International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958 (Online), Volume-9 Issue-5, June 2020

MapReduced Based: A New Stream Cipher Technique for Data Encryption

Galal A. AL-Rummana, G. N. Shinde, Abdulrazzaq H. A. Al-Ahdal

Abstract: ***Nowadays, data keeps increasing; this in turn makes big data one of the hot topics in the modern era of technology. The biggest challenge, however, is big data security and cryptography is one of the most secure techniques. In this proposed model, we use this technique to secure data via a proposed new stream cipher technique to process more than one block by dividing the total size of the block into two parts, and swapping them then, combine and apply XOR operation with key and make some of mathematical operation. This operation is of fifteen rounds which make it very difficult for attacks to guess the plaintext.***

Keywords **: Big data, Encryption, Cryptography, Stream cipher, Security, Hadoop, MapReduce.**

I. INTRODUCTION

This is the era of digital information within which data is incredibly crucial, therefore the world depends on data. With the enormous development of data, digitalization, technology and information technology tools, a large quantity of data generated quickly [1]. Data is the raw productive material, a new source of great economic and social interest. In addition, population growth, devices and sensors are now connected to the Internet, which has altered the capacity to produce, interact, exchange and access data. Data creates a lot of social and economic interest deriving from creativity, efficiency, skills and development [2].

Big data is a new and important issue in the field of academic research as well as in the field of industry research. Big Data is that term which is used to define huge volumes of structured, semi structured and unstructured data, Due to its high value, variety, volume, velocity, variability, and veracity, it is very complex to process this kind of data through using normal databases and software technologies. Through the present technology is also going to rule the world in future [3].

There are two techniques for securing big data: cryptography and steganography, both are very similar. Steganography uncovers the communication traces whereas cryptography uses encryption in order to make the message incomprehensible. The steganography makes no changes in the structure of the message, whereas the cryptography changes the standard confidential message structure when transferred through the network. In other hand you can say cryptography means secret writing and steganography is known as cover writing, cryptography is implemented on text file and steganography is implemented on audio, video, image and text [4].

The cryptography provides a lot of encoding styles in order to achieve the security in communicating through a public network. The term cryptography is a Greek word. It is used to signify “secret writing”. Cryptography is best explained through this example: a sender sends a message in plaintext, the message gets encrypted and converted into a cipher text before being transmitted over the network. When this message is received by the receiver, it gets decrypted back into the plaintext [5].

Cryptography can be categorized into tow type Symmetric key cryptography. This kind of cryptography utilizes a one key in order to respectively encrypt and decrypt the cipher text and plain text. The case here is that both encryption and decryption have the same key, and consume less time.
Asymmetric key cryptography we have two keys which are used at this time, the public key, that the sender provide by the sender for the encryption of the message, and the private one, which is applied by the receiver for the decryption of the message. The keys can be used again with different entities [6].

Recently, many cryptographic algorithms have been used to secure data being sent via internet. However, there are many cryptographic algorithms that are used for data encryption which can be described as follows: Data Encryption Standard (DES) invented in 1974 by IBM and commonly used symmetric key. There are different suggested algorithms which have lately proven that the DES algorithm is not secured DES due to using a small key size that results in not strong security and which can be easily broken. Additionally, particular drawback, DES also works slowly on software [7].

International Data Encryption Algorithm (IDEA), designed by James L. Massey of ETH Zurich and Xuegialai as block cipher algorithm in 1991 [8]. In fact the algorithm came under some changes that have become known later as IDEA. As a matter of fact, IDEA is also symmetric key and contains more numbers of not strong keys, this is considered as its main disadvantages. Moreover, a novel attack on round six of IDEA is detected.

Advanced Encryption Algorithm (AES), uses symmetric key known as O.S. FIPS NIST in 2001, an unbreakable algorithm with strong competency. However, AES has some drawbacks for example,

Revised Manuscript Received on June 11, 2020.
* Correspondence Author
Galal A. AL-Rummana *, School of Computational Sciences, S.R.T.M. University, Nanded, India. Email: galal300z@gmail.com
G. N. Shinde, Yeshwant College, Nanded, Maharashtra, India. Email: shindegn@yahoo.co.in
Abdulrazzaq H. A. Al-Ahdal, School of Computational Sciences, S.R.T.M. University, Nanded, India. Email: alahdal201211@gmail.com

it requires extra processing and needs extra rounds of communication when compared to DES [9]. Blowfish is one of the most popular types have 64 – bit symmetric block cipher with variable length key, which is distinguished by its efficiency and its guarantee in making the encryption process safe, strong and completely difficult to be hacked till date [10].

