International Journal of Future Generation Communication and Networking
Vol. 15, No. 1, (2022), pp. 69–79

A Multi-Stage Cloud Security for Cloud Data using Amalgamate Data Security

[1]E Ravi Kumar, [2] S Sai Satyanarayana Reddy, [3]M Babu Reddy
[1]Research Scholar JNTUK Kakinada, [2] Professor &Principal, [3]Assistant Professor
[1]Vardhaman College of engineering[2] Sreyas Institute of Engineering and Technology, [3]Krishna University College of Engineering & Technology
[1] ravikumar.e@vardhaman.org, [2] saisn90@gmail.com, [3]m_babureddy@yahoo.com

Abstract: Every organization is planning to deploy their projects and data into the cloud storage based on availability of the cloud services. From the past many years security plays the major role in cloud storage for the data availability. Various Encryption and decryption algorithms are that are utilized to provide the better security for the cloud data. Recently another challenge that is identified in cloud storage is uploading large files within the cloud server. To overcome this issue, the multi cloud security is required to store the files in cloud server. Splitting of files in equal chunks and upload the files into multiple clouds, generating the keys for every files and merging the file when the user required for access. Previously various algorithms, techniques are discussed such as dual encryption technique; Compound Encryption based Algorithm (CEBA) for rapid encryption, Compound Secure Storage (CSS) of data with Ensemble Access of Data (EAD) are the integrated methods to provide the security, improving the encryption and decryption time. A few drawbacks are identified in the above techniques such as limited file size and more time for encryption and decryption. In this paper, An Amalgamate Data Security (ADS) is developed to split the large files into the equal shares and stored into the cloud. Generating the private keys for every chunk of the file is also a tedious task. To overcome this, an efficient file merger (EFM) is developed to merge the requested file by the user by matching the private keys. Result shows the performance in terms of encryption and decryption time based on the size of the file.
Keywords: CEBA, CSS, IDS, Cloud**.**

1. Introduction

Cloud computing is most widely used to provide the web services to the various set of users. Nowadays cloud computing providers are increasing the data storage capacity and improving the security and privacy in cloud servers. Huge data can be stored in this cloud storage server. Security is to be provided to the data and access is to provide to the users according to the granted permissions. In this proposed cloud storage, various data can be stored such as files, videos, audio files, document etc. This cloud system involves in storage of data that is received from any clients or users and also from the companies that is made accessible from a cloud of various connected and distributed resources. Authentication is mostly used to prevent the malicious users that are damaging the cloud servers and loss of data can be occur. This paper mainly focuses on managing the data files, privacy and security and prevents the required data from criminals, hackers and attackers. The accessibility of data can be given to the various users that are authorized with the cloud provider.

Copyright ⓒ 2022 SERSC

Figure. 1.Uploading file into Cloud with generating key. (Stage-1)

In this paper, the proposed methodology focused on securing the data into cloud and provides multiple accessible data service to prevent the data from attackers and malicious users to maintain the data confidential. In cloud computing, many issues are facing such as vulnerability because of multiple dynamic factors and huge attack surface. Nowadays many cyber criminals are trying to attack the cloud servers with many types of attacks. Another advantage in this paper is splitting the file into 4 chunks and generating keys for every chunk. Maintaining the multiple keys in the cloud storage is challenging task. If the user wants the access the file, every chunk is stored in the different clouds. Master key is with data owner (DO) and child keys for every chunk is with key manager (KM). The authorised users can access these files. To download these files, the user should have five keys.

The decryption process is done by using the master key and other four keys that are generated by the data owner (DO). The authorized users can access their accounts and requests the DO to get the master key and request the KM to get the child keys. After requesting by the user, mail has been received by the user to open the files.

