An Efficient Big Data Security Method Using AES Algorithm With Conditional Filtering Approach

S.Rajarajesware [1]
(Research Scholar,Manonmaniam Sundaranar University,Tirunelveli,TN,India.
email: sivrajarajesware@gmail.com)
Dr.H.Abdul Rauf [2]
(Professor & Dean, Sree Sastha Institute of Engineering and Technology, Chennai, TN, India.
email: harauf@yahoo.com)
Dr.S.P.Victor [3]
(Dean of Science and Associate Professor,St.Xavier's College,Palayamkottai,TN, India.
email: drspvictor@gmail.com)

Abstract: The data which are voluminous and arrive at great velocity and variety are termed as big data. Big data dealings make user transactions hassle free and ease. One thing which needs to pay attention in big data storage and retrieval is its security. To have a secured data storage and secure communication and also to ensure that the data reaches the right receiver, when requested from a distributed file system, a system is needed that ensures that all these transactional aspects are carried out in a confidential manner, that too with integrity and authentication. Also, in a big data environment, the time taken to encrypt the data should be minimized to improve the speed and efficiency of the process. For this, in this paper an encryption technique using AES algorithm along with a filtering method is proposed, which will mitigate processing time as per the filtering criteria set.
Key words: Big data, cryptography, Encryption, Decryption, Security

1. Introduction

According to IDC's statistics, by 2020 the volume growth rate of the digital data will be 44 to 50 times. This type of data is mostly unstructured in nature. Accommodating in addition to value making with this Big Data is a crucial issue. Due to the evolution of distributed computing, the third party computing environments like cloud computing which are emerged recently renders the needs of Big Data storage. These Cloud computing resources are shared across many users on demand basis on pay per use over internet. But as they are third party service providers, cannot guarantee the trust on the customers' data storage [1].

With cloud computing, deployment of IT systems and data storage is shifted to off-premises third-party IT infrastructures. Deployment on off-premises third party IT infrastructures have the following characteristics.

• Data owners have only limited control over the IT infrastructure, therefore data owners must establish a mechanism to mandate the enforcement of their security policies to ensure data confidentiality and integrity.

• Cloud service providers have excessive privileges. This allows cloud service providers to control and modify users' IT system and data.

The above two characteristics lead directly to a very low level of trust on keeping and sharing data on a cloud, when comparing to that of conventional infrastructures where users have a certain degree of control on the underlying infrastructures.[3]

Apache Hadoop[4] is an open-source software framework that supports data-intensive distributed applications, the Hadoop Distributed File System (HDFS), is a distributed, scalable, and portable file system written in Java for the Hadoop framework, and HDFS is cloud storage the most widely used tool.

Given the wide range of applications for Big Data, it is not hard to imagine that sensitive data could end up in an organization's HDFS infrastructure, making it a ripe target for exploitation. Depending on the business of the organization, this could have the potential to be a treasure trove of information that an adversary could become acutely interested in. If the organization can gain financial benefit from the data stored within HDFS, someone else may be able to as well [5].

HDFS is a highly fault-tolerant Hadoop distributed file system [7].Fig:1 shows the HDFS architecture.

