (p-1) \times (q-1)$
4. Select e such that $1 < e < \phi(n)$ and e is coprime to $\phi(n)$.
5. $d \leftarrow e^{-1} \bmod \phi(n)$
6. Public key $\leftarrow (e,n)$
7. Private Key $\leftarrow d$

3.2 Proposed Encryption Method

The proposed method is also an asymmetric-key cryptosystem which uses two keys for encryption and decryption. Public key used for encryption and corresponding private key is used for decryption. [7] Proposed a 'n' prime number RSA cryptosystem. Proposed another version of RSA. Reverse the plain text then after encrypt it. Our proposed scheme has drawn idea from these schemes. In RSA , only two large prime numbers are considered while in our proposed encryption system, we have taken four large prime numbers. As in asymmetric-key cryptography, the point lies in how to make it difficult for the attacker to factorize n which is the multiplication of those four prime numbers. Our proposed system is not only about increasing prime numbers but also we have applied security in terms of public key and private key. In RSA, the public key consists of e and n but we have included another parameter f. And in private key, we have included three other parameters a and b.

3.2.1 Key Generation

1. Select four large prime numbers p, q, r and s such that they are all unique and not equal to each other.
2. n$\leftarrow$ p x q x r x s
3. ø(n) $\leftarrow$(p-1) x (q-1) x (r-1) x (s-1)
4. Select e such that 1 < e < ø (n) and e is coprime to ø (n).
5. d $\leftarrow$ e^{-1} mod ø (n)
6. Select an integer b < ø $(n)^{-1}$
7. Pick another integer a such that b < a < ø (n)
8. Pick another integer h such that a < h < ø (n)
9. Calculate f = $(b^a)^h$
10. Public key $\leftarrow$(e, n, f)
11. Private Key $\leftarrow$(d, a, b, h)

3.2.2 Encryption

1. Lets say one party X wants to send a message to Y securely over an insecure channel.
2. Let's take 'e' be Y's public key. 'e' is known to A because 'e' is public key.
3. Now we want to apply our encryption scheme to the message or plain text P.
4. First, we have to convert the plaintext or message to an integer in the range 0 < P < n.
5. Now in the final step calculate cipher text which is C=(P x f)e mod n

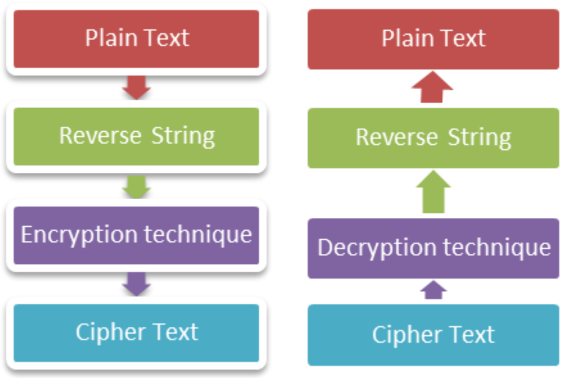


Fig 4 - Proposed Encryption & Decryption

3.2.3 Decryption

1. Let us consider C to be the cipher text which is received by Y from X.
2. Now we have to calculate the plain text from the received cipher text and reverse the received plain text which can be derived as:- P= (($b\phi$(n)−a mod n) x C)d mod n.

So, in this way we can encrypt and decrypt data as per our requirements. The data can be encrypted using the public key and can be decrypted using the corresponding private key as stated earlier. Comparison between Proposed Scheme and Existing Scheme

IV. COMPARISON BETWEEN PROPOSED SCHEME AND EXISTING SCHEME

The key generation algorithm takes a little more time than RSA algorithm. If we consider encryption of files then our proposed method stands out as it uses four unique prime numbers instead of two as in RSA algorithm. As the number of prime number is increased, it will make the adversary difficult to factorize it. Apart from that there is another natural number each included in both public key and private key. So the overall security increases.

File Size (in Bytes)	Time Taken By Hadoop without Encryption (in seconds)	Time Taken by Hadoop with Proposed Method (in seconds)	Difference (in second)
87040000	143	192	49
174080000	80	188	108
230400000	179	327	142
348160000	119	385	224
435200000	180	412	205
522240000	200	425	212

Table 1- Comparison between Proposed Scheme and Existing Scheme

Table 1 shows how much time Hadoop takes to complete the assigned job without using any encryption scheme and how much time it takes to complete the map-reduce job after applying the proposed encryption scheme. The file size is shown in no of bytes, the time taken by Hadoop to complete the job without encryption is calculated in seconds, time taken by Hadoop to complete the job with our proposed encryption system is calculated in seconds and then the difference between these two.

V. CONCLUSION

Though Hadoop enables us to overcome challenges faced in industries and institutions due to big data, it has not any security mechanism. The data stored in Hadoop can be compromised by an attacker or eavesdropper.

As Hadoop does not provide any security mechanism, the authenticity of data is always at stake. The proposed asymmetric key algorithm encrypts the content of files before storing it into HDFS thus by securing it from the various attacks on network. Hence, the data or files now can be stored in Hadoop without worrying about security issues by applying the encryption algorithm on the files before storing it in Hadoop.

REFERENCES

[1] Hadoop, http://Hadoop.apache.org/

[2] D. Krishna Madhuri “A NOVEL APPROACH FOR PROCESSING BIG DATA” International Journal of Database Management Systems (IJDMS) Vol.8, No.5, October 2016

[3] M. Zinkevich, J. Langford, and A. J. Smola, “Slow learners are fast,” in Proc. Adv. Neural Inf. Process. Syst., 2009, pp. 2331–2339.

[4] Ghemawat, Sanjay, and Howard Gobioff. "Shun-Tak Leung-The Google File system." ACM SIGOPS Operating Systems Review-SOSP 3 (2003): 37.

[5] Jam, Masoumeh Rezaei, Leili Mohammad Khanli, Morteza Sargolzaei Javan, and Mohammad Kazem Akbari. "A survey on security of Hadoop." In Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on, pp. 716-721. IEEE, 2014

[6] Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." Communications of the ACM 51, no. 1 (2008): 107-113.

[7] N. Somu, A. Gangaa, and V. S. Sriram, "Authentication Service in Hadoop Using one Time Pad," Indian Journal of Science and Technology, vol. 7, pp. 56-62, 2014.

[8] Jam, Masoumeh Rezaei, Leili Mohammad Khanli, Morteza Sargolzaei Javan, and Mohammad Kazem Akbari. "A survey on security of Hadoop." In Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on, pp. 716-721. IEEE, 2014