Copyright ⓒ 2022 SERSC

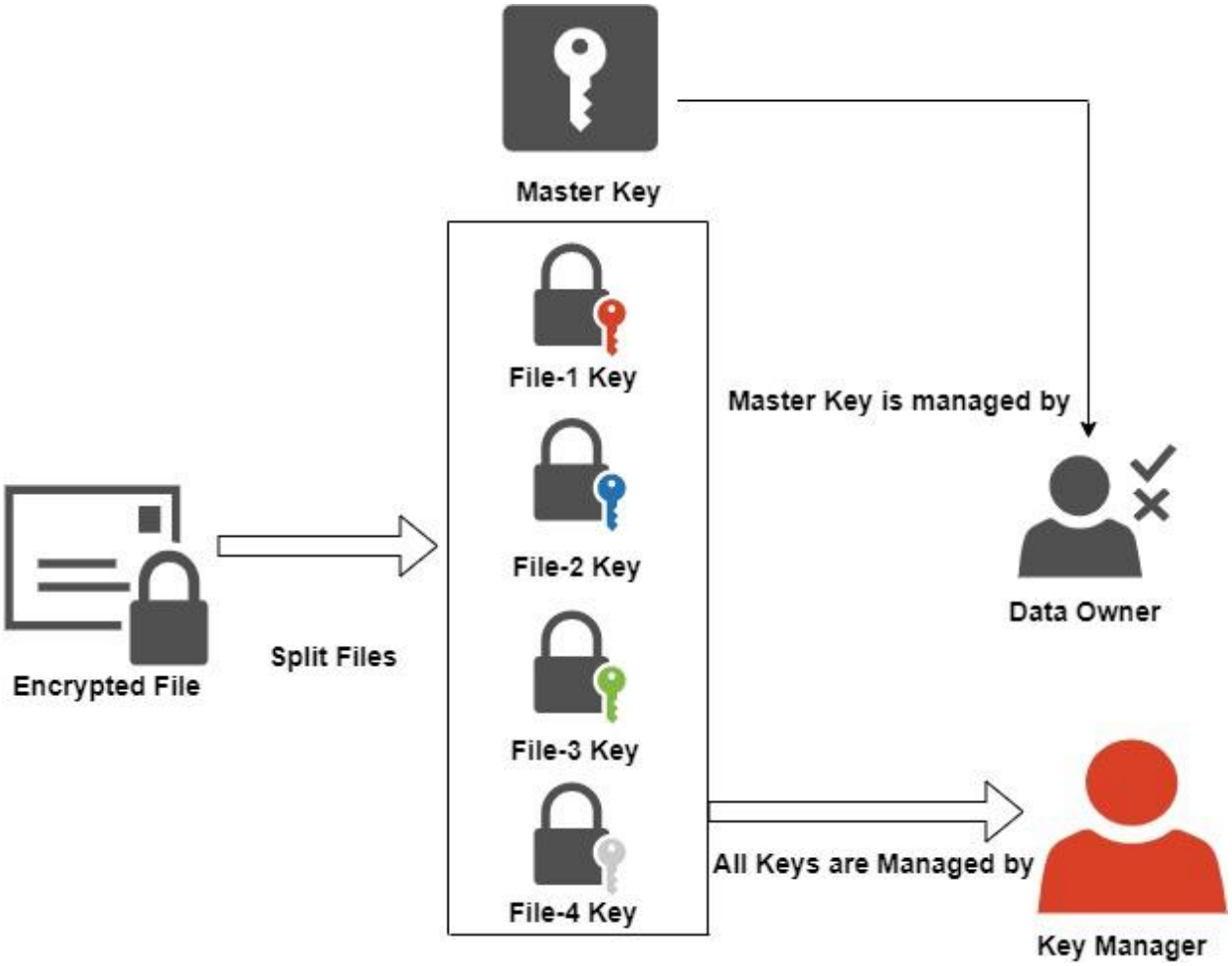

Figure. 2. Encryption, key generation and file chunk sharing's to cloud (stage-2)

2. Literature Review

Ali Gholami et.al features the exploration as per cloud reference engineering organization, physical asset, cloud administration the executives layers and asset control accessible investigating current reconstructions in security protecting touchy information strategies in distributed computing specifically security danger demonstrating and furthermore security improving conventions and arrangements [1]. Yoshita Sharma et al., (2019) [2] developed the multiple encryption method that focus on improving the data security and privacy protection. Many attacks and issues are identified by the authors and introduced this method that applies the efficient encryption algorithm to improve the security for the data. These algorithm can be used in many organizations, personal and communities etc.Shakeeba S et.al, Cloud client is essentially worried about made sure about progression of knowledge. The proposed system focuses on protecting the data by using the enhanced cryptographic algorithms that improves the security in cloud which is indicated by cloud client's data. [3]. Xiao Zhang et.al introduced a cure brooding about frameworks with typical capacity and reduplication stockpiling. The result uncovered that MLFS (Multi-Layer File Sharing System) cases to be sufficient space sparing contribution noticeable and defended I/O record activities [4].

