Hybrid Encryption Framework For Securing Big Data Storage in Multi-Cloud Environment

M.Nandhakumar[2], Dr.W.R.Salemjeyaseelan[2]
[1]Dept of Computer Science Engineering
[2]Associate Professor, Dept of Computer Science Engineering
[1, 2] Christian College of Engineering and Technology, Dindigul, TamilNadu.

Abstract- *In the present scenario, big data is facing many challenges regarding the data storage, data theft and unauthorized access. Many researchers are concentrated on developing the security mechanism for big data storage. To overcome the above issue, this paper concentrated on developing the encryption algorithm for storing big data in the multi cloud storage. The multi cloud storage environment permits the user to store the data in to different cloud storage services. This paper aims to develop the secure framework which restricts the insider attacks. The proposed framework contains data uploading, slicing, index- ing, encryption, distribution, decryption, retrieval and merging process. The hybrid encryption algorithm was developed to provide the security to the big data before storing it in to the multi cloud. The Simulation analysis is carried with real time cloud storage environments. The proposed algorithm recorded around 2630 KB/S for the encryption process. The results prove the superiority of the proposed algorithm compared to the bench mark algorithms.*

Keywords- Big data · Multi cloud · Security · Data storage · Attackers

I. INTRODUCTION

In the recent years, cloud technologies become one of the most emerging industries in the IT field. As per the recent survey, the cloud market share has been increase to 43% in 2018 [1]. Cloud provides different services like infrastructure, platform and software, but providing the security to the big data in the cloud is the most critical issue. In general, the medical data, military data and government data usually have the sensitive information which is stored in the cloud environment, but the user does not sure about the security provided by the service providers. The cloud contains many resources such as network, operating systems, databases, memory management and these are all vulnerable to some of the attacks. Hence, security and privacy plays vital role in cloud computing [2].

The cloud has many advantages like reliability, flexibility, unlimited storage and feasible collaboration. However having more advantages to the cloud, still they are lacking in providing the security to the stored data. The practice of storing the big data in single cloud is less popular due to the failure of resource availability and also there are some conditions where the malicious inside attackers will theft the data from the single cloud. The information like military or health care data requires high security. To overcome the discussed issues, we made an attempt to secure the big data in multi cloud environment. Many researchers concentrated on inter cloud, multi cloud or cloud-of-cloud for providing the data security [3, 4]. There is no single service provider who claimed complete security to the stored data. To provide security to the sensitive data, the user has to search for the multi cloud where they will provide availability, confidentiality and Integrity

Multi-cloud (MC) environment is the utilization of ser- vices from different cloud service providers. Multi cloud environment provides the single web interface to access the resources from heterogeneous cloud platforms. The MC has the ability to improve the data sharing and this help the data users with great extent. The MC also provides the flexibility of sharing the data by the data owners in the cloud. The major advantage of using the multi cloud for big data storage is to have data sharing and high security [5, 6].

In single cloud environment, the centralized data storage facility is used and it is vulnerable to attacks [7].

This paper focused on the Big data security issues in cloud computing. In general, the users store their data within the cloud, but they need to be more cautious about their sensitive information like health care data, bank details, and military data from the intruders. To overcome this, we propose a secure big data storage mechanism for multi cloud environment. This research work concentrated on develop- ing encryption algorithm to store the secure big data in the multi cloud.

The rest of the paper is organized as follows. Section 2 deals with the recent studies of security algorithms in big data and multi cloud environment. Section 3 deals with the multi cloud framework for big data storage. Section 4 explains about the hybrid encryption algorithm for big data storage. Section 5 explains about the simulation analysis of the proposed and benchmark algorithms. Finally, Sect. 6 concludes the research work.

II. LITERATURE REVIEW

This section explains about the recent studies in the security models of cloud computing. In [8–11], the researchers define the cloud as “A computing environment which provides IT- enable services to in the form of pay-as-you policy to the users”. Salesforce.com is the first organization which provided IT enabled services delivered to the user through the website in 1999 [12]. In 2002, amazon web services (AWS) provided storage and computation services to the users.[13]. In 2005, big data has been launched by O’Reilly Media [14], but it is available around much longer.

