Enhancing Security Feature of Hadoop using Data Encryption

Himani Verma, Dilbag Singh
Department of Computer Science and Applications, Chaudhary Devi Lal University, Sirsa, Haryana, India

ABSTRACT

Data size of organizations and institutions is growing in size rapidly. Organizations have to process data in terms of petabytes. The traditional database management system fails to process such huge amount of data. So, there was the need to find out an effective mean for handling and processing of such a large amount of data. It gives rise to big data analysis. Size or volume of the data is not only the single criteria to classify big data, but type of data such as structured or semi structured or unstructured also needs to be considered. Semi structured or unstructured data is difficult to manage through traditional database management system. To conquer this huge information issue, Apache delivered a product device named as Hadoop which utilizes MapReduce design to process expansive measure of information. MapReduce architecture processes the data in parallel. Though it gives solution to overcome the big data problem but it does not ensure the security of the stored files in Hadoop. In this paper, an awry key cryptosystem is proposed for the encryption of records put away in HDFS. The proposed cryptosystem is used to encrypt the data before storing into HDFS. The results of the proposed approach provide the security associated with big data.

Keywords: MapReduce, HDFS, Hadoop, Cryptography.

I. INTRODUCTION

In today's age of global era, extensive data is being generated with the velocity of light. Data from various sources like social platforms, web activities, life sciences, stock exchanges etc. contributes to this explosion. Increasing size of data made it difficult to understand and use in decision making.

Analysis and security of big data has astounded the researchers and scientists. Their main focus is the handling of such a vast amount of data by ensuring the security [1]. Generally the traditional database management system has a size cap in process of data. It is absolutely impossible to overcome the problems faced by the exponentially growing size of data. Moreover, some database supports infinite size of data but the data accessing time and processing time is extremely large which is not acceptable. So, there is a need to find an efficient solution for these problems or challenges faced by big data problem.

To deal with huge amount of data, Object Relational Databases concepts was used. Backups, recovery and searching made its use very complicated. Continuous research has been done on big data problems and to develop a parallel processing architecture for its solution. This continuous research resulted in invention of Hadoop. Hadoop processes the data in sequentially manner not like the traditional system which processes the data in parallel manner. Although Hadoop solves issues regarding big data like processing big data, managing big data within a tolerable time limit etc, it still lacks in security aspect. The security and privacy of data is not ensured. This motivated for developing security mechanism of data in Hadoop [2].

Computation is performed by Map Reduce Programming Framework. Map Reduce is a

distributed parallel processing engine of Hadoop, which processes the data in parallel steps with two phases Map and Reduce [3].

II. PROBLEM STATEMENT

Hadoop is widely used as Big Data Analytics framework but still there are some security concerns that are uncovered. The paper focuses on ensuring the security of Big Data system using asymmetric key encryption methods. This study implements a variation of RSA Algorithm to encrypt data before storing in Hadoop.

III. OBJECTIVES

Present study has been carried out with the following objectives in mind:

- To study the concept of Big Data.
- To analyze Big Data with Hadoop using Map-Reduce programming paradigm.
- An encryption scheme has been proposed used an asymmetric key cryptosystem for encrypting of data before storing it in Hadoop.

IV. RESEARCH METHODOLOGY

In present study, Amazon Web Services has been procured. A web service Amazon Elastic Compute Cloud provides resizable compute capacity in the cloud on the rented commodity hardware. Hadoop Cluster environment set up is done. Datasets are taken and data processing has been executed over the huge data and performance of Hadoop is evaluated. In traditional methods, data is stored at one location and to process the data it is moved to computational location it needs to archive a big portion of raw data, and the processing is carried out on small chunk of which lacks the efficient data analytics [4]. This problem may be overcome using the HADOOP framework as the HADOOP Framework facilitates the processing on the whole data. In Hadoop, storage and computation is performed at the same location which will give insights into the whole data and hence no need of data archiving [5]. Hence, present research is Experimental study.