For securing big data, such tools as Hadoop and MapReduce are used for efficiency purpose. Hadoop, developed under an Apache License, is an open source distributed processing framework for sorting data and working applications on clusters of commodity hardware. It offers huge storage for all kinds of date, many processing power and the chance to handle limitless tasks and jobs at the same time. It occupies the center of an ecosystem of big data technologies that are mainly utilized to enhance advanced analytic initiatives like predictive, data mining and machine learning analytics. Hadoop systems deal with various structured and unstructured data. This makes it more flexible for the users to collect, process, and analyze than databases and data warehouses provide [11].

A Map Reduce program contains a map procedure, which conducts filtering, sorting, and reducing method, which performs a summary operation. It considers as a software framework that comfortable to deal with huge, long - running jobs that cannot be grip within the reach of a single request. It is useful for distributed processing with an efficient process and general data sets on the computing cluster, it used to decrease the cost of security by automatically splitting input data into some parts, then, run a program parallel on that data parts with handle most of the problems at once, such consistency and fault tolerance. It is a useful framework for big data to deal with high efficiency and guaranteed handling of a large cluster environment [12]. The mechanism of MapReduce is shown in fig. (1).

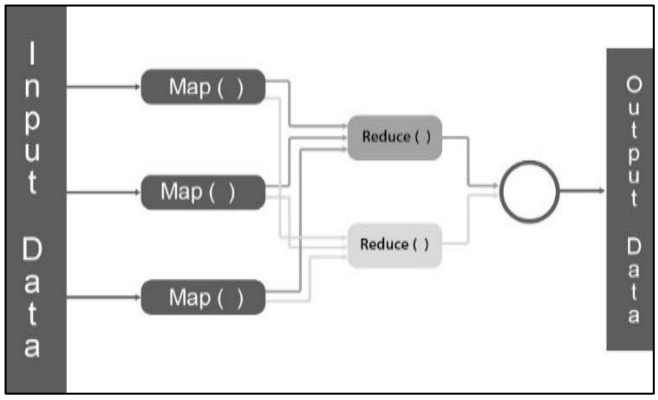

Fig.1. Mechanism of MapReduce

II. Related Work

AES-MR (Advanced Encryption Standard using Map Reduce) this technique is designed by Viplove Kadre and Sushil Chaturvedi [13]. They have used the new technique to conduct encryption process in parallel. The time needed in order to perform the encryption and decryption process is in comparison less for generated content of the user. This technique is suitable for protecting generated sensitive data of user deployed in the Hadoop Distributed File System (HDFS) environment. It is a blend of the AES encryption algorithm and Map Reduce parallel programming paradigm, which is a less demanding model to incorporate the Advanced Encryption Standard encryption algorithm in order to work in a parallel way and save time.

Bull eye algorithm is given by B. Saraladevi and N. Pazhaniraja [14]. It has been proposed for the purpose of maximizing security in HDFS. That is, it is presented to securing the sensitive like credit card numbers, account numbers, passwords, personal details …etc. This algorithm is introduced on Hadoop module so as to view all the sensitive information in 360° so that it is possible to see whether all the secured information is stored in a saved manner, and make sure that the authorized person can preserve personal information probably [15].

J. Zhao, et al. [16]. describes a framework to increase G Hadoop security by using security solutions like public key cryptography and Secure Socket Layer (SSL). G Hadoop is an augmentation of the Hadoop MapReduce system with the advantage of allowing the MapReduce assignments keep running on different clusters.

Recently, many researchers have dealt with big data security as M. Tharayil [17]. has proposed a general architecture for providing the security for big data and simultaneously enhances the data performance in motion. Moreover, S. Li. [18]. has proposed a sticky policy framework for security of big data by using a loose couple binding for data and their respective sticky policies. Also, Sanket Desai, et al. [19]. have improved the encryption performance by using MapReduce as a programming model to encrypt large amounts of data in a parallel distributed fashion.