Preserving security and privacy in big data is a very big issue. The cloud holds many threats to the big data at the time of data sharing. In [15], the authors developed the architecture for sharing the medical data in the multi- cloud environment. The proposed architecture uses the cryptographic secret sharing and attribute based encryption models for data sharing. Multi cloud environment fragment and encrypt the data and stores in to the different clouds. The limitations of this architecture is high computation is required to compute large records and waiting time is also increased. The ambiguity is more in the output due to the lack of file indexing. The third party involvement in the CP- ABE designing leads to the malicious access to the data.

In [16], to improve the security in big data, the authors used the advanced encryption standard (AES) algorithm for data sharing in cloud. This mechanism provides the flex- ibility to make the decision by the users to choose the cloud service providers. But this algorithm failed to address the data integrity attacks, insider attacks, ad colluding attacks.
In [17], the authors proposed the encryption model to secure the big data in the cloud. This model uses the third party server to store the some part of the encryption key and remaining part will be stored in to the user machine. If the cloud server encryption key and user machine key colludes then the cloud grants permission to access the data [18]. The major drawback of this method is it uses single cloud model. The waiting time for accessing the data will be high if the file size is large. Therefore, data sharing and encryp- tion takes much time to process in cloud [19]. In [20], the authors developed the secured file sharing model for multi- cloud environment. They used the base 64 encryption mechanism along with shamir’s secret key mechanism in their algorithm. The algorithm efficiently restricted the malicious insider attack. But the algorithm did not use the file index- ing mechanism. Therefore it creates some ambiguity in the output files.
Many algorithms are developed by the researchers to secure big data [21, 22], but they are all failed to imple- ment the framework in multi cloud environment. The exist- ing algorithms are not assured the efficient encryption and decryption process. To overcome the above discussed issues, we proposed an efficient frame work for secure big data storage using data slicing and merging in multi-cloud environment.

III. MULTI CLOUD FRAME WORK FOR DATA STORAGE

The framework for storing big data in multi cloud environment is given in Fig. 1. The proposed framework involves the following modules such as data uploading, slicing, indexing, encryption, distribution, decryption, retrieval and merging. Figure 2 shows the retrieval process of data from the multi-cloud.

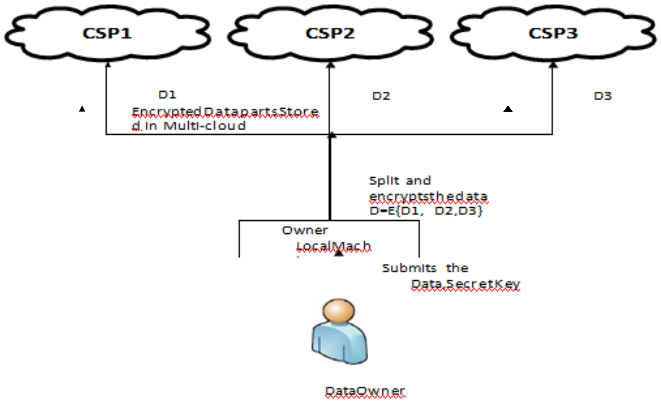


Fig.1 Bigdatastorageatsendersideinmulticloud

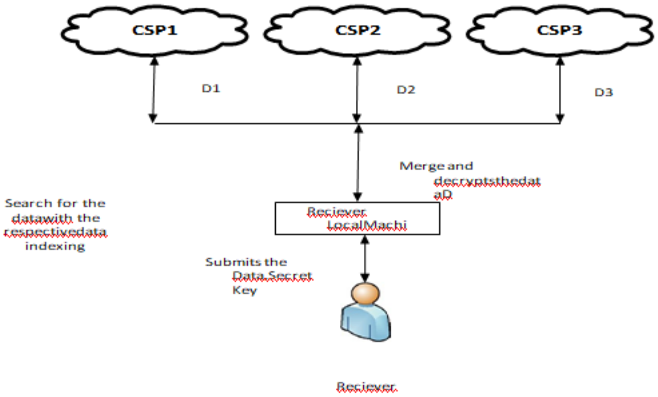


Fig.2 Retrievingthedatafromthemulti-cloud

3.1 Data owner

The user uploads the data to the cloud using the secret key. Upon receiving the request from the receiver, the data owner shares the secret key and file name. The major role of the data owner is to maintain the authorized user's list along with secret keys. In our proposed framework, the third party duties are performed by the data owner.

