International Journal of Computer Applications (0975 – 8887)
Volume 129 – No.12, November2015

AES – MR: A Novel Encryption Scheme for securing Data in HDFS Environment using MapReduce

Viplove Kadre
M. Tech Scholar
Department of Information Technology
S.A.T.I College,
Vidisha (M.P), India

Sushil Chaturvedi
Assistant Professor
Department of Information Technology
S.A.T.I College,
Vidisha (M.P), India

ABSTRACT

Data security is an important issue as far as storage of sensitive data is concerned. Hadoop is usually utilized for storage, large amount of data using its storage technology, namely Hadoop Distributed file System HDFS. Hadoop by default does not contain any security mechanism but as it has grown very much and it is the first choice of the business analyst and industries to store and manage data it is necessary to introduce security solutions to Hadoop in order to secure the important data in the Hadoop environment. Authentication, Authorization, Data Encryption, Security against various attacks are the key levels of data and information security in the Hadoop environment. Efforts have been made in order to attain each and every level of security over the period of recent years. Kerberos is one such effort in order to attain Authentication and Authorization and it succeeded in doing so, but with the attackers having new technologies and hacking tools attackers can easily bypass the security provided by Hadoop's Kerberos Authentication system and then the data at storage level is unencrypted can easily be stolen or damaged which is a big concern.

Encryption of large data stored in HDFS is actually a process which takes a lot of time and this time consuming nature of encryption should be controlled by encrypting the data using a parallel method. This study discusses a new technique to perform encryption in parallel using AES-MR (an Advanced Encryption standard based encryption using MapReduce) technique in MapReduce paradigm..The time taken for Performing the encryption and decryption process is relatively less for user generated content. Results show that AES-MR encryption process is found to be faster with mapper function alone in comparison with running the encryption process under mapper function and reducer function. Here a new encryption scheme is given by the combination of AES and MapReduce in order to secure data in HDFS environment.The results generated for encryption on different data proves that the proposed technique is well suited for protecting user generated sensitive data deployed in the HDFS environment.

Keywords

Hadoop, Hadoop distributed file systems (HDFS), Data Encryption, MapReduce and AES-MR.

1. INTRODUCTION

Hadoop [9] is an open-source framework to store and process large amounts of data. Hadoop has been developed under an Apache License. It is implemented to scale from a single cluster of thousands of servers. Hadoop consists of two main modules: the MapReduce and the Hadoop Distributed File System (HDFS) [1]. Hadoop [5] was developed from GFS (Google File System) [2, 3] and MapReduce papers published by Google in 2003 and 2004 respectively. It has been popular recently due to its highly scalable distributed programming or computing framework, it enables processing big data for data-intensive applications as well as many analytics. Hadoop is a framework of tools which supports running application on big data and it is implemented in Java. In Hadoop Distributed File System, files are stored in the form of smallest unit blocks. When an input file is copied from the local file system to HDFS filesystem, the file is broken down into chunks. These chunks are known as blocks. For HDFS, this block size is quite large, where the default being 64 MB. This is configurable and can be changed by configuring the parameter dfs. Block. Size in core-site. XML configuration file. In order to ensure HDFS being fault tolerant, blocks are replicated across all the DataNode servers, and the default replication number is 3. The blocks in HDFS preserve the rack-awareness, policy, such that two copies of a block are placed in the same rack while another copy is placed in a different rack [1].

MapReduce is a programming model, proposed by Google, taking into account the Lisp charges "Guide" and "Decrease". Usage of this model is ordinarily utilized for preparing extensive data sets [2]. As the span of the accessible stockpiling gadgets increment in conjunction with the abatement in costs for the gadgets, it is more practical for commercial enterprises to store bigger arrangements of data. Likewise, as the multifaceted nature of data frameworks expands, so does the extent of data put away in mechanical applications. These commercial enterprises are tested with the need to proficiently examine terabytes or even petabytes of data because of the expanded data size delivered from these applications. There is a need to discover an answer that takes into account bigger measures of data to be broken down and the calculation to be appropriated among various machines to build the preparing proficiency. With terabyte hard drives turning out to be more financially savvy, organizations can stand to have their own particular bunches, as more open-source items touch base on the scene to help software engineers in their customization endeavors. Hadoop is an open-source Java execution of the MapReduce programming show that can be set up on a wide assortment of equipment stages [2] [4] [7]. There has been a lot of examinations that spotlights on the execution of Hadoop.

2. RELATED WORK

The Hadoop Distributed File System [1] was designed to store and process very large amounts of data in terabytes or petabytes, and provides high-throughput access to this data. Files are stored in a replicated fashion across more than one machine to ensure their durability and high availability to parallel applications. This work introduces the design of HDFS and instructions on how to operate it using map reduce programming. HDFS is a file system where files are broken into blocks of a fixed size.

MapReduce [2] is the programming model is used for processing and generating large scale datasets. Here map and reduce functions are specified by users. Map function process and generate a set of intermediate key/value pairs and reduce function merges all intermediate key values associated with the same intermediate key. The map and reduce function allows to perform automatic parallelized operation easily and re-execute the mechanism for fault tolerance on a large group of commodity machines. Distributed file system nodes simultaneously perform computing and storage operations. The large file in partitioned into a number of chunks and allocate it to distinct nodes to perform MapReduce task parallel over the nodes. Typically, MapReduce task processes on many terabytes of data on thousands of machines.