III. The Proposed Method

A new stream cipher cryptographic algorithm is proposed to process a large quantity of data by processing more than one block of 256 bit. The processes followed are: First, the total size of the block is divided into two parts and be swapped. Second, combine the two parts and apply a simple logical operation XOR operation with 256 bit key size. Then, divide the block into 16 parts of 16 bits. Third, divide each part of 16 into two parts, the left part has 8 bits and right part has 8 bits. After that, collect all left part together and the right part. Finally, combine left part with right part in to one block. The processes are repeatedly performed for 15 times to make it difficult for the attackers to guess the plaintext. Fig. (2) Shows the proposed model for encryption and Fig. (3) Shows the proposed model for decryption.

A. Algorithm

Encryption procedure:

1. Divide 256 bits (X) to (XL) and (XR) both of 128 bit.
2. Swap (XL) and (XR) and,
3. Merge (XL) and (XR) to the (X) again.
   For the first round do the following
   - X=X^Ki
   - Divide X to 16 parts each of 16 bits each have (a) and (b)
   - Divide each part to two equal array (a) and (b) each of 8 bits
   - Recombi
     ne
     XL=a1+a

2+a3+…..+a32

- Recombine XR=b1+b2+b3+…..+b32
- Swap (XL) and (XR)
- Recombine XL and XR
- Repeat step 3 for 15 time

i=14

Decryption procedure:

1. Divide 256 bit (X) to (XL) and (XR) both of 128 bit.
2. Swap (XL) and (XR) and,
3. Divide each part (XL) and (XR) into equal array each has 8 bit
4. Recombine each 8 bit into 16 part each of 16 bit as following
   (XL1+XR1), (XL2+XR2), …….., (XL32+XR32)
5. Recombine 16 part into (X)
6. X=X^Ki
7. Repeat step (1) to step (6) for 15 round
8. Divide (X) to (XL) and (XR)
9. Swap (XL) and (XR)

Merge (XL) and (XR) to the (X) again.