3.2 Secret key

There is the flexibility to manage the secret keys in cloud. It has been done in three ways. First one is at the service provider side, second one is third party server side and the third one is the data owner side. The proposed framework manages the secret key at the owner side. There is a provision in Amazon S3 storage service to manage the secret key at the data owner side.

This environment consists of different cloud service providers. The data after encryption will be stored in to different storage services present in CSPs.

3.3 Multi-cloud environment

This environment consists of different cloud service provid- ers. The data after encryption will be stored in to different storage services present in CSPs.

IV. SECURE BIG DATA STORAGE IN MULTI CLOUD

The proposed framework assures the data partitioning, indexing and encrypting of data, storage in the multi cloud. There is no possibility to access the data from the cloud without the permission of data owner. The proposed framework receives the data from the user and it splits the data by assigning the indexes to each part. The advanced encryption algorithm has been developed to encrypt the data. The encrypted data parts are going to store in the multi cloud environment. If any user wants to access the data, they need to place the request for the data owner. The data owner will send the secret key along with file name to the requested user through secured chann3.el. The requested user submits the file name to the multi cloud environment and retrieves the encrypted data parts from the multi cloud. The decryption mechanism is applied by the requested user and retrieves the data. The merging operation is performed on the data and stores them in to the local machine of the requested user.

4.1 Encryption process for storing data

The proposed encryption mechanism consists of two mod-ules: Feistel network and AES with Diffie-Hellman. The submitted input data is partitioned in to equal number of sub parts. Each sub part in the encryption algorithm is called a block. Every block is divided in to plain text and key. The Feistel network divides the key in to small number of blocks and applies shift and rotate operations. The cipher key produced by the Feistel network is used in the AES algorithm. The AES Diffie-Hellman algorithm takes the input as the plain text and key from the Feistel network. It performs the 10 rounds for encryption of the plain text by using the substitution and permutation module. Figure 2 and 3 shows the encryption and decryption process of the data. Algorithm 1 shows the encryption procedure for big data in multi cloud.

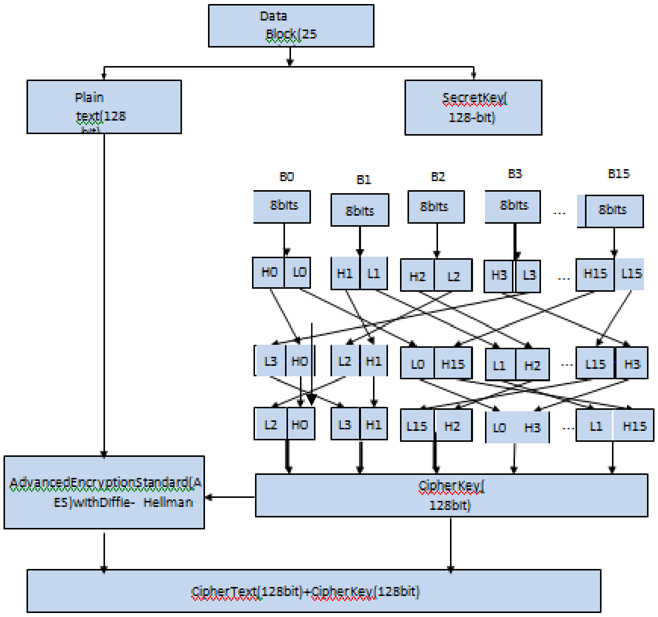
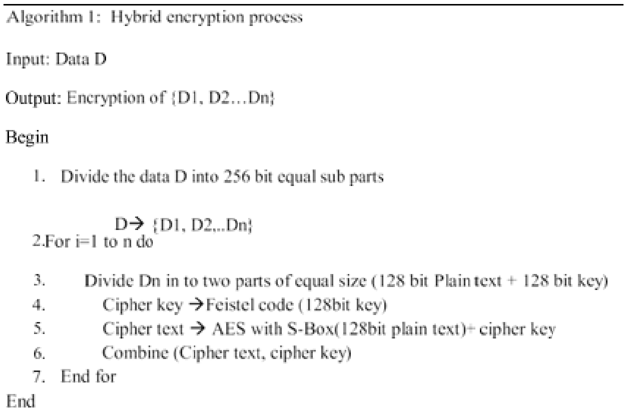