Park, S., Lee, J., Chung, T[10] have discussed the security model of Hadoop by using virtual private networks. Eavesdropping and network attacks were prevented and 7 every data node is authenticated. But the implementation of this approach is still in progress and not yet completed.

Prabhakar, R., Patrick, C., Kandemir, M, et.al [11] have proposed an integrated approach to protecting clouds/datacenters using DDOS defense mechanisms, access control techniques and piracy prohibition methods. The proposed security mechanism is inconvenient to the acceptance of web scale computing in the commercial world.

Abadi, D.J et.al [13] have proposed a layered structure for security in distributed storage and data layers. A Secure Co Processor (SCP) is utilized for e customer stockpiling of scrambled data in the cloud. High calculation cost and restricted memory turn into a badly designed while actualizing SCP.

Jam, M.R.; Khanli et.al [23] talked about Hadoop security issues and difficulties. The difficulties of security in Hadoop environment can be sorted into a validation level, data level, system level, and nonexclusive issues. They have likewise talked about methodologies like data encryption, system encryption, logging, hub support and algorithms for encryption procedures. The paper has taken a fast survey of ways to deal with understand the enormous data security issues alongside the essential properties of huge data. The latter piece of the paper has displayed the data encryption, cryptographic algorithms with the near investigation of these algorithms.

Becherer [26] have examined that Authentication remains a noteworthy security challenge in Hadoop environment. Hadoop does not unequivocally validate the customer. Subsequently, information hubs can be gotten to utilizing square areas. Crucial properties of a triangle and double servers to enhance the security level of Hadoop groups the watchword given by the client is translated and distanced into more than one unit utilizing the validation server and put away in numerous Backend Servers alongside the comparing username. The Authentication Server utilizes the qualities put away as a part of different Backend Servers to validate the client. Validation and Backend servers cooperate to confirm the client. The enlistment process and the verification procedure are facilitated as a web administration to confirm the clients before signing into the Hadoop group. The paper proposed three methodologies for security improvement in Hadoop environment in view of triangle properties. Kerberos Authentication Protocol was utilized as a part of request to give upper level of security that is Authentication yet this was insufficient to guarantee the information security of delicate information in HDFS environment.

B. Lake [29] proposed approaches for authentication service for Hadoop in a cloud environment. They used the properties of triangles for authenticating the user. To enhance the security level of Hadoop cloud, dual servers were used. If one server is hacked and theta value is modified, then user validation cannot be successful.

J. Zhao, L. Wang, J. Tao, J. Chen, W. Sun, R. Ranjan, et al. [40] have suggested that MapReduce is viewed as a satisfactory programming model for extensive scale information concentrated applications. The Hadoop structure is a surely understood MapReduce execution that runs the MapReduce undertakings on a bunch framework. G-Hadoop is an augmentation of the Hadoop MapReduce system with the usefulness of permitting the MapReduce assignments to keep running on different bunches. In any case, G-Hadoop essentially reuses the client validation and occupation accommodation system of Hadoop, which is intended for a solitary group. The work proposes another security model for G-Hadoop. The security model depends on a few security arrangements, for example, open key cryptography and the SSL convention, and is dedicatedly intended for appropriatiing situations. This security structure opens up the client's confirmation and employment accommodation procedure of the present G-Hadoop execution with a solitary sign-on methodology. Furthermore, the outlined security structure gives various diverse security components to shield the G-Hadoop framework from conventional assaults.

3. PROPOSED WORK

3.1 AES - MR

AES-MR is only the mix of the AES encryption algorithm and MapReduce parallel programming Paradigm. The AES encryption algorithm is one of the best, quickest conceivable approaches to encode the data very still, which is our goal in our work. With the MapReduce programming model it is less demanding to incorporate the AES encryption algorithm to work in parallel and spare a great deal of time. MapReduce will be utilized as a part of our work to encode the huge data volumes and vital data utilizing the AES encryption algorithm as a part of XTX mode. The broadly utilized IEEE 1619-2007 standard is XEX-TCB-CTS (XTS) [22] mode in which key material for XTS-AES comprises of an encryption key and in addition a change key that is utilized to consolidate the coherent position of the data hinder into the encryption. XTS yields tend to deliver autonomous yields which prompt the parallelization of such procedure. XTS is a solid instantiation of the class of change capable square figures. The XTS mode permits parallelization and pipelining in figure executions. It empowers the encryption of the last deficient piece of data while different modes are not having this office. Guide Reduce is by all accounts an alluring, financially savvy answer for substantial scalingdata, preparing administrations like securing data in the cloud through piece encryption [36]. MapReduce can fit in any sort of environment like shut or open frameworks. The system is intended for composing applications that quickly prepare endless data amid runtime on process groups. The code consequently segment information, data, performs planning, checking and have the adaptation to non-critical failure instrument through which it re executes the fizzled errands in a thing of vast bunch machines.