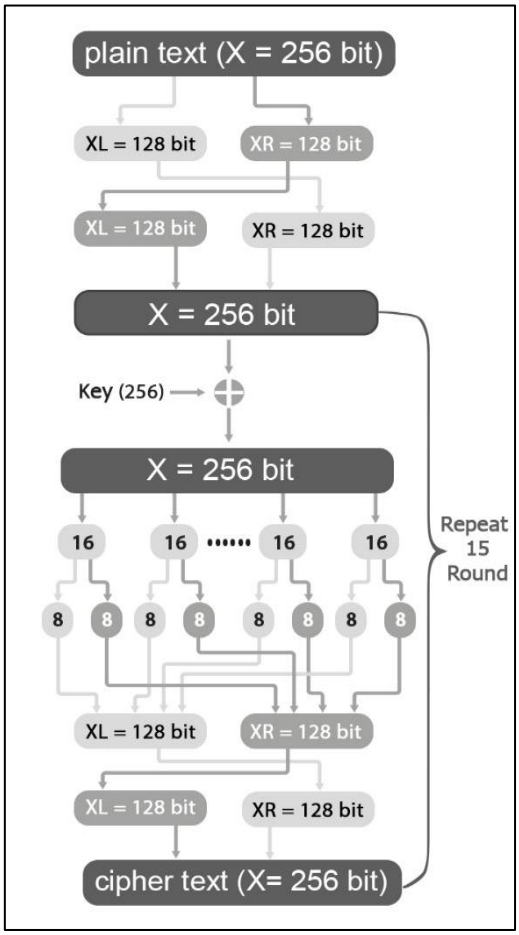

Fig.2. Encryption procedure

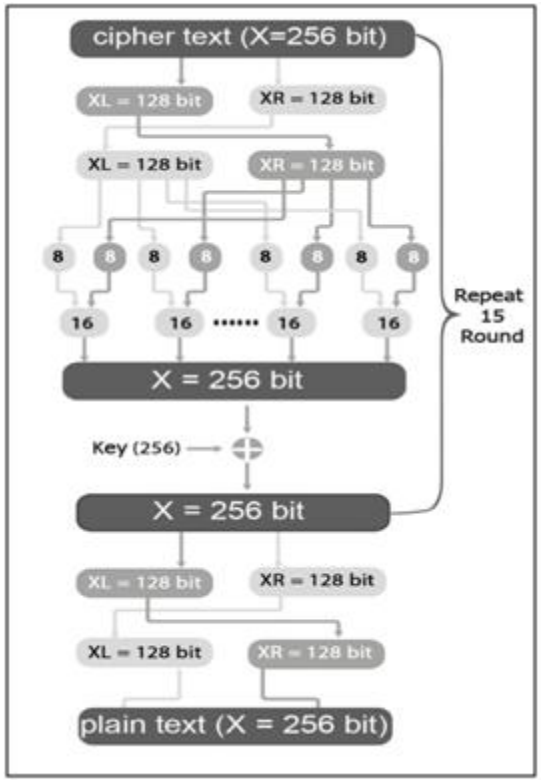

Fig.3. Decryption procedure

There are some issues to be taken in consideration while creating a new model such as:
1. Structure of algorithm: is the main reason to increases the time of execution. So the structure of algorithm should be simple to make algorithm faster.
2. The longer the key is, the higher security it provides in comparison to a shorter one.
3. Any algorithm's overall performance depends on the selection of the mathematical and/or logical operations applied to plain text and key [20].

IV. Experimental Setup

We used VMware Workstation simulation and loaded for a distributed file system through on Hadoop 1x Ubuntu 14.04 using java and Core i5 (8th generation) CPU of Lenovo laptop with 8 GB RAM and 64 bit Windows 10 operation system . Moreover, we have used files created randomly.

V. Performance Analysis

The performance evaluation of proposed algorithm and the results obtained by using MapReduce and without using MapReduce, different analyses are performed to measure the proposed method's performance with comparison to various cryptographic algorithms.

A. Security Analysis

In this section, we compared our proposed method to some related algorithms as shown in table [I].

Table- I: comparison of algorithms

Security algorithms	Cipher type	Key size (bit)	Block size (bit)	Key space	Round
DES	Symmetric block cipher	64	64	2^{64}	16
AES	Symmetric block cipher	128,129 or 256	128,129 or 256	$2^{128}, 2^{192}, 2^{256}$	10,12,14 depend on the size of the key
IDEA	Symmetric block cipher	128	64	2^{128}	8.5
Blowfish	Symmetric block cipher	(32-448)	64	$2^{32} - 2^{488}$	16
The proposed method	Symmetric block cipher	256-3840	256	$2^{256} - 2^{3840}$	15

B. Key space analysis

Key space is a set of potential probabilities to get a key ,viz, the number of potential possibilities that hackers work to get the correct key. Therefore, when using a new technology we must focus on determining the size of the key and also focus on the establishment of cryptographic algorithms which are more sensitive and efficient for any minor change that may affect the key during the encryption process. It should be also resistant to attacks, most notably strong force attack.

From the above, it is clear that we must use large key space to ensure the technique's resistance of used for brute force attack. In order to ensure the response to this attack or to ensure resistance to this attack we need a maximum complexity 2 ^ key size to find the correct key. When the k –size is enormous, it's difficult to get the right key. Looking at the table (1) we can see that the proposed algorithm has the largest key space, which means that there is no chance for the brute force attack to break the proposed algorithm.

C. Key sensitivity

The key sensitivity analysis is applied to verify the sensitivity of the encryption scheme to the change in the initial conditions. This means that any simple change in the key of the encryption process leads to a change in the encrypted text, and this text is completely different from the previous text before making the change.

In the proposed model, we can observe this difference through the mechanism in which the text is encrypted using the key based on the key XOR text. After that, dividing it into several blocks, then dividing each block into two parts (a , b) and then assembling each part of inside (a) block and the same process for (b). After that comes the process of swapping blocks. Finally, this process is repeated 15 times as shown in the diagram.

D. Statistical Analysis

This analysis is used to analyze the confusion properties of an encrypted data. In this paper, we will make statistical analysis depending on correlation analysis and Information Entropy Analysis.

a. Correlation analysis

This analysis is used to calculate the correlation among the encrypted data and plaintext. To observe the relation between plaintext and cipher text after measuring the correlation, one can check the result in an encrypted data if the correlation has zero value, this indicates that the proposed algorithm is good enough. Equation (1) shows the mathematical expression of correlation [21].

$$r\alpha\beta = \frac{cov(\alpha,\beta)}{\sqrt{D(\alpha)}\sqrt{D(\beta)}} \quad (1)$$