Fig. 3 Encryption mechanism in big data using the proposed framework


```
Algorithm 1:  Hybrid encryption process
Input: Data D
Output: Encryption of {D1, D2…Dn}
Begin
   1.  Divide the data D into 256 bit equal sub parts
              D→ {D1, D2...Dn}
   2.For i=1 to n do
   3.     Divide Dn in to two parts of equal size (128 bit Plain text + 128 bit key)
   4.        Cipher key →Feistel code (128bit key)
   5.        Cipher text → AES with S-Box(128bit plain text)+ cipher key
   6.        Combine (Cipher text, cipher key)
   7.  End for
End
```

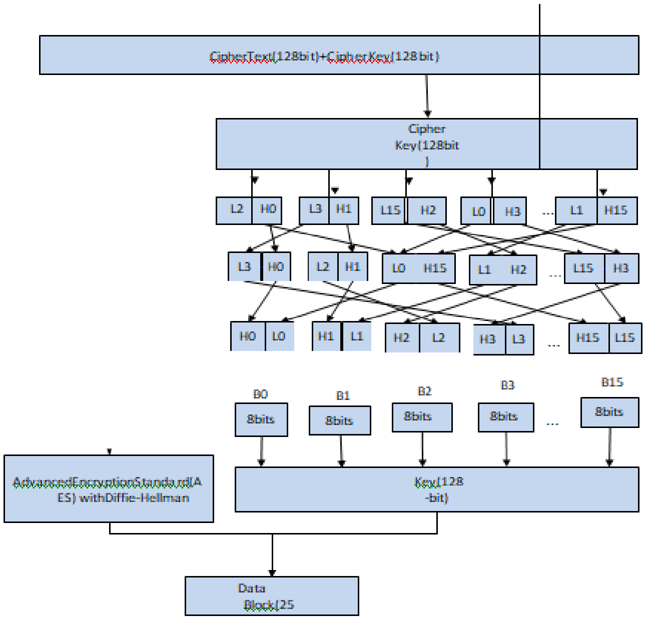


Fig.4 Decryption mechanism for retrieving data

Algorithm 1 reads the data from the data set serially in the form of blocks. Each block is of same size i.e., 256 bits. As an initial step,each block is divided into two equal partitions: 128 bit plain text and 128 bit key which is shown in Fig.3.

FeistelAlgorithm[23]:

a. TheFeistelalgorithmisappliedtothe128bitkeywhichisdividedinto16 bytes.
b. Eachbyteisagaindividedintotwohalves.Oneislowerhalfofthebyteandanotheroneisupperhalfofthebyte.
c. The Feistel algorithm is composed of three modules:logical right shift operation of the upper half, logicalleft shift of the lower half and perform XOR operationforbothupperandlower halfofthebyte.
d. For each byte, the upper part is shifted to the right andlowerpartis shiftedto theleft.
e. The operations like rotating, swapping and adjustingchange the complete order of the bits. Final output willresults the cipher key of 128 bit size. The cipher keysubmittedtotheAESwithDiffie-Hellmanalgorithmforencrypt-ingtheplaintext.

AESwith Diffie-hellmanAlgorithm:

a. The128bitplaintextissubmittedtotheAESalgorithmforencryption.
b. TheAESperformstheencryptionoperationontheplaintextbasedonthe128bitcipherkey.
c. AES algorithm is associated with three modules: initialround,operational roundsand finalround.
d. In the initial round, the add round key() is performed using the XOR technique by combining the round key block.
e. In the operational rounds module, Shift Byte (), ShiftRows(), Mix columns () and Add Round key () process will be performed on each byte.
f. In the final round, except mix columns process remaining process (ShiftByte(), ShiftRows(), AddRoundkey()) is performed.
g. After completion of all the rounds, the plain text is converted into the cipher text.