3.2 Parallelizing encryption using MapReduce Process

As expressed some time recently, the fundamental advantage of this programming worldview is the simplicity of parallelization. Since every mapper just works on one and only tuple at once, the framework can have numerous occasions of our administrators on distinctive tuples in Uralian parallel. After the guide step, the framework parcels the arrangement of tuples yield all through the examples of which are made in light of their key. That is, part I of the segment have all keys; worth matches that have key ki. Since reducer pr just works on one piece of this allotment, the framework in light of the application makes numerous occasions of PR running in distinctive parts in parallel. The information will be put away as coterminous squares and each piece is spoken to by an interesting square id (Io, IjJ1. 2.... 1n_1A). MapReduce incorporates set of mappers (M1, M2..... M,) and reducers (R1, R2..... Rr).The info is given to mapper as <block id, object>.This item is the substance of the HDFS (Hadoop Distributed File System) square or the information put away in the comparing piece id. Amid encryption the mapper and the reducer capacity in view of the <key; value> pair as talked about above. The figure underneath demonstrates the execution of AES XTX mode in parallel.

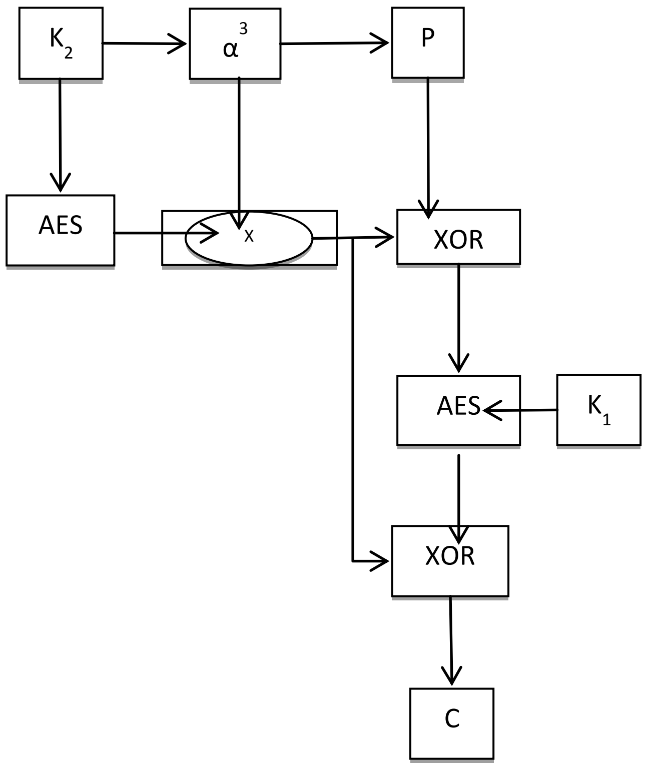

Fig 1 Operation of AES XTX Mode

- **Execution of the mapper:** Block <1r_jJobjecdts1> is given to mapper M; The mapper will produce the relating encoded yield 1'Rand send it to the reducer RR let. W = {<1o, object1>, <1lJobject2>, <12, object3..>... <I, lJobject n>} then broad configuration is spoken to as. R, = 1r-1-W<block id. ~o«bbleloctc>kid, Enc-compobject»•
- **Execution of reducer:** The gathered yields from different mappers are composed to the plate in the consecutive request (1\,1'2......1'n).In the proposed work the calculation is outlined in such a path for performing so as to secure vast information sets square encryption took after by pressure under every mapper and consolidating the outcome and putting away it on HDFS (Hadoop Distributed File System).The Mapper peruses piece of equivalent size which can be enhanced in light of the accessible free hubs in the cloud environment as demonstrated as : -

$$\boldsymbol{BLOCK\ SIZE} = \frac{\boldsymbol{INPUT\ DATA\ SIZE}}{\boldsymbol{NO.OF\ NODES\ CONFIGURED\ FOR\ MAPPER}} \ldots\ldots (1)$$

$$\boldsymbol{B} = \frac{\boldsymbol{I}}{\boldsymbol{N}} \qquad \ldots\ldots\ldots\ldots (2)$$

In our case the number of Mappers configured to do the task is equal to 1.

$$\boldsymbol{No.OF\ MAPPERS\ NODES}(N) = \mathbf{1} \ldots\ldots\ldots (3)$$

Therefore, in this case we can say that

$$\boldsymbol{BLOCK\ SIZE}(\boldsymbol{B}) = \boldsymbol{INPUTDATA\ SIZE}\ (\boldsymbol{I}) \ldots (4)$$

The figure shows how HDFS contents got assigned to mapper where mapper performs encryption followed by compression and the reducer is used to write the contents onto HDFS (i.e., output). Though HDFS (Hadoop Distributed File System) supports record level and block level compression of input data, in the proposed method after encryption under each mapper, a compression through gzip is done in order to reduce the storage space.