Yang et.al, (2014)proposed the new cloud model to access the data with effective security. In cloud the data is outsourced and also within the untrusted cloud servers, accessing of data can be very

Copyright ⓒ 2022 SERSC

tedious task. Cipher text-Policy Attribute-based Encryption (CP-ABE) is introduced and this most widely used algorithm in all the cloud data storage applications. The proposed algorithm is efficient and has revocable data access control method for multi-authority cloud storage systems. [5]. In distributed computing fine-grained admittance control are often encouraged by absorbing correlation based encryption method. It is a property subordinate encryption using forward/in reverse deduction capacities implementing diverse reach requirements on number credits, additionally level and worldly ascribes [6]. The headway of distributed computing is repressed by the safety and protection issues [7].

Pallavi Kulkarni et al., [8] (2016) addressed various issues in cloud computing data storage and security. In cloud computing security plays the major role. The two main keys that to protect the cloud data are privacy and security. Encryption is most widely used by many applications to protect the data. The author proposed the Identity based and attribute based access policy for encryption technique (IB-ABAP) is introduced which is implemented in cloud. In [9], the new cryptography is introduced and this consists of two security methods such as encryption and decryption. Within encryption, the plain data is converted to cipher text and the converted data cannot be used by the others and this is not understandable code. This can be accessed by the recipient with the secret key. The Blowfish and AES are two algorithms that are used for encryption and decryption and these are considered as cross bread approach in cryptography.

ArunaKumari et al., (2019) [10] mainly focused on gathering the major key security requirements such as Data Integrity (DI) and Data Confidentiality (DC) and availability are noticed after verifying the Cloud Storage(CS). These three issues are most widely discussed by the author. These security requirements are most widely used in many cloud applications.

Rishitha et al., (2018) [11] discussed security risks in the cloud. This is mainly focused on 'Loss of governance'; 'Isolation failure' is utilized to reduce the gaps in security and backup vulnerabilities. By using Amazon Web Services in the cloud platform the Isolation failure and loss of data are reduced. Markandey et al., (2018) [12] discussed the several data security storages which are used in cloud computing. Accessing the cloud data by using several applications provides security in the cloud.D. Zhe et al., (2017) [13] introduced the new data security in the cloud that author developed the policies for accessing of data. These policies prevent the malicious attacks on cloud storage without any data leakage which occurs frequently.

Liang Huang et al., (2020) [14] introduces the new cloud computing techniques that integrates the block chain nodes to the cloud server. This technique also maintains the medical data with proper identity based interface with authentication which overcomes the insufficient abilities of computing with blockchain to verify the security and availability of data.

Leilei Du et a., (2020) [15] proposed the Dynamic Multi-client SSE (DMSSE) that supports the Boolean queries that are used in integrate the users secure information into search tokens and indexes. In this system, data owner gives access to multiple users that are authorized to perform Boolean queries on encrypted database, this is also limits the search to irrelevant keywords.K. Lee et al., (2020) [16] proposed the extended the concept of identity-based encryption (IBE) to support key revocation and ciphertext update functionalities, and proposed a revocable-storage identity-based encryption (RS-IBE) scheme.Wang et al., [17] proposed the lightweight certificate-based public/private auditing scheme associated with asymmetric bilinear pairing for cloud storage. To improve the performance of security, the random oracle model is introduced. The author fully