Finally the cipher text and cipher key combined together and submitted to the multi cloud.

Table1 File sizes for

Data size	Scenarios
1 KB–1 MB	Config1
1 MB–100 MB	Config2
100 MB–500 MB	Config3
500 MB–1GB	Config4

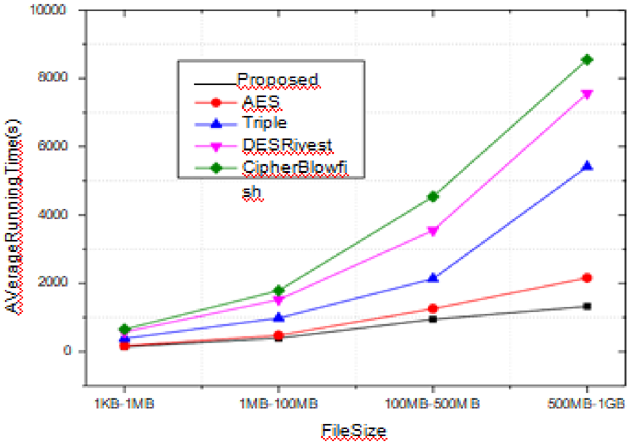


Fig.5 Averageencryptiontimewithdifferentfilesizes

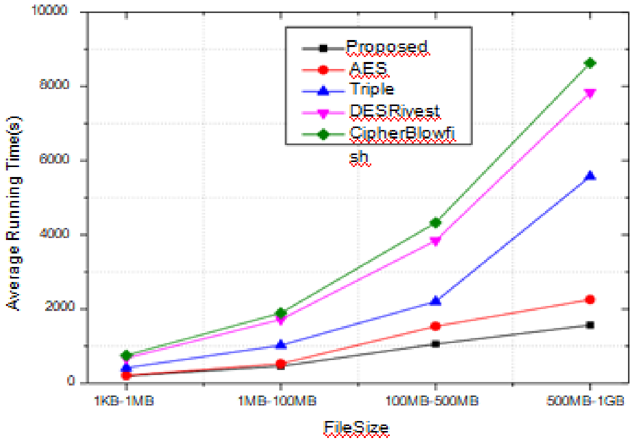


Fig.6 Averagedecryptiontimewithdifferentfilesizes

4.2 Decryption process for retrieving data

The decryption process is same as that of the encryption process, but it is in the reverse process. Figure 4 explains about the decryption process. The cipher data is partitioned in to two parts of 128 bits. The right side block is submit-ted to the Feistel algorithm to retrieve the key and the left side block is submitted to the AES with Diffie-Hellman algorithm to retrieve the plain text. After decryption process, the plan text andkeyisgroupedtogetherandforwardastheoutputdatatotherequesteduser