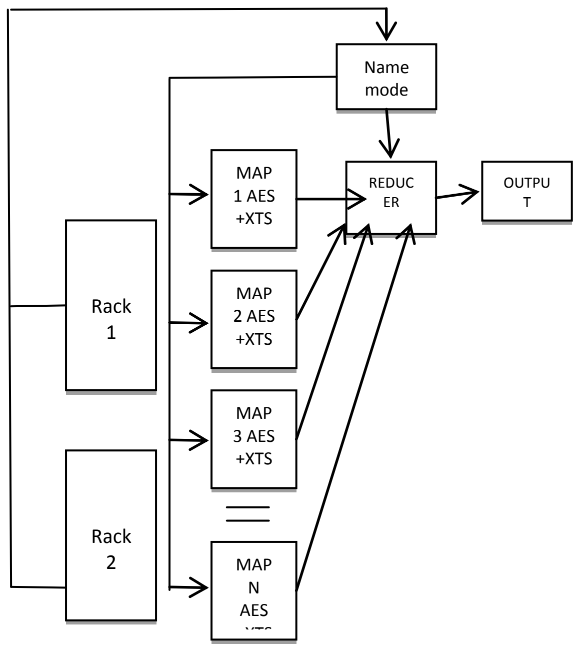

Fig 2 Operation of AES – MR in XTX mode

3.3 Execution of AES – MR-

(1) Encryption

AES-MR is used to encrypt the data in the large Datastores using the parallel mode of the AES encryption algorithm with the parallel workers namely mappers and reducers. The data are taken from the detector which can be widely considered as an HDFS in the form of the chunks which are passed on to the mapper where the chunks are encrypted one by one in parallel using the AES XTX mode. These encrypted chunks forward to the reducer phase, which sums up and combines the encrypted chunks from the mappers and combines it to form the complete encrypted chunk which is stored in the Datastore. AES-MR is the combination of MapReduce and AES encryption algorithm in XTX mode. Here we utilized the capabilities of MapReduce to encrypt the data files from the data store with the help of AES to increase the security of the sensitive data. The working of AES-MR is described step by step asunder-

- The data are taken from the Datastore in the form of chunks of fixed sizes.
- These chunks are then forwarded to the MapReduce for the process of encryption.
- The Map function contains the code for the encryption, AES in XTX mode, which encrypts the data chunk by chunk in parallel and converts the chunks into encrypted chunks.
- These encrypted chunks are forwarded to the Reducer function of the Reduce phase, which combines all the encrypted chunks in a single encrypted file.
- This single encrypted file of the original file is stored in the Datastore and the process of encryption is completed.

(2) Decryption
In the Decryption phase the opposite process of what has happened in the encryption phase happens. Decryption phase is also important as encryption where the encrypted file must be retrieved as it was in the original. So in AES-MR each map function is configured in such a way that it decrypts the entire data encrypted file chunk by chunk in the Map phase and forwards the result to the reducer phase to retrieve the original file. The step by step execution of decrypts in the AES-MR is described as below-

- The input the decryption phases are taken from the Datastore in the Encrypted format.
- This encrypted file is broken into chunks and then forwarded to the MapReduce functions Mapper class
- The Map function of the Mapper class contains the decryption code which decrypts the encrypted chunks one by one in parallel and converts it to plain data format
- These unencrypted data chunks are then forwarded to the Reducer function of the Reduce class which sums up and combines the unencrypted simple data chunks into a single unencrypted file.
- This single unencrypted file is again stored in the data store and can be viewed easily.

In the decryption phases the opposite process of what has happened in the encryption phase happens. The encrypted data is divided into the encrypted data chunks and forwarded to the MapReduce phase, which decrypts the data into simple file and then stores it into the datacenter. Both of these phases are diagrammatically depicted as under: -