Copyright ⓒ 2022 SERSC

focused on providing security for data and auditing for cloud storage data.Awad et al., (2019) [18] proposed the new model that provides the encryption algorithms to secure data in cloud. The proposed models such as searchable encryption and hybrid fragmentation are implemented by using Java for simulating the hybrid cloud. The simulation is totally based on private cloud and this improves the data confidentiality in terms of rapid response and improved security. Hidayat et al., (2020) [19] proposed the advanced Encryption Standard (AES) algorithm that prevents the cloud data from attackers. The data is transmitted and stored after applying the encrypt and decrypt. To increase the security features for the data the Markov Chain and Forecasting methods are adopted to improve the security.Kim et al., [20] proposed the secure and effective KNN query processing algorithm to hide the data access patterns by using data privacy and query privacy. For the effective query processing, the secure protocol such as Yao's garbled circuit and a data packing technique is adopted to improve the performance of proposed KNN.

3. Data Encryption and Decryption

This is the process that converts the plain text to cipher text or the other encrypted text which is meaningless. This text is not understood by any user. Decryption is the process the converts the cipher text to plaintext. Symmetric Encryption is used to encrypt the small amount of data. In this process the symmetric key is most widely utilized. Whatever the key used to encrypt the data, the same should be used for the decrypt the data. The aim of the encryption algorithm is tedious to decrypt the data without using the key. It is very difficult to analyze the quality of encryption algorithm. Sometimes this may be easily broken with the proper attack. After the successfully observation the proposed security algorithm is introduced to provide the security for the files that are stored in the cloud.

3.1.An Amalgamate Data Security (ADS) consists of the following algorithms:

Step1: The setup is the step that initializes the Setup (λ, N) $\rightarrow$ (mpk (master public parameters), msk (master secret key)). This step executed by trusted party and security parameters is λ and N is the maximum number of receivers in one encryption. The outputs are represented as mpk and msk.

Step 2: This step extract the private key which is having identity for every user. The input is msk and identity id and output is private key skid.

Step 3: The Encryption is represented as Enc(id and M) $\rightarrow$ C-called as original ciphertext: The Enc run by the data owners (DO) to encrypt the message with DO identity. M represents the input message, an identity id, the original ciphertext C is represented that can be further re-encrypted.

Step 4: In this step, the generation of re-encryption is done by the DO and the inputs are represented as RKeyGen(id, skid, S, k) $\rightarrow$ rk. id-as input identity, skid-private key, the set of DO's are represented with identities S={id1, • • • , idn} and k is maximum revocation number, where id /$\in$ S and $k \leq n \leq N$. The outputs of re-encryption key is represented as rk. This is used to convert an original ciphertext C which is done by specific DO is represented as S (dataowner id).

Step 5: In this step, the Decryption (DEC (skid, C/CT) $\rightarrow$ M/$\perp$) is represented as dec which is done by the Enduser (EU). The EU decrypts the original ciphertext. The private key skid takes as input, an actual/re-encrypted ciphertext C/CT. If the rk is valid which is given by the EU, then the plaintext is represented as M, if any error occurs it is not valid which is represented as $\perp$