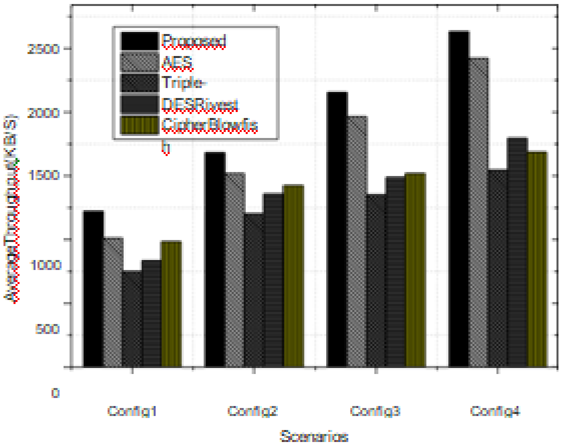


Fig.7 Averagethroughputwithdifferentscenariosoverencryptionprocess

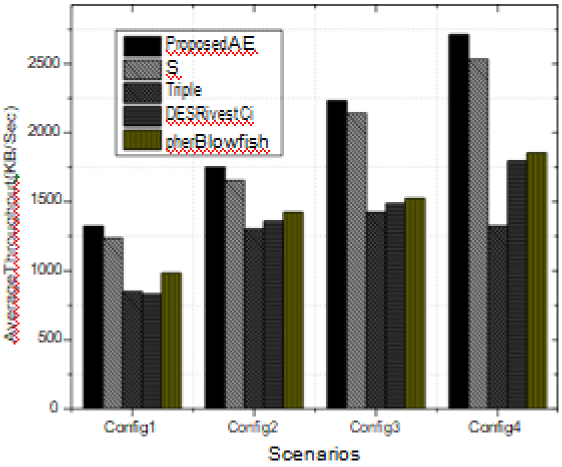
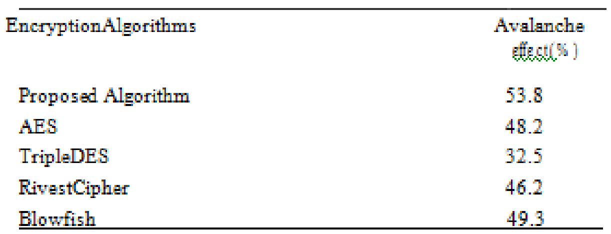
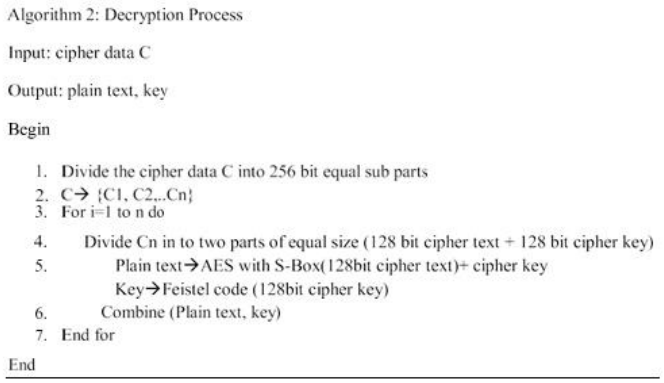


Fig.8 Averagethroughputwithdifferentscenariosoverdecryptionprocess

Table2 Avalancheeffect



EncryptionAlgorithms	Avalanche effect(%)
Proposed Algorithm	53.8
AES	48.2
TripleDES	32.5
RivestCipher	46.2
Blowfish	49.3



```
Algorithm 2: Decryption Process
Input: cipher data C
Output: plain text, key
Begin
    1. Divide the cipher data C into 256 bit equal sub parts
    2. C→ {C1, C2,..Cn}
    3. For i=1 to n do
    4.     Divide Cn in to two parts of equal size (128 bit cipher text + 128 bit cipher key)
    5.         Plain text→AES with S-Box(128bit cipher text)+ cipher key
               Key→Feistel code (128bit cipher key)
    6.       Combine (Plain text, key)
    7. End for
End
```

V. SIMULATION ANALYSIS

The proposed framework uses the multi cloud storages such as Amazon S3, Dropbox and Google Drive for stor- ing the encrypted data. The application used for encryption and decryption of data is designed in JAVA language. The performance of the proposed encryption algorithm is tested with the symmetric encryption algorithms like Triple-DES [24], Blow Fish [25], AES [26] and Rivest Cipher [27]. The parameters considered for the evaluation are throughput and running time of the algorithms. All the simulation experi- ments are conducted using the real time IoT data collected from the health data web site [28]. The proposed algorithm is tested with different file sizes which are given in Table 1. The encryption and decryption process of existing and proposed algorithms are differ from each other. The Triple DES algorithm performs DES algorithm three times with three different keys. From Fig. 5, it is proved that the pro- posed encryption algorithm achieved minimal running time which does not crossed the 1350 s for 500 MB–1 GB of data size. The proposed algorithm is a hybrid algorithm which contains the benefit of AES and Feistel algorithms. AES achieved less running time compared to the Triple DES, Rivest Cipher and Blowfish. In general, the AES is very fast and more secure against all the attacks.