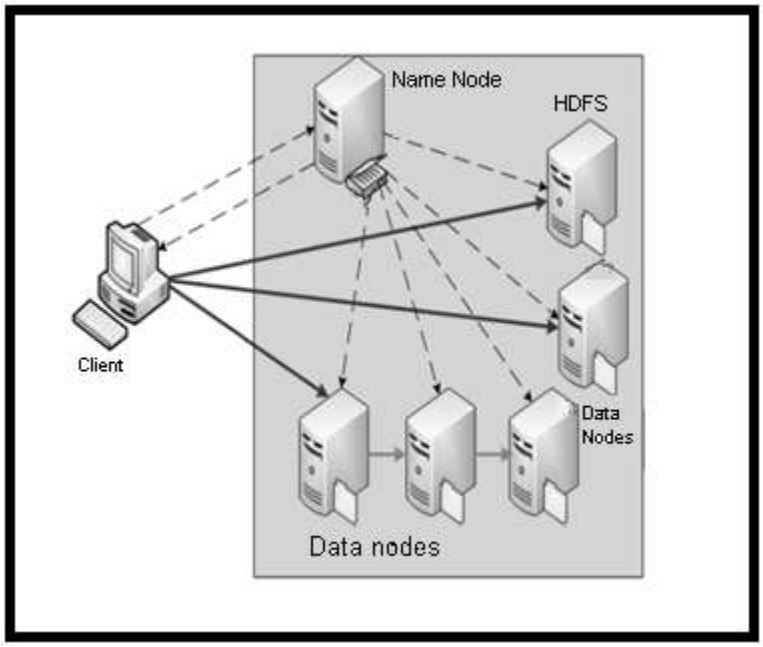

Fig 1: HDFS architecture

Hadoop supports various file systems and the default one is Hadoop distributed file system (HDFS). In HDFS, all files are stored in cleartext and controlled by a central server. Thus, HDFS is not secure against storage servers that may peep at data content. Hadoop also supports Amazon Simple Storage Service (S3). In the official website of Amazon S3, users are recommended to encrypt their data before uploading them to strengthen data security. Applying cryptographic primitives to guarantee strong data confidentiality is a common practice in conventional networked and distributed storage systems [8].

Big data is a word which describes a huge amount of data for both structured and unstructured data. Big data is huge data set with volume, velocity, and variety. With the increased access to the web-based, mobile and cloud-based application, sensitive data is accessed from different platforms by different users. These platforms are vulnerable to hacking, mainly if they are free or low cost. Now a days, companies are collecting and processing huge amount of data or information. Data which are stored by the user ensure that this data is secure. The loss in data security leads to company's financial loss and decreases company's reputation. Therefore security is most important in big data [11].

Big data is an all-encompassing term for any collection of data sets so large and complex that it becomes difficult to process using traditional data processing applications. With advanced big data analyzing technologies, insights can be acquired to enable better decision making for critical development areas such as health care, economic productivity, energy, and natural disaster prediction. The big data refers to massive amounts of digital information, companies and government collect about us and our surroundings. Voluminous data are generated from a variety of users and devices, and are to be stored and processed in powerful data centres. As such, there is a strong demand for building an unimpeded network infrastructure to gather geologically distributed and rapidly generated data, and

move them to data centres for effective knowledge discovery. It's just standard data that's usually distributed across multiple locations, from a diverse array of sources, in different formats and often unstructured. The challenges include analysis, capture, curation, search, sharing, storage, transfer, visualization, and privacy violations.

Considering privacy and security, with a variety of personal data such as buying preference, healthcare records, and location-based information being collected by big data applications and transferred over networks, the public's concerns about data privacy and security naturally arise. While there have been significant studies on protecting data centres from being attacked, the privacy and security loopholes when moving the sourced data to data centres remain to be addressed. There is an urgent demand on technologies that endeavour to enforce privacy and security in data transmission. Given the huge data volume and number of sources, this requires a new generation of encryption solutions [12].

Many encryption algorithms are widely available and used in information security. They can be categorized into Symmetric (private) and Asymmetric (public) keys encryption. In Symmetric keys encryption or secret key encryption, only one key is used to encrypt and decrypt data. In Asymmetric keys, two keys are used; private and public keys. Public key is used for encryption and private key is used for decryption (e.g. RSA). Public key encryption is based on mathematical functions, computationally intensive. There are many examples of strong and weak keys of cryptography algorithms like DES, AES. DES uses one 64-bits key while AES uses various 128,192,256 bits keys.

Asymmetric key encryption or public key encryption is used to solve the problem of key distribution. In Asymmetric keys, two keys are used; private and public keys. Public key is used for encryption and private key is used for decryption (E.g. RSA and Digital Signatures). Because users tend to use two keys: public key, which is known to the public and private key which is known only to the user. There is no need for distributing them prior to transmission. However, public key encryption is based on mathematical functions, computationally intensive and is not very efficient for small mobile devices. Asymmetric encryption techniques are almost 1000 times slower than Symmetric techniques, because they require more computational processing power. Encryption is a well known technology for protecting sensitive data. [13].

In this paper, it is proposed to design a filtering algorithm that has to be used with big data. Also by analysing, the best among the encryption algorithms available will be identified. Combination of these two algorithms are used to provide better data security for sensitive data in big data environment.

2 Literature Survey

Ganesan Zhao et al [2010], propose a system for trusted data sharing through untrusted cloud providers, a progressive encryption scheme based on elliptic curve encryption. This proposed progressive encryption scheme allows data to be encrypted multiple times with different keys and produces a final ciphertext that can be decrypted with a single decryption key in a single decryption operation. This scheme allows changing the encryption key without decrypting the data first, enabling the re-encryption of data in an untrusted environment.