V. PROPOSED ENCRYPTION METHOD

The proposed method uses an asymmetric-key cryptosystem. It uses two keys: one for encryption and one for decryption. Public key is used for encryption and corresponding private key is used for decryption. A '4' prime number RSA cryptosystem is proposed. In existing RSA only two prime numbers are being used while in the proposed encryption system four prime numbers are used. 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 these four prime numbers. Hence, the proposed system is providing greater security.

VI. PROPOSED ALGORTIHM

Step 1. Take four large prime numbers n1, n2, n3 and n4 such that they are all unique and not equal to each other.

Step 2. $N = n1 * n2 * n3 * n4$

Step 3. $O(N) = (n1-1) * (n2-1) * (n3-1) * (n4-1)$

Step 4. Take e such that $1 < e < O(N)$ and e is co-prime to O(N).

Step 5. $d = e^{-1} \bmod O(N)$

Step 6. Public key (e, N)

Step 7. Private Key (d)

Encryption

1. Sender X wants to send a message to receiver Y by ensuring the security through an insecure channel.

2. Let's take 'e' be Y's public key. 'e' is known to A because 'e' is public key.

3. Apply this encryption scheme to the message or plain text P. First, convert the plaintext or message to an integer in the range 0< P < N.

4. Now in the final step calculate cipher text which is C= $(P)^e$ mod N.

Decryption

1. Consider C to be the cipher text which is received by Y from X.

2. Calculate Plain text P=$(C)^d$ mod N.

Output

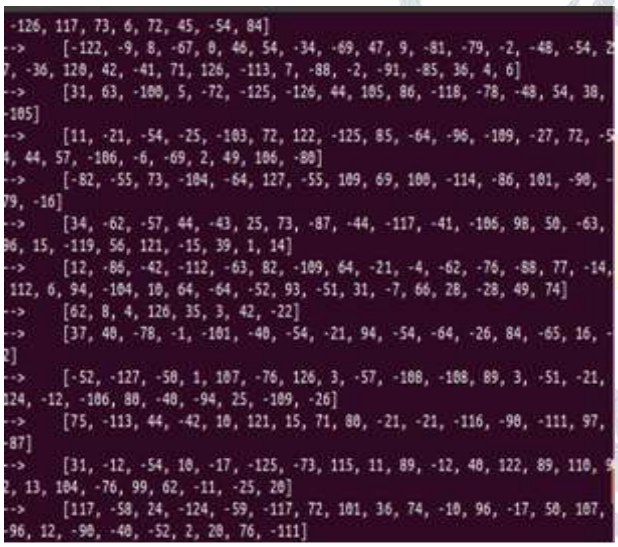

Figure (i) Encrypted output of Data

Figure (i) shows the output of the proposed method. Data is stored in byte format rather than in text format. This data in this given format is not useful for any unauthorized user being encrypted and nearly impossible for a user to decrypt without the algorithm keys. It is very difficult to factorize that large number into four unique prime numbers in order to decrypt. The security threats with the Hadoop systems by such implementations can be overcome.

Comparison of Existing System and Proposed System

The key generation algorithm takes a little more time than no security system implemented. If encryption of files is considered then the 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. So the overall security increases but here in this section the discussion of the time overhead when proposed security model is implemented.

TABLE 1: Comparison of Existing and Proposed Method

Data Volume (in GB)	Execution Time in Hadoop Without Encryption Method (in Seconds)	Execution Time in Hadoop With Encryption Method (in Seconds)	Difference %
0.01	12	15	20
0.1	90	96	6.25
0.5	280	290	3.45
1	490	510	3.92
2	732	750	2.4
5	1205	1260	4.37

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 job after applying the proposed encryption scheme. The execution time in Hadoop without encryption method is taken by running the instances of raw data on Hadoop cluster set up on Amazon Web Services using Elastic Cloud Compute. Hive provides us the feature to query over the data and it is noticed that Hadoop takes 12 seconds to fetch the raw data of size 0.01 GB, 90 seconds for 0.1 GB data, 280 seconds for 0.5 GB data, 490 seconds for 1 GB data, 732 seconds for 2 GB data and 1205 seconds for 5 GB data from the server and 15 seconds to fetch data of size 0.01 GB when it is stored in the encrypted form in Server, 96 seconds for 0.1 GB data, 290 seconds for 0.5 GB data, 510 seconds for 1 GB data, 750 seconds for 2 GB data and 1260 seconds

for 5 GB data. From the above table it is clear that as the data size increases then the difference in execution time goes on decreasing.