Figure 6 shows the decryption time of the proposed and existing algorithms. It is almost same as the graph in Fig. 5. It is due to the decryption process which is a reverse process of encryption and also all algorithms follow the symmetric key. The proposed algorithm shows less decryption time, i.e., around 1600 s for 500 MB–1 GB. The AES also performed well in decryption which recorded 2260 s for 500 MB–1 GB of file size. The Proposed algorithm achieved less running time compared to the AES, Triple DES, Rivest Cipher and Blowfish algorithms. The Triple DES recorded the 5570 s for completing the decryption process of 500 MB–1 GB of file size.

The average throughput of the encryption and decryp- tion mechanisms is shown in Fig. 7 and 8. The throughput is calculated by the time taken to encrypt the plaintext [29]. Figure 7 shows the superiority of the proposed algorithm against AES, Triple DES, Rivest Cipher and Blowfish algo- rithms. The proposed algorithm recorded around 2630 KB/S for the Config 4 in the encryption process. The proposed algorithm contains the substitution and permutation process and it performs 10 rounds to reach to the output. The AES also given closer counter to the proposed algorithm which recorded 2425 KB/S. The triple DES recorded with lowest rate of 1545 KB/S compared to all the algorithms. It is due to the processing of DES algorithm three times to achieve the output.

Figure 8 shows the average throughput of the algorithms with different scenarios over decryption. The proposed algorithm achieved superior results compared to the existing algorithms. The proposed algorithm recorded 2712 KB/S for decrypting the Config 4. The AES, Triple DES, Rivest Cipher and Blowfish recorded the average throughput as 2532 KB/S, 1324 KB/S, 1796 KB/S and 1853 KB/S.

5.1 Security analysis

This section deals with security analysis of proposed and existing algorithms. The avalanche effect is calculated for finding the strength of the encryption algorithms against the attacks. This research work considered the brute force attacks and hacker's attacks for evaluating the performance of the proposed algorithm. The avalanche effect in encryption algorithms is defined as the number of changed bits in the cipher text to the number of bits in the cipher text. In the symmetric encryption algorithm, if one bit is changed in the plaintext, it leads to the large number of bit changes in the cipher text (Table 2).

The proposed algorithm achieved highest avalanche effect compared to the other benchmark algorithms. It is recorded 53.8% compared to the AES, Triple DES, Rivesr Cipher and Blowfish.

VI. CONCLUSION

This paper proposed the encryption algorithm for securing the big data in multi cloud storage framework. The proposed framework secures the big data from the insider attacks, tampering attacks and DoS attacks. The proposed frame- work uses the data uploading, slicing, indexing, encryption, distribution, decryption, retrieval and merging process to secure the big data stored in the multi cloud. The developed encryption algorithm combines the functionality of AES and Feistel network to achieve high Avalanche effect. The pro- posed algorithm used the medical data set from health data website to evaluate the performance. The results proved that the proposed algorithm achieved performance objectives and high security compared to the bench mark algorithms.

REFERENCES

[1] Canalys,2018,"https://www.canalys.com/newsroom/cloud-market-share-q4-2018-and-full-year-2018",Accessed 11 Jan 2019

[2] Hashem IAT, Yaqoob I, Anuar NB, Mokhtar S, Gani A, Khan SU, 2015,"The rise of "big data" on cloud computing: review and open research issues", Inf Syst 47:98–115

[3] AlZain, MA., Eric P, Ben S, James AT,2012,"Cloud computing security: from single to multi-clouds","In: 2012 45th Hawaii inter- national conference on system sciences", IEEE, pp 5490–5499

[4] Fu Z, Sun X, Liu Q, Zhou L, Shu J,2015, "Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing", IEICE Trans Commun 98(1):190–200

[5] Li R, Xu Z, Kang W, Yow KC, Xu CZ,2014," Efficient multi- keyword ranked query over encrypted data in cloud computing", Future Gener Comput Syst 30:179–190

[6] Xia Z, Wang X, Sun X, Wang Q,2015,"A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data", IEEE Trans Parallel Distrib Syst 27(2):340–352