Where, $cov(\alpha,\beta) = \frac{1}{N}\sum_{i=1}^{N}(\alpha i - E((\alpha))(\beta i - E(\beta))$

$$D(\alpha) = \frac{1}{N}\sum_{i=1}^{N}\left(\alpha i - E(\alpha)\right)^2$$

$$D(\beta) = \frac{1}{N}\sum_{i=1}^{N}\left(\beta i - E(\beta)\right)^2,$$

Here, (α) and (β) have two values for which the correlation has to be calculated, N refers to the total number of elements obtained from the data, E (α) = mean of (α), and E (β) = mean of (β). As shown in fig. (3) Our proposed algorithm has the smallest value of correlation coefficient which makes it the best result in comparison with AES, DES, and IDEA.

b. Information Entropy Analysis

It is used to measure the amount of randomness of a message (M) mathematically, the entropy H(S) of message m can be described as shown in Equation (2)

$$H(s) = \sum_{i=0}^{n-1} P(S_i) \log_2 \frac{1}{P(S_i)} \quad (2)$$

Where P (Si) is probability symbol Si , log is of base 2. In this analysis the ideal value for result should be H(S) = 8, the result of entropy close to value 8 refers that encrypted output is highly random in nature.

The randomness of message of cipher text data is measured by entropy. The proposed model has a good result that comes close to the ideal value (8) as shown in fig. (4) Which is compared to AES, DES, and IDEA.

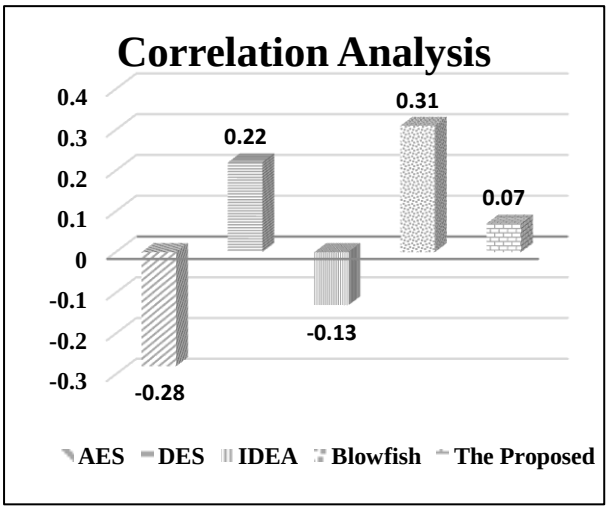

Fig.4. Representation of the correlation analysis

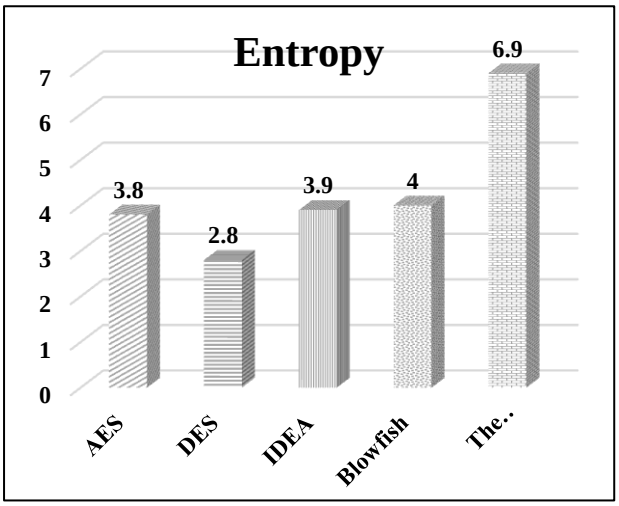

Fig.5. Representation of the entropy analysis

VI. EXECUTION TIME

The time of execution is the time the algorithm needs to encrypt and decrypt the data. Fig. 6 reveals the execution time to implement the encryption of the proposed algorithm for the different sizes of a file such as 1MB, 10MB, 100MB, 200MB, and 250MB. Whereas Fig. 7 shows the time of execution of decryption of the proposed algorithm for the same target files. Fig. 8 shows the different time needed to implement the proposed algorithm for encryption data using MapReduce. However, Fig. 9 shows the different time needed to implement the proposed algorithm for decryption data using MapReduce. Therefore, it can be observed that using MapReduce shows less time consuming. On that ground, we can say that the results may be somewhat convergent when encrypting data is of a small size, but the proposed algorithm proves its worth more when the file size increases.