Copyright ⓒ 2022 SERSC

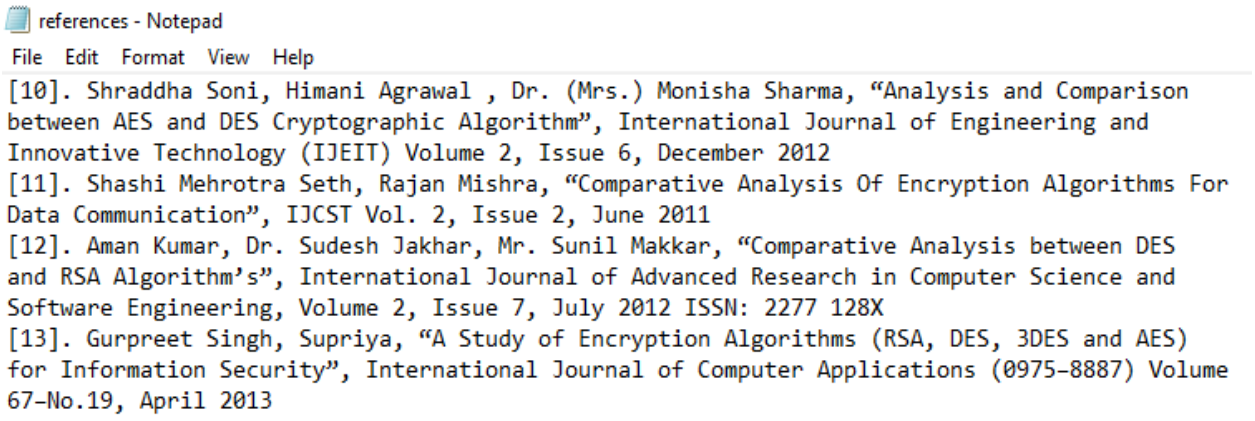

Figure .3. Plain text file to be uploaded to cloud server

3.1.1. Algorithm: 1

By combining thedifferent approaches these algorithms are developed

```
Encryption E;
String File;
File=fileupload.filename;
finalSecureRandom random = new SecureRandom();  //Random Secret Key Generator
   // the largest proposed algorithm key length which is supported by the OS
finalKeyGenerator gen = KeyGenerator.getInstance("E");   //Proposed algorithm
generator.init(KEY_SIZE, random);   //random key is generating with specific key
generator.init(192, random);
returnSecurePreferences.encode(generator.generateKey().getEncoded());
 //Encrypt
      Cipher ciph = Cipher.getInstance("E");
cipher.init(Cipher.ENCRYPT_MODE, secretKey, ivSpec);
cipher.update(File.getBytes());
```

Figure. 4. Encrypted File

3.1.2. Algorithm 2:

File Splitting Algorithm

Input: Encrypted File F.

Output: <File F1, F2, F3, F4> , where F1, F2, F3 and F4 are generated shares.

Step 1: Initialize the size of the file.

Copyright ⓒ 2022 SERSC

Step 2: Threshold value: 4 (this represents the number of shares)
Step 3: Divide the file shares.
Step 4: for (i=filesize; i<4; i++)
Step 5: Split S= F/4;
Step 6: Files Splitted into 4 shares.
Step 7: generate the key for 4 files and master key is generated for F.
Step 8: data stored in cloud.

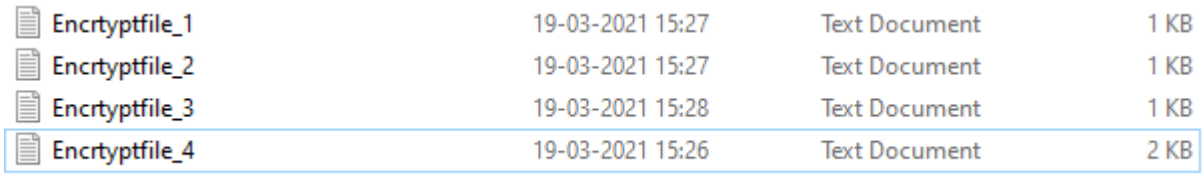

Figure .5.Splitted Encrypted Files

3.1.3. Algorithm 3

Merger Algorithm

Step 1: User request file (R)
Step 2: request to data owner (DO) for master key.
Step 3: request child keys from key manager (KM).
Step 4: permission granted.
Step 5: Private Sub MergeFiles()

```
    With _FileSplitMerge.FileName = txtFileName.Text
        .OutputPath = txtOutputFolder.Text
        .DeleteFilesAfterMerge = chkOption.Checked
backgroundThread = New _
Threading.Thread(AddressOf .MergeFile)backgroundThread.Start()
    End With
End Sub
```

Step 6: now the original file merges and show the data.

3.1.4. Algorithm: 4

Decryption Algorithm

1. Input:EncryptedFile (EncrpFile) (String Format)
2. Output: Decrypted file (.txt file)
3. Method:
4. Step1: Read EncrpFile
5. Step2: For i= 0 to EncrpFile.length
6. *//cipher array*
7. Begin
8. Flag=0;
9. If EncrpFile[i] <0 then
10. Begin
11. EncrpFile[i]=-EncrpFile[i];
12. Flag=1;
13. End

Copyright ⓒ 2022 SERSC

14. Step 3: Decrypt using Algorithm
15. Pos =Darray[i];
16. *//array*
17. If Flag=1 thenDarray[i]=-Pos
18. Step 4: Convert Byte Array intotxtfile
19. Step 5: Produce original txtfile