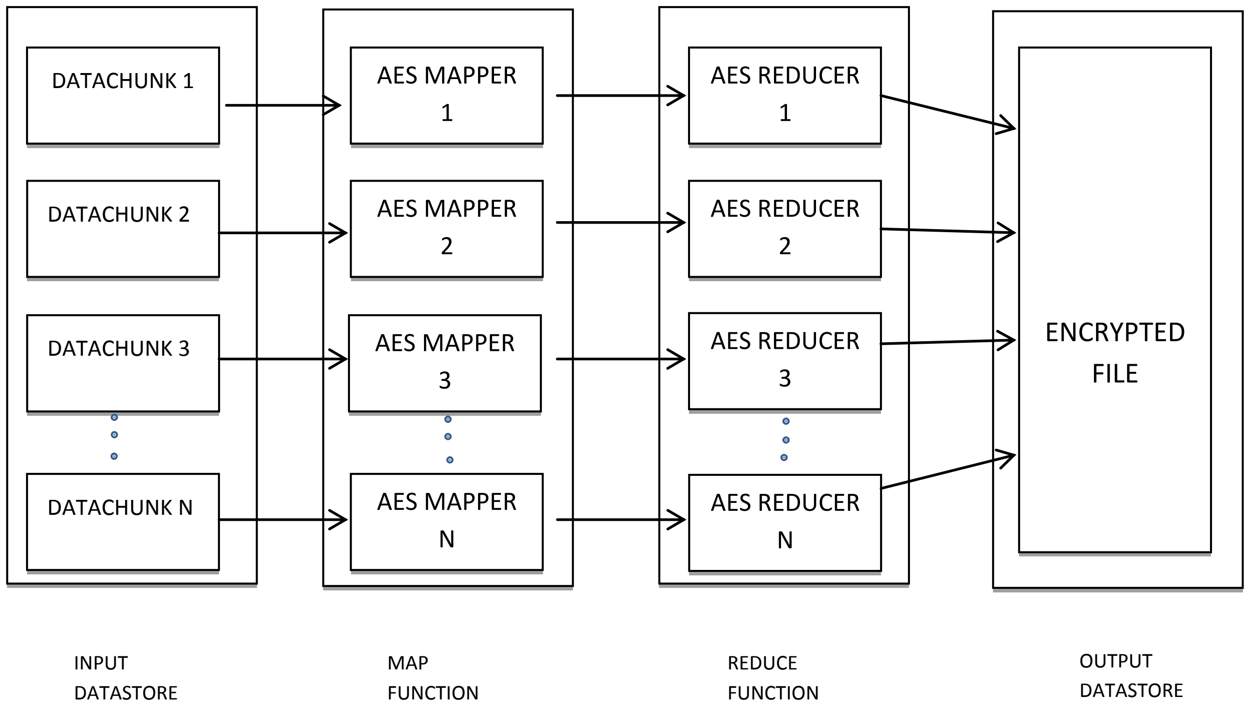

FIG 3 Encryption of Data Chunks Using AES –MR to generate Encrypted File

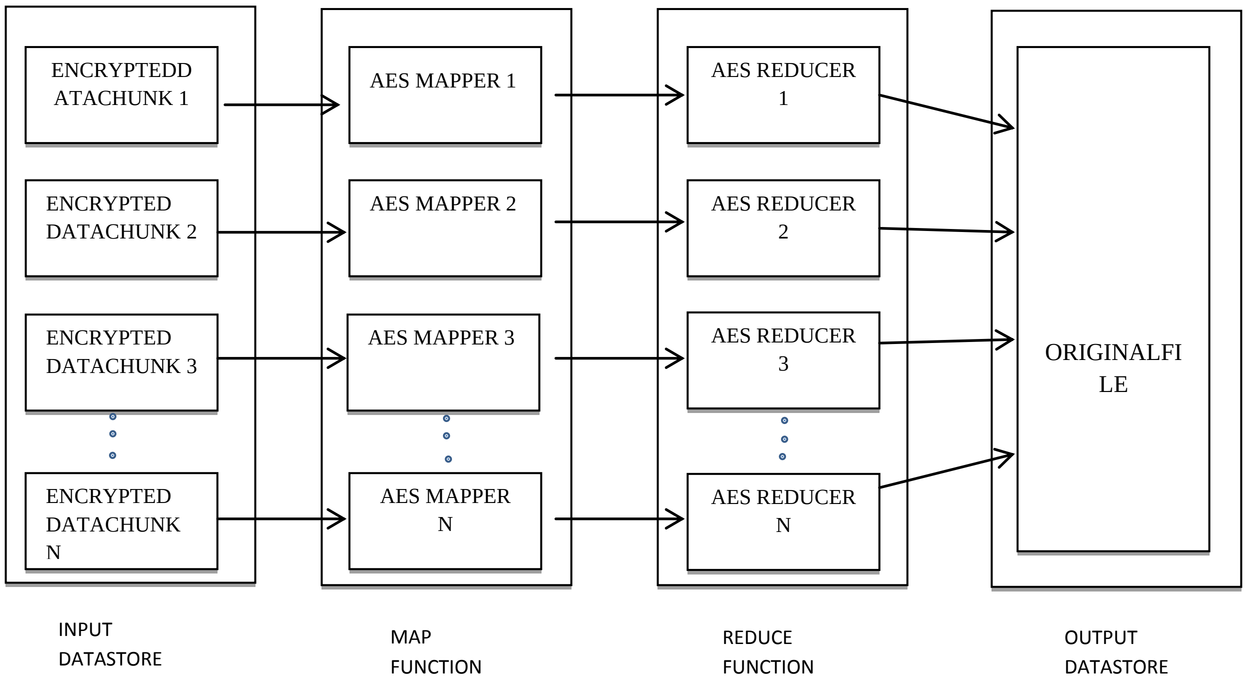

FIG 4 Decryption of Encrypted Data Chunks Using AES –MR to generate Original File

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Simulation Environment

We have evaluated the performance of our algorithm to rebalance the load for distributed file system through simulation on MATLAB r2015a implemented on Intel (R) Core (TM) i3 CPU M 370@ 2.40 GHz, 4GB RAM and 64 bit Window 7 home basic operation system .The datasets which are used for the experiment are created randomly of various sizes for the experimental results. Various experiments are carried out to calculate the average values of the parameters evaluated for performance and comparisons.

(1) Evaluation parameters

Following are the parameters used for evaluating performance of the proposed scheme:

The essential levels of the security are **Authentication, Authorization, Auditing and Data Encryption** basically it is named as **a 3ADE level** of security. Each and every level of security has its own weighted and importance as far as the overall security architecture of an application in concerned. In Hadoop also these are the levels of security at the various stages. The details of these levels of security are Authentication, Authorization, Auditing, Encryption and Security from the External Attacks and are depicted asunder:-

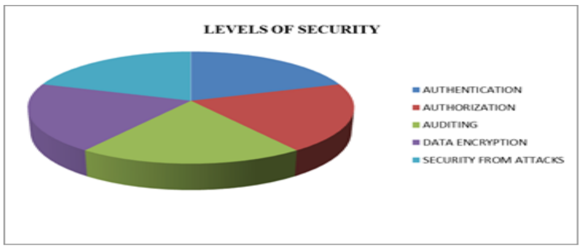

Fig 5 Various Levels Of Security

Kerberos and other security protocols provide only an Authentication and Authorization and rest all security levelsare untouched, which are depicted by the pie chartasunder:-