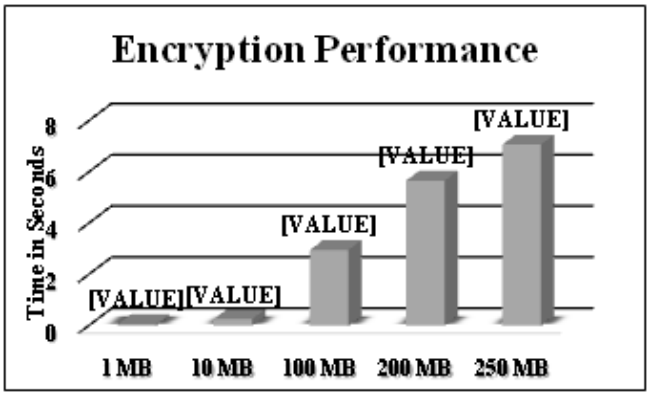

Fig.6. Encryption performance

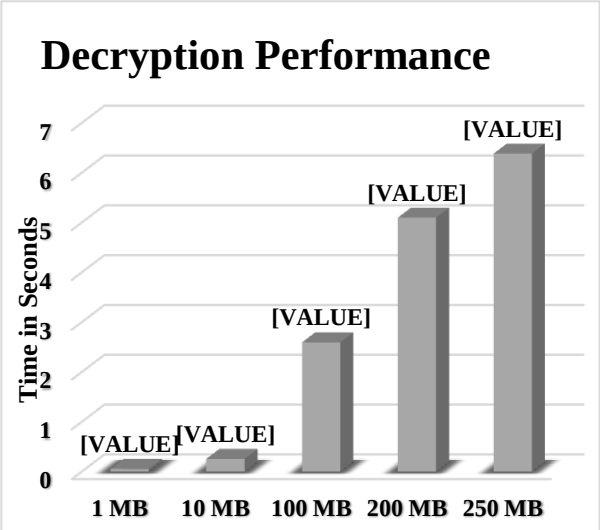

Fig.7. Decryption performance

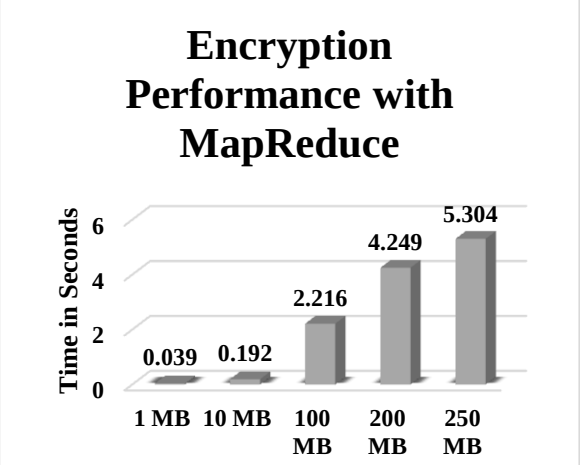

Fig.8. Encryption performance with MapReduce

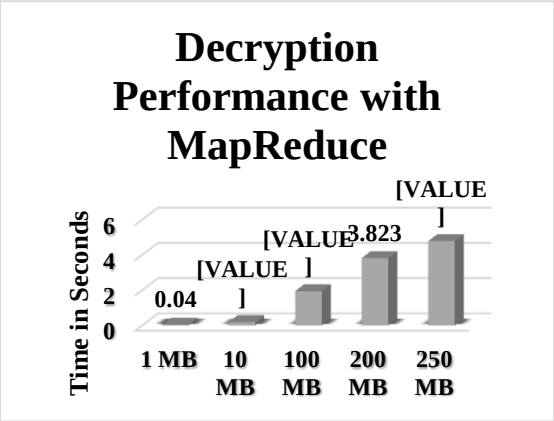

Fig.9. Encryption performance with MapReduce

VII. CONCLUSION

In the present paper, we have used a new stream cipher technique for data encryption and applied the results by MapReduce. We processed data by dividing them into more than one block of 256 bit, then, processed each block by dividing into two parts and made swap then merge. After that, we applied XOR operation with 256 bit key size. Again we divided the block into 16 parts of 16 bits, each part was divided into two arrays after that we combined left part in one array with the right part in one array then made a swap and merge. Finally we repeated this process for 15 times to insure the security from attacks.