Anup Mathew [2012] points out that people are increasingly relying on a number of online file storage systems to backup their data or use it as a collaborative tool in real time. All these services bring with it a fair share of security and privacy vulnerabilities for all the conveniences provided by them. This includes issues related to data security, privacy and availability with storing data on third party service providers. As a solution he proposed a technique to enforce security on the service providers by using provenance labels so that the clients or consumers are assured that they get the correct service they are paying for and thus ensuring maximum security for their data [2].

H. Y. Lin et al [2012] addressed the data confidentiality issue by integrating hybrid encryption schemes and the Hadoop distributed file system (HDFS). They proposed two integrations, HDFS-RSA and HDFS-Pairing, as extensions of HDFS. They have introduced integration of hybrid encryption schemes and HDFS as an alternative secure storage system for Hadoop. It is stated that HDFSRSA and HDFS-Pairing have considerable overhead on writing operations and acceptable overhead on reading operations and HDFS-RSA and HDFS-Pairing are suitable for write-once read-many applications.

J. Cohen et al [2013] examined the concept of combining trusted computing technologies with the Apache Hadoop Distributed File System (HDFS) in an effort to address concerns of data confidentiality and integrity. To accomplish this, they had used hardware accelerated encryption with key protections tied to the hardware based TPM (Trusted Platform Module).
S. Park et al [2013] said that the current Hadoop does not support encryption of storing HDFS blocks, which is a fundamental solution for secure Hadoop. Therefore, a secure Hadoop architecture is proposed by adding encryption and decryption functions in HDFS by adding the AES encrypt/decrypt class to CompressionCodec in Hadoop. From experiments with a small Hadoop test bed, they had shown that the representative MapReduce job on encrypted HDFS generates affordable computation overhead.

S. Jin, S. Yang et al [2013] proposed a design of trusted file system for Hadoop. The design uses the latest cryptography—fully homomorphic encryption technology and authentication agent technology. It ensures the reliability and safety from the three levels of hardware, data, users and operations. The homomorphic encryption technology enables the encrypted data to be operable to protect the security of the data and the efficiency of the application. The authentication agent technology offers a variety of access control rules, which are a combination of access control mechanisms, privilege separation and security audit mechanisms, to ensure the safety for the data stored in the Hadoop file system.

Dr.PrernaMahajan (2013) et al., in their research work surveyed the performance of existing encryption techniques like AES, DES and RSA algorithms. Based on the text files used and the experimental result it was concluded that AES algorithm consumes least encryption and RSA consume longest encryption time. Also observed that Decryption of AES algorithm is better than other algorithms. From the simulation result, they evaluated that AES algorithm is much better than DES and RSA algorithm.

H. Zhou et al [2014] tried to solve the security issues of network features and data sharing features in cloud storage service through a data security access scheme in cloud storage Based on Attribute-Group. In this scheme, data owners do not participate in the specific operation of the property and user rights. Re-encryption on the NameNode, can reduce the cost of the computation and management of the client, and also reduce the complexity of rights management and property management

Garima Saini et al [2014] introduced a mechanism to provide secure data by combining three algorithm DSA , DES and Steganography to provide security of data in cloud computing. Their plan is to provide authenticity, security and data integrity to the data by means of implementing these three algorithm. But it seems that the Time complexity is high because it is a one by one process, which may be improved by using other security algorithms.

Priyadarshinipatil(2015) et al., made a comprehensive evaluation of cryptographic algorithms DES, 3DES, AES, RSA and Blowfish and arrived at the conclusion based on their implementation results obtained on the evaluation parameters, encryption time, decryption time, memory used etc., that AES can be used in applications where confidentiality and integrity is of highest priority. And Blowfish is strongest against guessing attacks. Also Blowfish consumes least time and less memory amongst all.

Bhargavi, et. al [2016] states that normally cloud storage is used for storing big data. But despite of effective cost saving, cloud storage is prone to many security threats. The confidentiality of the users' data is a critical issue at cloud based services. At a base level of cloud based storage, data security is a critical issue and prone to security violations. The main security violations are: Data Leakage, Unauthorized access, Denial of service of Resources. The three key components on securing Big Data on General cloud based storage are given as: Integrity, confidentiality and availability. The feasible solutions to address the BIG DATA storage at cloud environments are presented in two perspectives. (i) By considering underlying technology of the cloud as a black box. (II) By considering HADOOP distributed framework based cloud data center [1].