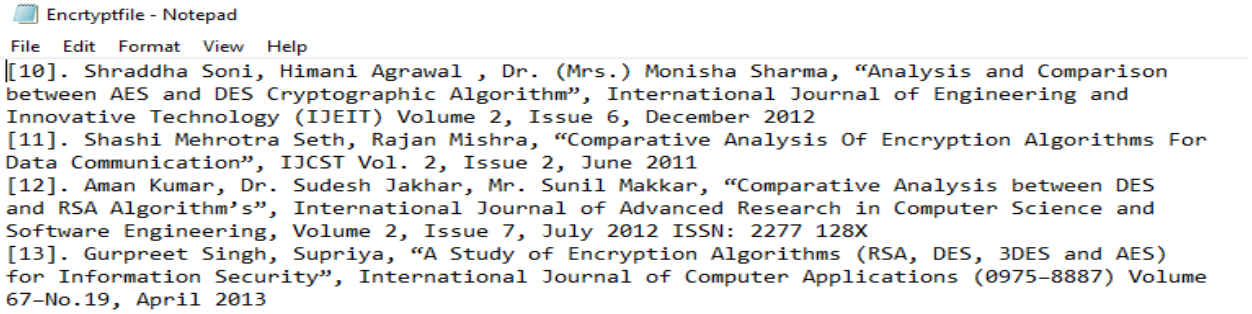

Figure. 6. Merged Decrypted File

4. Results

Java and JDK 1.8 is used to develop the four algorithms and Mysql 5.7 as database. Functionalities are assigned according to the data access control. Data owner (DO), upload the data with encryption of data with private key which is generated by DO. The encrypted file is divided into 4 parts and every part assigned with private key. Four keys are assigned for 4 parts of encrypted files. To decrypt this, the permissions are given by the key manager (KM) and DO.

Below table shows the file size (kb). In table 2 the time is shown for encryption and decryption for RSA, Triple DES, Triple Based Encryption, CSS, CEBA, ADS-EFM algorithm.

Table 1. The performance of proposed model with Encryption time (ms)

File size (kb)	Encryption time (ms)					
	RSA	Triple DES	Triple Based Encryption	CEBA	CSS	ADS-EFM
10	1154	1198	1120	1080	1060	767
20	1205	1187	1211	1160	1145	987
30	1278	1214	1241	1170	1155	997
40	1391	1356	1332	1220	1190	1011
10 kb- 100mb	2909	2798	2509	2219	2101	1465

Copyright ⓒ 2022 SERSC

File Size (kb)	Decryption time (ms)					
	RSA	Triple DES	Triple Based Encryption	CEBA	CSS	ADS-EFM
10	1145	1042	1021	990	976	876
20	1213	1187	1131	1030	1010	998
30	1254	1205	1178	1097	1067	1010
40	1321	1297	1278	1178	1154	1076
10 kb-100mb	1765	1656	1545	1456	1387	1212