In our proposed model, we used large key space which made it strong to guess the plain text, in the same path makes it far away from brute force attack.

In this model, we have tried to consume less time to process a huge data because we used parallel processing with the help of map and reduce function, in addition we used some of the concepts to measure performance analysis to carry out this work properly simultaneously ensure the protection of data.

REFERENCES

1. X. Dong, R. Li, H. He, W. Zhou, Z. Xue, and H. Wu, "Secure sensitive data sharing on a big data platform," Tsinghua Sci. Technol., vol. 20, no. 1, pp. 72–80, 2015, doi: 10.1109/TST.2015.7040516.
2. P. Johri, A. Kumar, S. Das, and S. Arora, "Security framework using Hadoop for big data," in Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2017, 2017, vol. 2017-Janua, pp. 268–272, doi: 10.1109/CCAA.2017.8229813.
3. G. A. AL-Rummana and G. N. Shende, "Homomorphic Encryption for Big Data Security A Survey," Int. J. Comput. Sci. Eng., vol. 6, no. 10, pp. 503–511, 2018, doi: 10.26438/ijcse/v6i10.503511.
4. S. Song, J. Zhang, X. Liao, J. Du, and Q. Wen, "A novel secure communication protocol combining steganography and cryptography," in Procedia Engineering, 2011, vol. 15, pp. 2767–2772, doi: 10.1016/j.proeng.2011.08.521.
5. M. I. S. Reddy and D. A. P. S. Kumar, "Secured Data Transmission Using Wavelet Based Steganography and Cryptography by Using AES Algorithm," Procedia - Procedia Comput. Sci., vol. 85, no. Cms, pp. 62–69, 2016, doi: 10.1016/j.procs.2016.05.177.
6. J. Thakur and N. Kumar, "DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis," Int. J. Emerg. Technol. Adv. Eng., vol. 1, no. 2, pp. 6–12, 2011.
7. P. Patil, P. Narayankar, D. G. Narayan, and S. M. Meena, "A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish," in Procedia Computer Science, 2016, vol. 78, pp. 617–624, doi: 10.1016/j.procs.2016.02.108.
8. T. S. Algaradi and B. Rama, "A Novel Blowfish Based-Algorithm To Improve Encryption Performance In Hadoop Using Mapreduce," Int. J. Sci. Technol. Res., vol. 8, no. 11, pp. 2074–2081, 2019.
9. S. Kansal and M. Mittal, "Performance Evaluation of Various Symmetric Encryption Algorithms," in International Conference on Parallel, Distributed and Grid Computing, IEEE., 2014, pp. 105–109.
10. S. Pavithra and E. Ramadevi, "Performance Evaluation of Symmetric Algorithms," J. Glob. Res. Comput. Sci., vol. 3, no. 8, pp. 43–45, 2012.
11. M. RezaeiJam, L. M. Khanli, M. S. Javan, and M. K. Akbari, "A survey on security of Hadoop," in Proceedings of the 4th International Conference on Computer and Knowledge Engineering, ICCKE 2014, 2014, pp. 716–721, doi: 10.1109/ICCKE.2014.6993455.
12. A. Jayan and B. R. Upadhyay, "RC4 in Hadoop security using MapReduce," in ICCIDS 2017 - International Conference on Computational Intelligence in Data Science, Proceedings, 2018, vol. 2018-Janua, pp. 1–5, doi: 10.1109/ICCIDS.2017.8272637.
13. V. Kadre and S. Chaturvedi, "AES – MR: A Novel Encryption Scheme for securing Data in HDFS Environment using MapReduce," Int. J. Comput. Appl., vol. 129, pp. 12–19, 2015, doi: 10.5120/ijca2015906994.
14. P. D. B. Saraladevia, N. Pazhanirajaa, P. Victer Paula, M.S. Saleem Bashab, "Big Data and Hadoop-A Study in Security Perspective," in Procedia Computer Science, 2015, pp. 596–601.
15. T. Devika and M. Khurana, "Security of Big Data in Hadoop Using AES-MR with Auditing," Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 7, no. 1, pp. 100–105, 2017, doi: 10.23956/ijarcsse/V6I5/0387.
16. J. Zhao et al., "A security framework in G-Hadoop for big data computing across distributed Cloud data centres," in Journal of Computer and System Sciences, 2014, vol. 80, no. 5, pp. 994–1007, doi: 10.1016/j.jcss.2014.02.006.
17. S. M. Tharayil, K.Kalaiselvi, D. P. Rodrigues, and D. A. Kumar, "Enhancing Performance and security for 'Data in Motion ' in BIG DATA," in International Conference on Interdisciplinary Research in Electronics and Instrumentation Engineering 2015 [ICIREIE 2015] 978-81-929866-3-0, 2015, pp. 32–39.
18. T. Z. Shuyu Li and Y. P. Jerry Gao, "A Sticky Policy Framework for Big Data Security," in In 2015 IEEE First International Conference on Big Data Computing Service and Applications, 2015, pp. 130–137.
19. S. Desai, Y. Park, J. Gao, S.-Y. Chang, and C. Song, "Improving Encryption Performance using MapReduce," in In 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, 2015, pp. 1350–1355, doi: 10.1109/HPCC-CSS-ICESS.2015.206.
20. D. Nilesh and M. Nagle, "The new cryptography algorithm with high throughput," in 2014 International Conference on Computer Communication and Informatics: Ushering in Technologies of Tomorrow, Today, ICCCI 2014, 2014, doi: 10.1109/ICCCI.2014.6921739.
21. B. S. Al-Attab, H. S. Fadewar, and M. E. Hodeish, "Lightweight Effective Encryption Algorithm for Securing Data in Cloud Computing," Comput. Commun. Signal Process., vol. 810, pp. 105–121, 2019, doi: https://doi.org/10.1007/978-981-13-1513-8_13.