Sreenivasa B.L [2016] et al., had taken into consideration the following encryption techniques: Homomorphic encryption algorithm, Verifiable computation algorithm, Message digest algorithm (MD5) or Hash function, Key rotation algorithm, DES, Rijndael Encryption or AES algorithm. To summarize the performance of these encryption / decryption algorithms measures such as confidentiality, integrity, security etc., were considered. It shows the preferable order is AES, DES and Key rotation algorithms.

Afolabi, A.O [2016] et.al. had performed a comparative analysis of some selected cryptographic algorithms and presented the conclusion of results of their analysis as given below:

Encryption algorithms play an important role in communication security where encryption time, memory usage, output byte and battery power are the major issues of concern. The performance evaluation of the selected AES, DES, 3DES and RSA encryption algorithms were carried out based on the encryption time, memory usage, output byte, power consumption rate, flexibility and security. Based on the text files used and the experimental results, it was concluded that AES algorithm consumes least encryption time and AES algorithm has least memory usage. While encryption time difference is very minor in case of AES algorithm and DES algorithm, RSA consumes large encryption time and memory space that is very high. During this analysis it was observed that AES (Rijndael) was the best among all the encryption algorithms in terms of Security, Flexibility, Memory usage, Encryption performance, and power consumption rate [14].

From the above analysis, it is found that AES is the best algorithm among the available algorithms. Hence it is proposed to combine the filtering method with this AES algorithm for minimizing the processing time and thereby increasing the efficiency of the system.

3. Objective

The main objective of this paper is to devise a method by which data should be securely sent and retrieved in a big data environment and at the same time it should not increase any kind of overheads in computation, storage or communication.

4. Proposed Work

One way of providing security to the big data is to encrypt the contents and store them and again decrypt them when requested by valid users. But, when this is applied to the entire volume of data, it will take extra processing time and computational power, which is not desirable. This can be mitigated by applying a new strategy. It is to filter the sensitive data out of the big data and applying encryption to only those sensitive data. Therefore, sensitive data will be stored after encryption, whereas non-sensitive data items will be stored directly without encryption. While retrieving these data items, the encrypted ones will get decrypted and sent to the requesting user, once their authentication is verified by password or other authentication means. This will provide better security without much increase in processing time and power. Also, it provides security against unauthorised access or intrusions.

5 Design and Implementation

In this paper, the following algorithm is used to filter sensitive data out of the data that arrive and to be stored as big data.

Algorithm:

Step 1: Start the process.
Step 2: Set the criteria to filter sensitive data from the data that arrive.
Step 3: Check if the data received satisfies the criteria set in step 2.
Step 4: If yes, encrypt and store the data
Else
Store the data.
Step 5: Repeat step 4 until end of data arrival.
Step 6: Stop the process.

This is depicted by the following diagram.

Fig 2: Encryption process after applying filtering method based on criteria set.

From the heterogenous data that arrive as big data, the sensitive portion is filtered by setting up the criteria. Each data should pass through this and if the data satisfies the criteria, then it will be encrypted and stored, otherwise it will just be stored. For encryption, AES algorithm is used.

6. Experimentation and Results

The efficiency of the big data transactions on applying this combination of filtering algorithm and encryption algorithm will be more with respect to processing time and storage and provides better security than applying encryption algorithm alone.

This is depicted in the following graph based on the tables given below. As an example, here the data of bank transactions is considered in which the filtering criteria may filter those entries containing numeric values having 6 digits or more, if it is required to secure transaction details of 1 lakh and more. Filtering criteria can also contain more than one condition as per our need. For encryption, here AES algorithm is used.

S.NO	NAME OF THE ACCOUNT HOLDER	BALANCE (in Rupees)
1	ANIL	25000
2	AJAY	50000
3	AKIL	234500
4	ARTHI	345000
5	BALA	12000
6	BARANI	102000
7	BABU	400000
8	BAMA	23400
9	CHARAN	154000
10	CHITRA	52000
11	DEEPAK	545000
12	JAYAN	32670
13	KARTHIK	3000
14	LATHA	56000
15	MANI	371000
16	MANOJ	30500
17	NALINI	200000
18	PAARI	31034
19	RANI	512967
20	SARA	34000

Table 1: Sample data depicting various account holders with their final balance, before applying filtering algorithm.