Table 2.describes the performance for proposed model

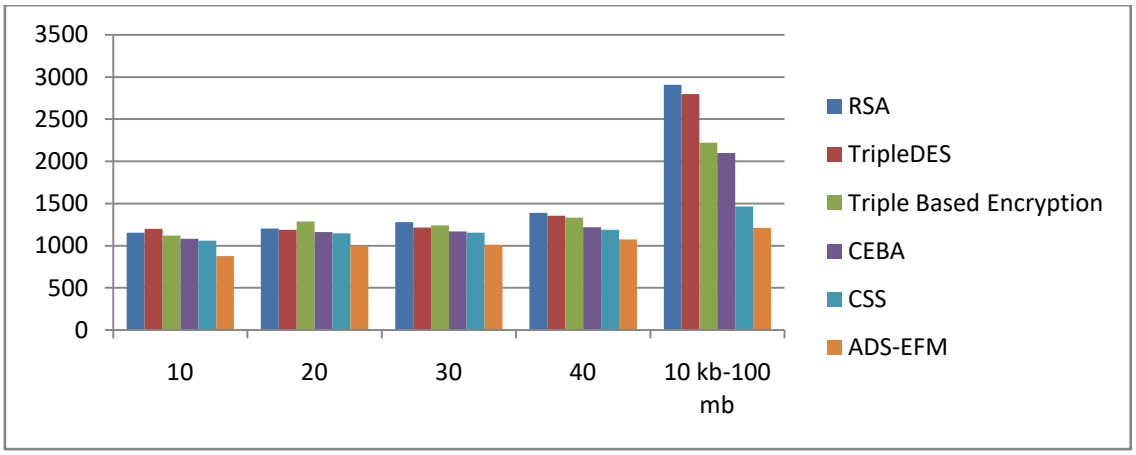

Figure.7. Encryption Time (Sec)

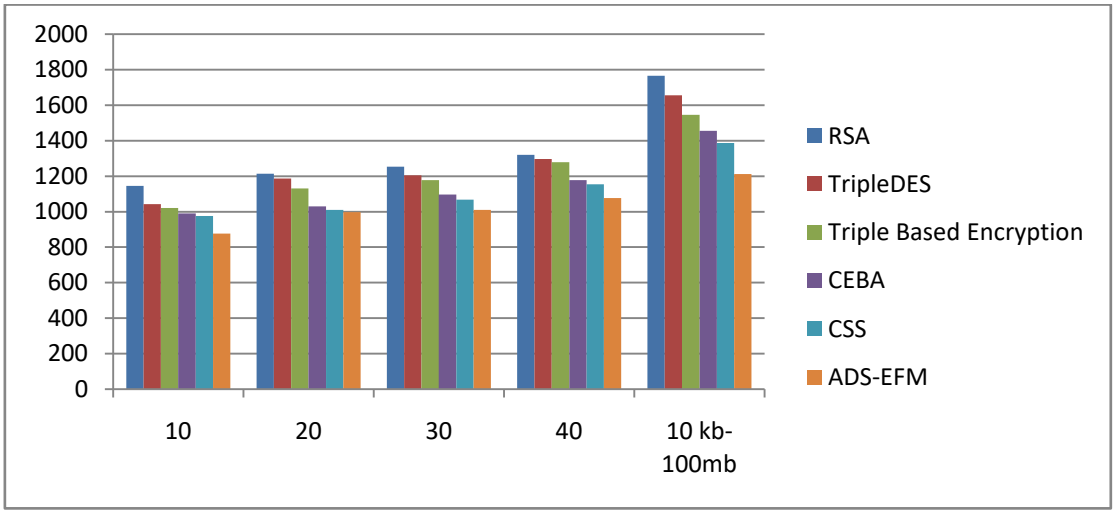

Figure.8.Decryption Time (Sec)

Copyright ⓒ 2022 SERSC

5. Conclusion

In this paper, the security is improved with the splitting of the encrypted file into 4 shares and all these four shares are stored in the four different clouds. Efficient merging of these files and providing the decryption for the merged file. The proposed system focused on providing security for the files that are selected to store in a cloud server. Every technique worked very efficiently to improve cloud storage. Private keys generation is done with a powerful encryption technique. Thus this system can upload all the services to original clouds gets better results in the future.

References

[1] Ali Gholami and Erwin Laure, “Security and Privacy of Sensitive Data in Cloud Computing: A Survey of Recent Developments”, Computer Science & Information Technology (CS & IT), pp. 131 – 150.

[2] Y. Sharma, H. Gupta and S. K. Khatri, "A Security Model for the Enhancement of Data Privacy in Cloud Computing," 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 2019, pp. 898-902.

[3] Shakeeba S. Khan, R. R. Tuteja (2016)“Cloud Security Using Multilevel Encryption Algorithms”, International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Vol. 5, pp. 70 – 75.

[4] Xiao Zhang, Wan Guo, Zhanhuai Li, Xiaonan Zhao, Xiao Qin, (2014)“MLFS:A Multiple Layers Share File System for Cloud Computing”, Globecom Workshop, pp. 99 – 105.

[5] K. Yang and X. Jia, "Expressive, Efficient, and Revocable Data Access Control for Multi-Authority Cloud Storage," in IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 7, pp. 1735-1744, July 2014.