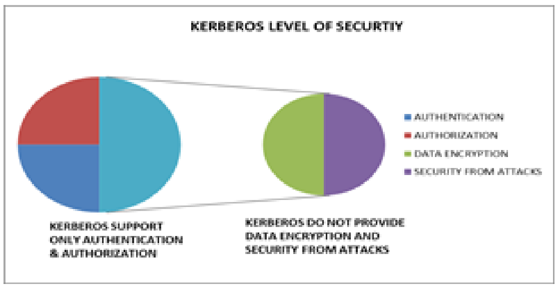

FIG 6 Limitations of Kerberos Authentication Protocol

3 Experimental results

AES-MR provides data encryption, which in turn provides security against various external attacks as encryption is not easy to break and get the actual data. There are no such kind of parameters which are considered while checking the results which are provided by AES-MR but us in our work calculated the **Encryption Time**, **Decryption Time**, **Total Time** taken by **AES-MR** to encrypt and decrypt the data, Overall **Performance** of **AES-MR** and at last we have provided a pie chart depicting shortcomings of Kerberos which are removed of overcome by AES-MR technique which are also shown as under-

1. AES-MR Data Encryption Time-

It is the time taken by our work AES-MR to encrypt the data completely. The data are taken from the Datastores and then it is forwarded to AES-MR where the mapper function execute the task of encrypting the chunks in parallel, then these encrypted chunks are forwarded to the reducer function where it sums up all the encrypted chunks to give the final output encrypted file. Thus the total time elapsed from picking up data from the data store to finally storing the encrypted data file in the Datastore is known as AES-MR encryption time. A graph below shows the encryption time of various data size files such as 1MB, 25MB, 50MB, 75MB, 100MB by AES-MR-

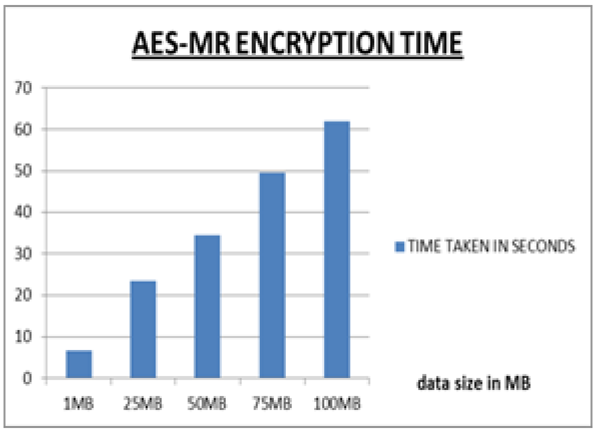

FIG 7 Totaltime taken by AES – MR to Encrypt Data

AES-MR Data Decryption Time-

It is the time taken by our work AES-MR to decrypt the data completely. The encrypted data are taken from the Datastores and then it is forwarded to AES-MR where the mapper function execute the task of decrypting the chunks in parallel, then these decrypted chunks are forwarded to the reducer function where it sums up all the decrypted chunks to give the final output decrypted original file . Thus the total time elapsed from picking up data from the encrypted Datastore to finally storing the original decrypted data file in the Datastore is known as AES-MR decryption time. A graph below shows the decryption time of various encrypted data size files such as 1MB, 25MB, 50MB, 75MB, 100MB by AES-MR-

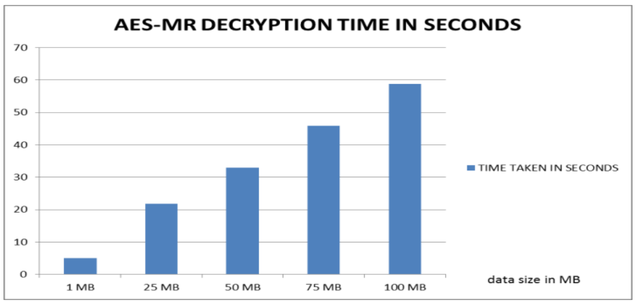

Fig 8 Times Taken by AES – MR to decrypt Data

4. Performance of AES-MR-

The overall performance of the AES-MR is being measured by our team in terms of the time taken by it for everything i.e. encryption of the data chunks, decryption of encrypted data chunks to the original and the overall total time for all these operations. The below table depicts the overall performance and its corresponding graph shows the AES-MR performance.

Table 1 For The Performance Evaluation Of Aes-Mr

DATA SIZE IN MB	TIME TAKEN FOR ENCRYPTION	TIME TAKEN FOR DECRYPTION	TOTAL TIME TAKEN
1 MB	6.79	5.06	11.85
25 MB	23.48	21.85	45.33
50 MB	34.56	32.89	67.45
75 MB	49.59	45.82	95.41
100 MB	62.04	58.75	120.79