S.NO	NAME OF THE ACCOUNT HOLDER	BALANCE (in Rupees)
1.	AKIL	234500
2.	ARTHI	345000
3.	BARANI	102000
4.	BABU	400000
5.	CHARAN	154000
6.	DEEPAK	545000
7.	MANI	371000
8.	NALINI	200000
9.	RANI	512967

Table 2: After applying filtering algorithm.

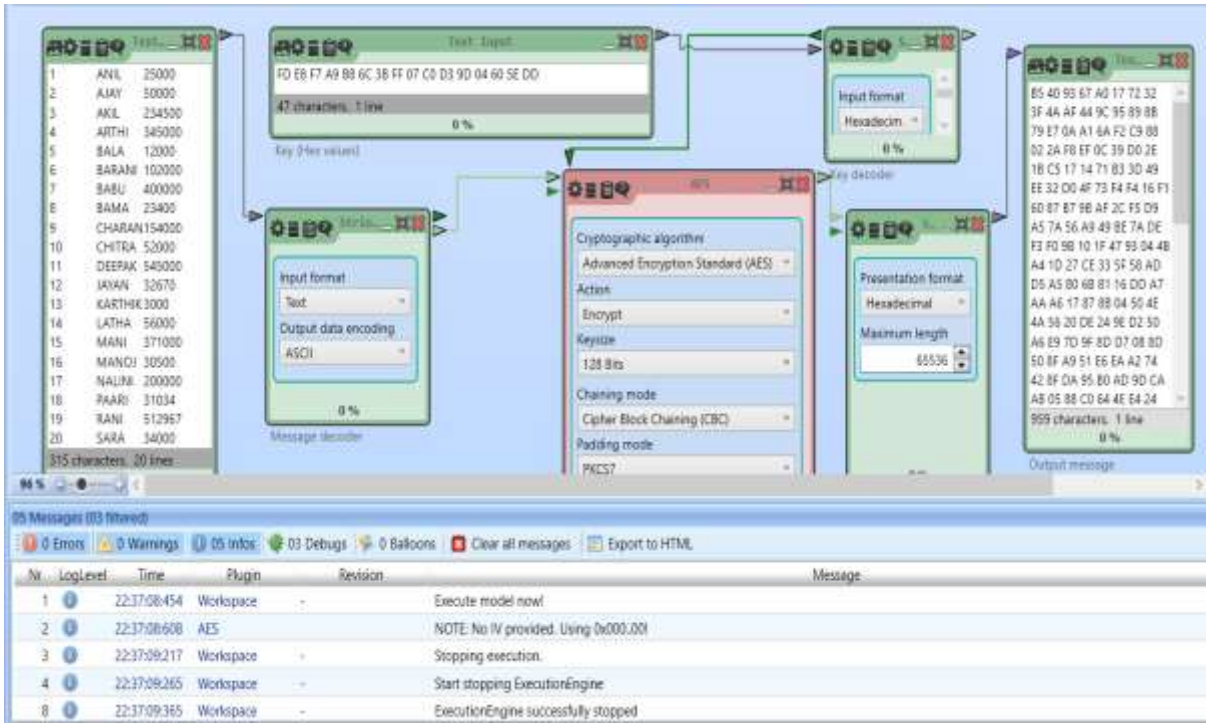

Fig 3: Encryption with AES, before applying filtering algorithm.

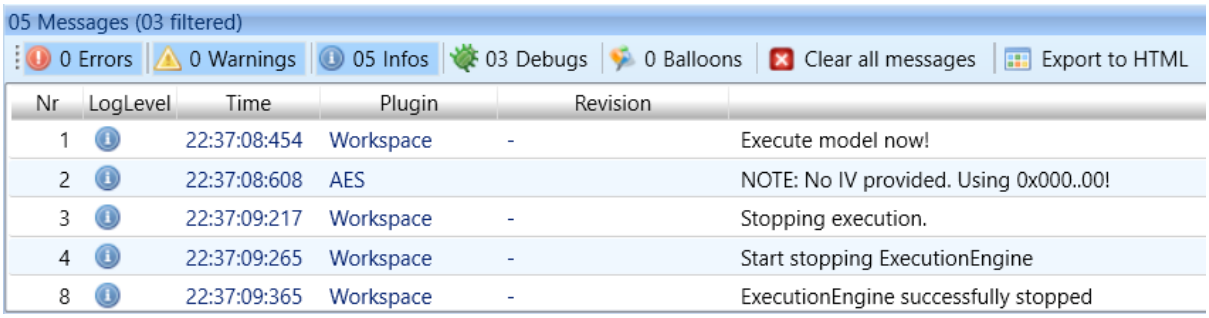

Fig 4: Encryption with AES - Time details (before applying filtering algorithm).