[6] Yan Zhu, Hongxin Hu, Gail-JoonAhn, Mengyang Yu, Hongjia Zhao, Comparison-Based Encryption for Fine-grained Access Control in Clouds, p.p. 105 – 116.

[7] Saravana Kumar N, Rajya Lakshmi G V, Balamurugan B,(2014) “Enhanced Attribute Based Encryption for Cloud Computing”, International Conference on Information and Communication Technologies (ICICT), pp. 689 – 696.

[8] AnkitaNandgaonkar, Pallavi Kulkarni,(2016) “Encryption Algorithm for Cloud Computing”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7, No. 2, pp. 983 – 989.

[9] DiaaSalamaAbdElminaam (2018) “Improving the Security of Cloud Computing by Building New Hybrid Cryptography Algorithms”, IJEIE, Vol.8, No.1, pp. 40 - 48.

[10] A. K. B and S. M, "A Review on Challenges of Security for Secure Data Storage in Cloud," 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2019, pp. 178-184.

[11] Rishitha and T. R. Reshmi, "Security in Cloud Computing," 2018 International Conference on Recent Trends in Advance Computing (ICRTAC), 2018, pp. 14-20, doi: 10.1109/ICRTAC.2018.8679158.

[12] A. Markandey, P. Dhamdhere and Y. Gajmal, "Data Access Security in Cloud Computing: A Review," 2018 International Conference on Computing, Power and Communication Technologies (GUCON), 2018, pp. 633-636, doi: 10.1109/GUCON.2018.8675033.

Copyright ⓒ 2022 SERSC

[13] D. Zhe, W. Qinghong, S. Naizheng and Z. Yuhan, "Study on Data Security Policy Based on Cloud Storage," 2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids), 2017, pp. 145-149, doi: 10.1109/BigDataSecurity.2017.12.
[14] Liang Huang, Hyung-Hyo Lee, "A Medical Data Privacy Protection Scheme Based on Blockchain and Cloud Computing", Wireless Communications and Mobile Computing, vol. 2020, Article ID 8859961, 11 pages, 2020. https://doi.org/10.1155/2020/8859961.
[15] Leilei Du, Kenli Li, Qin Liu, Zhiqiang Wu, Shaobo Zhang, Dynamic multi-client searchable symmetric encryption with support for boolean queries, Information Sciences, Volume 506, 2020, Pages 234-257, https://doi.org/10.1016/j.ins.2019.08.014.
[16] K. Lee, "Comments on “Secure Data Sharing in Cloud Computing Using Revocable-Storage Identity-Based Encryption”," in IEEE Transactions on Cloud Computing, vol. 8, no. 4, pp. 1299-1300, 1 Oct.-Dec. 2020, doi: 10.1109/TCC.2020.2973623.
[17] F. Wang, L. Xu, K. R. Choo, Y. Zhang, H. Wang and J. Li, "Lightweight Certificate-Based Public/Private Auditing Scheme Based on Bilinear Pairing for Cloud Storage," in IEEE Access, vol. 8, pp. 2258-2271, 2020, doi: 10.1109/ACCESS.2019.2960853.
[18] A. S. Awad, A. Yousif and G. Kadoda, "Enhanced Model for Cloud Data Security based on Searchable Encryption and Hybrid Fragmentation," 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), 2019, pp. 1-4, doi: 10.1109/ICCCEEE46830.2019.9070918.
[19] T. Hidayat, D. SianturiTigorFranky and R. Mahardiko, "Forecast Analysis of Research Chance on AES Algorithm to Encrypt during Data Transmission on Cloud Computing," 2020 2nd International Conference on Broadband Communications, Wireless Sensors and Powering (BCWSP), 2020, pp. 163-166, doi: 10.1109/BCWSP50066.2020.9249478.
[20] H. -J. Kim, H. -J. Lee and J. -W. Chang, "A Secure and Efficient Query Processing Algorithm Over Encrypted Database in Cloud Computing," 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), 2021, pp. 219-225, doi: 10.1109/BigComp51126.2021.00049.

Copyright ⓒ 2022 SERSC