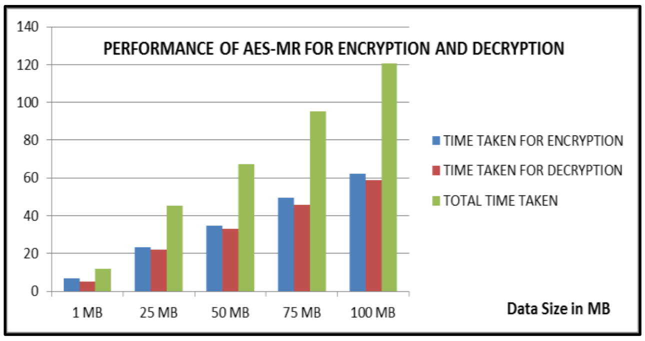

FIG 9 Performance Evaluations of AES – MR

Thus With the usage of AES-MR we can increase the security of data by encrypting the data chunks using the map and reduce the function of MapReduce with the AES encryption algorithm in the XTX mode. This is the fastest method of encrypting the Data chunks, and securing important data in the Hadoop Datastore. The shortcoming of Kerberos protocols were that it does not offer Data encryption and security against the attackers is being overcome by this method of encrypting the data with AES-MR. Here we have considered only 4 main levels of security, namely Authentication, Authorization, Data Encryption and Security against the attackers.Thus With the usage of AES-MR we can increase the security of data by encrypting the data chunks using the map and reduce the function of MapReduce with the AES encryption algorithm in the XTX mode. This is the fastest method of encrypting the Data chunks, and securing important data in the Hadoop Datastore. The shortcoming of Kerberos protocols were that it does not offer Data encryption and security against the attackers is being overcome by this method of encrypting the data with AES-MR. Here we have considered only 4 main levels of security, namely Authentication, Authorization, Data Encryption and Security against the attackers.

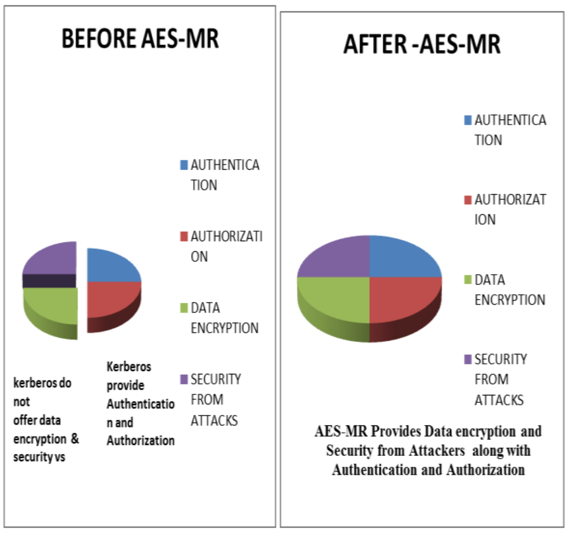

FIG 10 Security enhancement Before and After AES - MR

5. CONCLUSION

In this work we tried to solve this issue of data security at the storage level at HDFS by encrypting the data using AES-MR, where we have combined AES the fastest encryption algorithm with MapReduce in order to encrypt the data in a faster manner. The timing graphs show that the AES-MR encryption technique is fast enough to encrypt the data chunks from the data store using parallel processing of map and reduce functions. Thus with the use of AES-MR we have not only enhanced the security of important data at HDFS level, but with the help of parallel processing we can do it with a faster manner as well. The motive of this work was to ensure the security of important data at an HDFS storage level which has not been achieved by Kerberos only, but with our work along with Kerberos ensures security at each and every level and makes it almost impossible for the attacker to get the data which has been achieved in our work. In our simulation the work is carried out by a single worker this work can be done with even more speed if we increase the number of workers which is possible in the real world. The chunk sizes can also be distributed with the use of large numbers of workers, which in our case was the data size itself. Overall, if this work gives the data security a new height along with the use of Kerberos to attain all the levels of security that too at a faster speed, AES-MR is a good approach to carry out his work and attain data security.

6. REFERENCES

[1] "Hadoop Distributed File System", http://hadoop.apache.Org /hdfs.

[2] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *in Proc. 6th Symp. Operating System Design and Implementation (OSDI'04)*, pp. 137–150, Dec. 2004.

[3] Hadoop Distributed File System, "Working of HDFS Blocks," http://developer .yahoo.com/hadoop/tutorial/module2.

[4] S. Ghemawat, H. Gobioff, and S.-T. Leung,. "The Google File System*," in Proc. 19th ACM Symp. Operating Systems*, pp. 29–43, Oct 2003.

[5] Apache Hadoop, http://hadoop.apache.org/.

[6] HDFS Federation, http://hadoop.apache.org/common/docs/r0.23.0 /hadoop-yarn/ hadoop-yarn-site/Federation.html.

[7] Dean, J., Ghemawat, S., "MapReduce: Simplified Data Processing on Large Clusters,"*Communications of the ACM - 50th anniversary issue: 1958 - 2008, volume 51, issue 1 (January 2008), pg 107-113*

[8] Shvachko, K., Kuang, H., Radia, S., Chansler, R., "The Hadoop Distributed File Sys- tem," *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), p.1-10, May 03-07, 2010*

[9] White, T, Hadoop: The Definitive Guide. *O'Reilly Media, 2009.*

[10] Park, S., Lee, J., Chung, T., "Cluster-Based Trust Model against Attacks in Ad-Hoc Networks," *Convergence and Hybrid Information Technology, 2008. ICCIT. '08. Third International Conference on , vol.1, no., pp.526-532, 11-13 Nov. 2008*

[11] Prabhakar, R., Patrick, C., Kandemir, M., "MPISec I/O: Providing Data Confidential- ity in MPI-I/O," *Cluster Computing and the Grid, 2009. CCGRID '09. 9th IEEE/ACM International Symposium on , vol., no., pp.388-395, 18-21 May 2009*