Fig 5: Encryption with AES after applying filtering algorithm

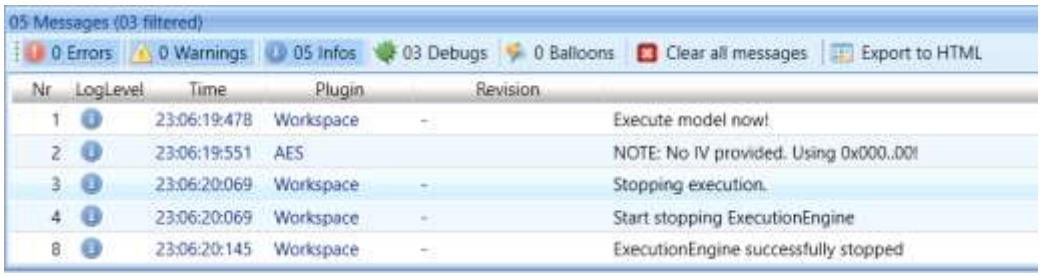

Fig 6: Encryption with AES - Time details (after applying filtering algorithm).

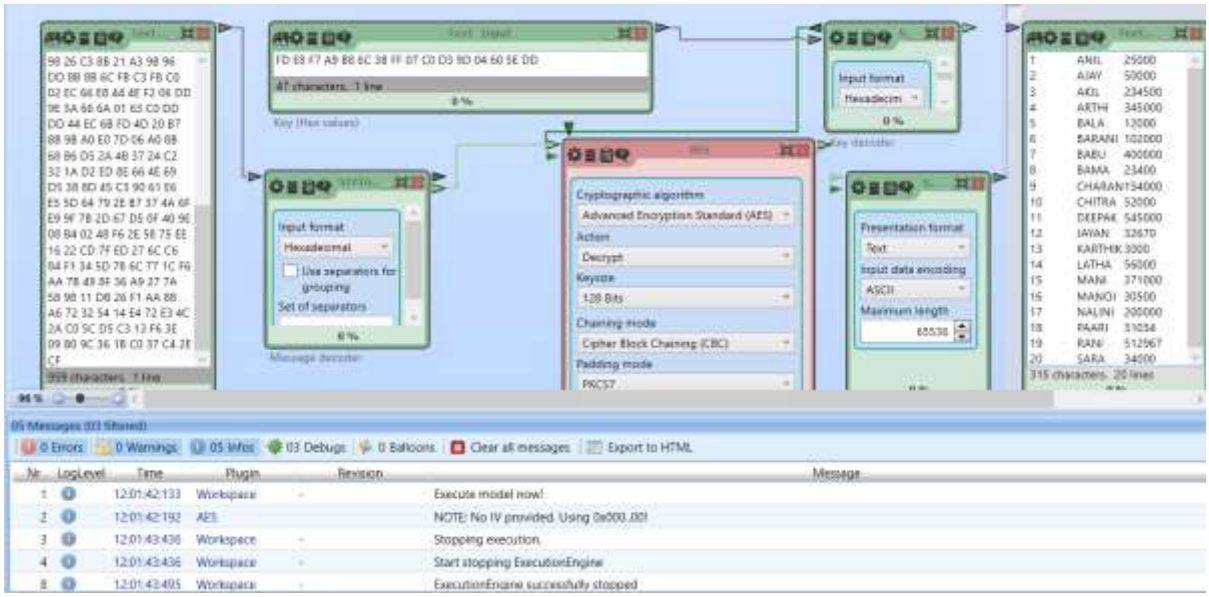

Fig 7: Decryption using AES

05 Messages (03 filtered)

0 Errors | 0 Warnings | 05 Infos | 03 Debugs | 0 Balloons | Clear all messages | Export to HTML

Nr	LogLevel	Time	Plugin	Revision	
1		12:01:42:133	Workspace	-	Execute model now!
2		12:01:42:192	AES		NOTE: No IV provided. Using 0x000..00!
3		12:01:43:436	Workspace	-	Stopping execution.
4		12:01:43:436	Workspace	-	Start stopping ExecutionEngine
8		12:01:43:495	Workspace	-	ExecutionEngine successfully stopped