AUTHORS PROFILE

Galal A. AL-Rummana

Currently, he is a Ph.D candidate in the School of Computational Sciences, at Swami Ramanand Teerth Marathwada University, Nanded. He has received his M.Sc. degree in Computer Networking from the School of Computational Sciences at Swami Ramanand Teerth Marathwada University, in 2017, Nanded, India. He has received his B.E degree in Computer Engineering form the faculty of Computer Science & Engineering, at Hodeidah University, Yemen, 2014. He is currently working on Big Data Security.

Dr. G. N. Shinde is working as Principal, Yeshwant College, Nanded (India). Earlier he was Pro-Vice Chancellor, SRTM University, Nanded, Maharashtra, INDIA. He has received "Ideal State Teacher Award" from Government of Maharashtra, India for 2008-09 and "Best Principal Award" for 2009-2010 from S.R.T.M. University, Nanded, Maharashtra. He has received M. Sc. & Ph.D. degree from Dr. B.A.M. University, Aurangabad. He has awarded Benjonji Jalnawala award for securing highest marks at B.Sc. Seventeen research scholars were awarded Ph.D. degree under his guidance. He has published more than 90 papers in the International Journals and presented more than 50 papers in International Conferences. He was more than the five times Chairperson for International Conference in abroad. In his account one book is published, which is reference book for different courses in different Universities. He is also member of different academic & professional bodies such as IAENG (Hon Kong), ANAS (Jordan). He is in reviewer panel for different Journals such as IEEE (Transactions on Neural Networks), International Journal of Physical Sciences (U.S.A.), Journal of Electromagnetic Waves and Applications (JEMWA, U.S.A.). His abroad Visit includes U.S.A., Thailand, Portugal, Germany, Switzerland, Italy, Vatican City, Monaco, France, Maldives, Sri Lanka, U. K., Scotland, China , New Zealand and Hong Kong, Singapore. His research interest includes Filters, Wireless Sensor Network System, Image processing and Multimedia analysis and retrieval system and Data mining.

Abdulrazzaq H. A. Al-Ahdal Currently, he is a Ph.D candidate in the School of Computational Sciences, at Swami Ramanand Teerth Marathwada University, Nanded. He has received his M.Sc. degree in Computer Networking from the School of Computational Sciences at Swami Ramanand Teerth Marathwada University, in 2016, Nanded, India. He has received his B.E degree in Computer Science from the faculty of Computer Science & Engineering, at Hodeidah University, Yemen, 2004. He was worked as a lecturer until 2014 .He is currently working on Development Lightweight Cryptographic schema for IoT.