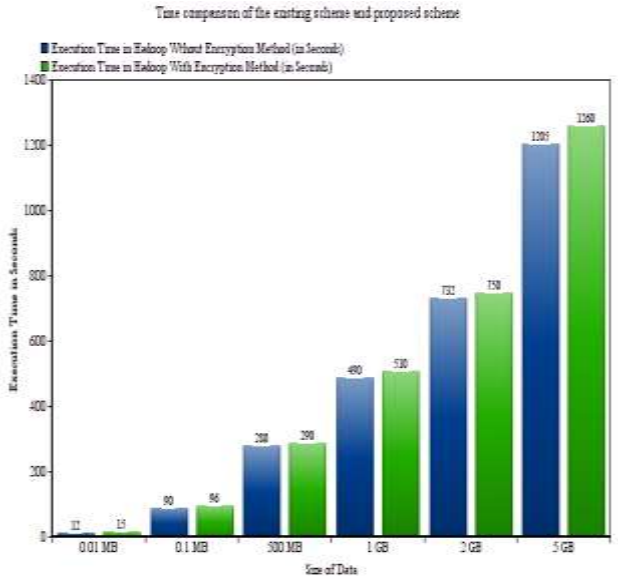

Figure (ii) Execution time of existing and proposed system.

In Figure 4.6 the graph represents the execution time of existing and proposed system. X-axis represents the size of data and y-axis represents execution time in seconds. As seen from the graph, when the size of data is 5GB, the execution time is 1260 seconds (with encryption) and 1205 seconds (without encryption) which is not so much overhead. So this proposed algorithm can be used for better security of data.

VII. CONCLUSION

The proposed encryption scheme secures the content by encrypting the contents using proposed encryption method. The proposed algorithm uses four prime numbers to make the factorization difficult before storing it into HDFS which results into the security of the contents form the attacker. In present study comparison of execution time in Hadoop is made with and without encryption. When the size of data is 0.01 GB, the execution time in Hadoop without encryption method is 12 seconds, for 0.1 GB data it takes 90 seconds, for 0.5 GB data it takes 280 seconds, for 1 GB data it takes 490 seconds, for 2 GB data it takes 732 seconds and for 5 GB data it takes 1205 seconds. The execution time in Hadoop with encryption method is 15 seconds when the size of data is 0.01 GB, 96 seconds for 0.1 GB data, 290 seconds for 0.5 GB data, 510 seconds for 1 GB data, 750 seconds for 2 GB data and 1260 seconds for 5 GB data. It can be observed that as the size of the data increases, the difference in execution time in Hadoop with encryption goes on decreasing. Some overhead time has been noticed after applying encryption algorithm but the time factor is not so much impactful for large data items. Data now can be stored in Hadoop without worrying about security issues. Hence, it is recommended that the organizations/researchers must use the Hadoop Framework with encryption model for handling the huge amount of data.

VIII. REFERENCES

[1] Big data - wikipedia, en.wikipedia.org/wiki/big data.

[2] Tom White “Hadoop Definitive Guide”, Second Edition, O’Reilly Media, pp 1-9, October-2010.

[3] Aditya B Patel, Manashvi Birla, and Ushma Nair. Addressing big data problem using hadoop and map reduce. In Engineering (NUiCONE), 2012 Nirma University International Conference on, pages 1{5.IEEE, 2016.

[4] Vinod Sharma1, Prof. N.K. Joshi2 The Evolution of Big Data Security through Hadoop Incremental Security Model ISSN(Online): 2319-8753 ISSN (Print): 2347-6710 International Journal of Innovative Research in Science, Engineering and Technology Vol. 4, Issue 5, May 2015

[5] Stephen Kaisler, Frank Armour, J Alberto Espinosa, and William Money. Big data: Issues and challenges moving forward. In System Sciences (HICSS), 2013 46th Hawaii International Conference on, pages 995{1004.IEEE, 2013.