Fig 8: Decryption using AES – Time details

Fig 9: Decryption (on conditional encryption with AES)

05 Messages (03 filtered)

0 Errors | 0 Warnings | 05 Infos | 03 Debugs | 0 Balloons | Clear all messages | Export to HTML

Nr	LogLevel	Time	Plugin	Revision	
1		12:17:22:978	Workspace	-	Execute model now!
2		12:17:23:094	AES		NOTE: No IV provided. Using 0x000..00!
3		12:17:23:506	Workspace	-	Stopping execution.
4		12:17:23:506	Workspace	-	Start stopping ExecutionEngine
8		12:17:23:589	Workspace	-	ExecutionEngine successfully stopped

Fig 10: Decryption time details (on conditional encryption with AES)

S.NO	INPUT FILE SIZE (KB)	ENCRYPTION TIME [BEFORE FILTERING] (in Sec)	ENCRYPTION TIME [AFTER FILTERING] (in Sec)	DECRYPTION TIME [BEFORE FILTERING] (in Sec)	DECRYPTION TIME [AFTER FILTERING] (in Sec)
1.	142	1.4	1	1	0.6
2.	169	1.6	1	1.3	0.7
3.	280	1.7	1.2	1.5	1
4.	425	1.9	1.3	1.6	1
5.	536	2	1.4	1.7	1.1

Table 3: Time taken to encrypt and decrypt sample files of different sizes with and without filtering operation.

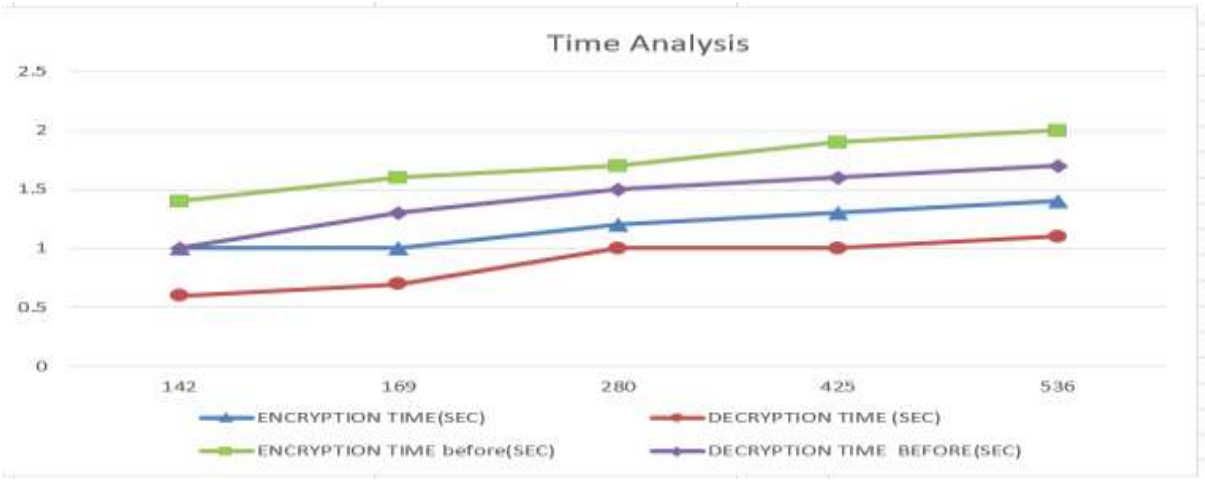

Fig 11: Graph showing the time analysis of encryption and decryption process with and without filtering operation.

5 Conclusion

In this system which uses AES algorithm with filtering approach, the selective data are given more security than normal data based on the requirement of the user. Also this method saves processing time and storage. As per the criteria set it may provide an efficiency improvement of about 30% or more. This is a varying factor based on the criteria set.

6 Future Work

As a future work, the efficiency of the above method can further be improved by incorporating new methods and techniques.