[12] National Institute of Standards and Technology, Federal Information Processing Stan- dards Publication 197, "Advanced Encryption Standard," *November 26 2001*

[13] Abadi, D.J."Data Management in the Cloud: Limitations and Opportunities". *IEEE Data Eng. Bull., 32(1), 3-12.* 2009

[14] Chaiken, R., Jenkins, B., Larson, P. Å., Ramsey, B., Shakib, D., Weaver, S., & Zhou, J. "SCOPE: easy and efficient parallel processing of massive data sets". *Proceedings of the VLDB Endowment*, *1*(2), *1265-1276.2008*

[15] Chen, Y., Paxson, V., & Katz, R. H. "What's new about cloud computing security". *University of California, Berkeley Report No. UCB/EECS-2010-5 January*, *20*(2010), 2010-5.

[16] Dworkin, M. *Recommendation for block cipher modes of operation: The CMAC mode for authentication*. "US Department of Commerce, Technology Administration, National Institute of Standards and Technology".2005.

[17] El-Fotouh, M. A., & Diepold, K. "Statistical Testing for Disk Encryption Modes of Operations". *IACR Cryptology ePrint Archive*, *2007*, 362. 2007.

[18] Foster, I., Zhao, Y., Raicu, I., & Lu, S. "Cloud computing and grid computing 360-degree compared". In *Grid Computing Environments Workshop, 2008. GCE'08* (pp. 1-10). IEEE. November 2008.

[19] Gropp, W., Lusk, E., & Thakur, R. *Using MPI-2: Advanced features of the message-passing interface*. MIT press. 1999.

[20] Kaufman, L. M. "Data security in the world of cloud computing. *Security & Privacy", IEEE*, *7*(4), 61-64. 2009.

[21] Keller, S. S., & Hall, T. A. (2010). "The XTS-AES Validation System (XTSVS)".

[22] Martin, L. "XTS: A mode of AES for encrypting hard disks". *IEEE Security & Privacy*, (3), 68-69. 2010.

[23] Jam, M.R.; Khanli, L.M.; Javan, M.S.; Akbari, M.K., "A survey on security of Hadoop," in *Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on* , vol., no., pp.716-721, 29-30 Oct. 2014

[24] Yu Xianqing; Peng Ning; Vouk, M.A., "Enhancing security of Hadoop in a public cloud," in *Information and Communication Systems (ICICS), 2015 6th International Conference on* , vol., no., pp.38-43, 7-9 April 2015

[25] O'Malley, O., Zhang, K., Radia, S., Marti, R., & Harrell, C. "Hadoop security design". *Yahoo, Inc., Tech. Rep*.2009

[26] Becherer, A.Hadoop Security Design: Just Add Kerberos? Really? iSEC Partners. *Inc.: San Francisco, CA, USA*.2010

[27] Yuan, M. "Study of Security Mechanism based on Hadoop". In *Information Security and Communications Privacy*, *6*, 042.2012

[28] Wang, L., Tao, J., Ranjan, R., Marten, H., Streit, A., Chen, J., & Chen, D. “G-Hadoop: MapReduce across distributed data centers for data-intensive computing”. In *Future Generation Computer Systems*, *29*(3), 739-750.

[29] Lakhe, B. Introducing Hadoop Security. In *Practical Hadoop Security* (pp. 37-47).

[30] Lin, H. Y., Shen, S. T., Tzeng, W. G., & Lin, B. S. P. “Toward data confidentiality via integrating hybrid encryption schemes and Hadoop Distributed File System”. In *Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on* (pp. 740-747). IEEE. March 2012.

[31] White, T. *In Hadoop: The definitive guide*. " O'Reilly Media, Inc.".2012

[32] Roy, I., Setty, S. T., Kilzer, A., Shmatikov, V., & Witchel, E. Airavat: “Security and Privacy for MapReduce”. In *NSDI* (Vol. 10, pp. 297-312).April 2010.

[33] Bhatt, P., Toshiro Yano, E., & Gustavsson, P. M. “Towards a Framework to Detect Multi-stage Advanced Persistent Threats Attacks”. In*Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on* (pp. 390-395). IEEE. April 2014.

[34] P. Mell and T. Grance, "Draft NIST working definition of cloud computing," *Referenced on June. 3rd,* vol. 15, 2009.

[35] 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.

[36] M. Yuan, "Study of Security Mechanism based on Hadoop”. *Information Security and Communications Privacy*, vol. 6, p. 042, 2012.

[37] V. Shukla, "Hadoop Security Today & Tomorrow," ed: Hortonworks Inc., 2014.

[38] *Comprehensive and Coordinated Security for Enterprise Hadoop*. Available: http://hortonworks.com/labs/security/ 2014

[39] L. Wang, J. Tao, H. Marten, A. Streit, S. U. Khan, J. Kolodziej, *et al.*, "MapReduce across distributed clusters for data-intensive applications," in *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International*, 2012, pp. 2004-2011.

[40] J. Zhao, L. Wang, J. Tao, J. Chen, W. Sun, R. Ranjan, *et al.*, "A security framework in G-Hadoop for big data computing across distributed Cloud data centres," *Journal of Computer and System Sciences*, vol. 80, pp.994-1007, 2014