[7] Somani U, Kanika L, Manish M,2010,"Implementing digital signature with RSA encryption algorithm to enhance the Data Security of cloud in Cloud Computing","In: 2010 First international conference on parallel, distributed and grid computing", IEEE, pp. 211–216

[8] Li P, Li J, Huang Z, Gao C-Z, Chen W-B, Chen K,2018, "Privacy- preserving outsourced classification in cloud computing",Cluster Comput 21(1):277–286

[9] Yang K, Jia X, Ren K, Zhang Bo, Xie R,2013,"DAC-MACS: effective data access control for multiauthority cloud storage systems", IEEE Trans Inf Forensics Secur 8(11):1790–1801

[10] Li M, Shucheng Yu, Zheng Y, Ren K, Lou W,2012, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption", IEEE Trans Parallel Distrib Syst 24(1):131–143

[11] Yang J-J, Li J-Q, Niu Yu ,2015,"A hybrid solution for privacy preserving medical data sharing in the cloud environment.",Future Gener Comput Syst 43:74–86

[12] Akioka S, Muraoka Y,2010,"HPC benchmarks on Amazon EC2", IEEE, pp 1029–1034

[13] O'Reilly Media:" https://www.oreilly.com/ideas/what-is-big-data"

[14] Wang F, Mickens J, Zeldovich N, Vaikuntanathan V,2016,"Sieve: cryptographically enforced access control for user data in untrusted clouds",NSDI 16:611–626

[15] Singh AP,Pasupuleti SK,2016,"Optimized public auditing and data dynamics for data storage security in cloud computing", Proc Comput Sci 93:751–759

[16] Matallah H, Belalem G, Bouamrane K,2017,"Towards a new model of storage and access to data in big data and cloud computing", Ambient Comput Intell 8(4):31–44

[17] Manogaran G, Thota C, Lopez D, Sundarasekar R, 2017,"Big data security intelligence for healthcare industry"," In: Cybersecurity for Industry 4.0. Springer", Cham, pp 103–126

[18] Jakóbik A, Grzonka D, Palmieri F,2017,"Non-deterministic security driven meta scheduler for distributed cloud organizations", Simul Model Pract Theory 76:67–81

[19] Cai H, Boyi Xu, Jiang L, Vasilakos AV,2017," IoT-based big data storage systems in cloud computing: perspectives and challenges" IEEE Internet Things J 4(1):75–87

[20] Althamary IA, Alkharobi TM,2016,"Secure file sharing in multi- cloud using shamir's secret sharing scheme", Trans Netw Commun 4(6):53–67

[21] Jouini M, Rabai LBA,2019,"A security framework for secure cloud computing environment","In: Cloud security: concepts, methodologies, tools, and applications", IGI Global, pp. 249–263

[22] Du M, Wang Q, He M, Weng J,2018,"Privacy-preserving indexing and query processing for secure dynamic cloud storage" IEEE Trans Inf Forensics Secur 13(9):2320–2332

[23] Kuwakado H, Morii M ,2010,"Quantum distinguisher between the 3-round Feistel cipher and the random permutation","In: IEEE international symposium on information theory", IEEE, pp 2682–2685

[24] Mahajan P, Sachdeva A,2013,"A study of encryption algorithms AES, DES and RSA for security", Glob J Comput Sci Technol 13(15):15–22

[25] Devi G, Kumar MP ,2012," Cloud computing: a CRM service based on a separate encryption and decryption using Blowfish Algorithm",Int J Comput Trends Technol 3(4):592–596

[26] Lu CC, Tseng SY,2002,"Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter","In: Proceedings. The IEEE international conference on application-specific systems, architectures and processors", IEEE, pp 277–285

[27] Rivest RL, Robshaw MJ, Sidney R, Yin YL,1998," The RC6 block cipher","In: First advanced encryption standard (AES) conference"

[28] HealthData:"https://healthdata.gov/search/type/dataset.", Accessed on 12 Apr 2019

[29] Elminaam DS, Abdual-Kader HM, Hadhoud MM ,2010,"Evaluating the performance of symmetric encryption algorithms", IJ Netw Secur 10:216–222