7 References

[1] Securing BIG Storage: Present and Future I. Bhargavi, T. MaruthiPadmaja, 2016 Online International Conference on Green Engineering and Technologies (IC-GET), 978-1-5090-4556-3/16©2016 IEEE

[2] Anup Mathew, "Survey Paper on Security & Privacy Issues in Cloud Storage Systems," EECE 571B, TERM SURVEY PAPER, APRIL2012.

[3] Gansen Zhao et al, “Trusted Data Sharing over Untrusted Cloud Storage Providers,” In Proc. of 2nd IEEE International Conference on Cloud Computing Technology and Science, 2010, pp. 97-103.

[4] H. Zhou and Q. Wen, "Data Security Accessing for HDFS Based on Attribute-Group in Cloud Computing," In Proc.of International Conference on Logistics Engineering, Management and Computer Science (LEMCS 2014), 2014, pp. 525-528.

[5] J. Cohen, S. Acharya, “Towards a Trusted Hadoop Storage Platform: Design Considerations of an AES Based Encryption Scheme with TPM Rooted Key Protections,” IEEE 10th International Conference on and Autonomic and Trusted Computing (UIC/ATC), Ubiquitous Intelligence and Computing, 2013, pp. 444 – 451

[6] S. Park, Y. Lee, “Secure Hadoop with Encrypted HDFS,” Chapter Grid and Pervasive Computing, Vol. 7861 of the Series Lecture Notes in Computer Science, pp 134-141, 2013

[7] S. Jin, S. Yang, X. Zhu, and H. Yin, "Design of a Trusted File System Based on Hadoop," In Proc. of Trustworthy Computing and Services, ed: YuyuYuan,Xu Wu, Yueming Lu, 2013, pp. 673-680.

[8] H. Y. Lin, S. T. Shen, W. G. Tzeng, B. S. P.Lin. “Toward Data Confidentiality via Integrating Hybrid Encryption Schemes and Hadoop Distributed File System,” In Proceedings of 26th International Conference on Advanced Information Networking and Applications, IEEE Computer Society Washington, DC, USA, 2012, pp. 740-747.

[9] Garima Saini and Naveen Sharma, “Triple security of data in cloud computing,” International Journal of Computer Science and Information Technologies, Vol. 5, No 4, pp.5825-5827, 2014.

[10] Comparative analysis of some selected cryptographic algorithms, Afolabi,A.O & Atanda,O.G, Computing Information Systems, Development Informatics & Allied Research Journal, Vol& No.2,June 2016.

[11] A Study on Big Data Security and Data Storage Infrastructure, SupriyaHaribhauPawar Department of Computer Engineering, BharatiVidyapeeth Deemed University, Pune, Maharashtra, India, Volume 6, Issue 7, July 2016 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering

[12] Big-Data Security, KalyaniShirudkar, DilipMotwani, Department of Computer Engineering VIT, Mumbai, India, Volume 5, Issue 3, March 2015 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering

[13] A Study of Encryption Algorithms AES, DES and RSA for Security By Dr.Prerna Mahajan &AbhishekSachdeva IITM, India, Global Journal of Computer Science and Technology Network, Web & Security Volume 13 Issue 15 Version 1.0 Year 2013, International Research Journal Publisher: Global Journals Inc. (USA) Online ISSN: 0975-4172 & Print ISSN: 0975-4350

[14] Secure Data Transmission For Multisharing in Big Data Storage, M. InduMaheswari, S. Revathy and R. Tamilarasi, Sathyabama University, Chennai - 600119, Tamil Nadu, India, ISSN (Print) : 0974-6846 ISSN (Online) : 0974-5645,Indian Journal of Science and Technology, Vol 9(21), DOI: June 2016

[15] A Comprehensive evaluation of cryptographic algorithms: DES, 3DES, RSA and Blowfish, Priyadharshinipatil, PrashantNarayankar, Narayan D.G, Meena S.M, International conference on information security and privacy(ICISP2015), 11-12 December 2015, Nagpur, India,

[16] A study on encryption decryption algorithm for big-data analytics in cloud, SreenivasaB.L, Manish Kumar, Mohammed Nueed Shaikh and Dr. S Sathyanarayana, International *Journal of Latest Trends in* Engineering and Technology Special Issue SACAIM 2016, pp. 323-329 e-ISSN:2278-621X