2021 International Conference on Emerging Smart Computing and Informatics (ESCI)
AISSMS Institute of Information Technology, Pune, India. Mar 5-7, 2021

Efficient and Secure File Transfer in Cloud Through Double Encryption Using AES and RSA Algorithm

K.Jaspin
Assistant Professor,
St.Joseph's Institute of Technology,
Chennai, Tamil Nadu,
India 600119,
jaspinjose@gmail.com

Shirley Selvan
Associate Professor,
St. Joseph's College of Engineering,
Chennai, Tamil Nadu,
India 600 119,
shirleycharlethenry@gmail.com

Sahana.S
UG Scholar,
St.Joseph's Institute of Technology,
Chennai, Tamil Nadu,
India 600 119,
sahayesyes@gmail.com

Thanmai.G
UG Scholar,
St.Joseph's Institute of Technology,
Chennai, Tamil Nadu,
India 600 119,
guttathanmai98@gmail.com

2021 International Conference on Emerging Smart Computing and Informatics (ESCI) | 978-1-7281-8519-4/20/$31.00 ©2021 IEEE | DOI: 10.1109/ESCI50559.2021.9397005

***Abstract*—With recent advances in Cloud computing, information is being contracted by cloud services. Dropbox and Google Drive provide cloud services to users with low-cost storage. Here we present a protection method by encrypting and decrypting the files which offer an enhanced level of protection. To encrypt the file that we upload in cloud, we make use of Double encryption technique. The file is being encrypted twice using the two algorithms one after the other. The file is first encrypted using AES algorithm and then by RSA algorithm. The corresponding keys are being generated during the execution of the algorithm. This technique increases the security level. The various parameters that we have considered here are security level, speed, data confidentiality, data integrity and cipher text size. Our method is more efficient as it satisfies all the parameters where the conventional methods failed to do so. The cloud we used to store the content of the file is DropBox, which is in the encrypted format using AES and RSA algorithms.**

Keywords—Double Encryption, Security in Cloud storage, Security analysis, AES, RSA.

I. INTRODUCTION

In this fast-moving world data security plays a vital role. Cloud is becoming very much popular with the person users for the purpose of data storage. Cloud platform has been used by the individual's as it offers a lot of free services. Cloud-based services like DropBox provide personal users with cost effective storage, but this can affect fidelity of the service provided. As they give a lot of services at low-cost this situation raises the problem of trustworthiness of Cloud service providers. The necessity to secure data has also been increased drastically. There are many attacks known in the cloud. Some privacy and security breaches are also observed in today's cloud services [1] [2] [3]. The service providers also face many external attacks. In 2018, cosmetic data was looted from a wellness system in Singapore. In such cases, people lose their confidence in providers. Personal data could be used as in the case of Cambridge Analytica data scandal [1]. Therefore, it becomes necessary that end users protect their data from providers by keeping the data onto the personal computer. The other solution will be to use the encryption algorithms. Our base paper suggested the idea of selective encryption along with novel data protection such as fragmentation, encryption and dispersion. Fragmentation methods used a public cloud of the less confidential data fragments [1]. We prefer to present a novel data protection scheme by using a double encryption technique using AES and RSA algorithm for encryption and decryption.

II. RELATED WORKS

Yibin Li et.al, [1] focused on the data over collection problem.They tried to put all customer details into a cloud the security of customer details could be increased They have explored various experiments and the output shows the effectiveness of their approach. Their most direct improvement was reducing the storage in customer smartphone Pictures, videos and other storage information or data occupy more storage space so these are vacated which enable users to install new applications. They showcased an active approach. Whenever an application requires customer data it needs to access request in cloud.

Liwei Kuang et.al [2], implemented a method that could process large scale heterogeneous data that safely decompose a tensor [15][37]. Tensor is used in applications that are rich in data or information [15]. Required number of orthogonal bases is multiplied along with the core tensor. Fully homomorphic encryption is used to encrypt the data, after which decomposition is performed by an algorithm. It could secure data processing on the cloud. A security scheme for cluster management detailed by Jun Wu et.al provides high security [3].

Krikor et.al, in [4] presented a selective encryption method by using high frequency DCT coefficients that contain more visual information. Security is added to the encrypted block by making use of shuffling method. The use of DCT transform helps in data reduction. It is well known that multimedia data are compressed using DCT. At the receiver end, Han Qui et.al [5] estimated the DC coefficients, which help to reduce the transmission error.

Andreas Pommer et.al in [6] designed a scheme to protect content and provide security for a specific multimedia application. This scheme which made use of classical ciphers on the multimedia proved to be inefficient as it required high computation. Med Karim Abdmouleh et.al [7] encrypted the LL band after performing DWT on

the image. This method proved to be fast, robust, and efficient. Keke Gai et.al, in [8] proposed CRN it is widely used in wireless networking. CRN make use of WSGNs. Their proposed approach was examined and the outputs were positive. A method of data storage from end-users to clouds was presented by Han Qiu et.al [9]. Zafar Shahid et.al [10] presented a selective encryption idea that satisfies all real time constraints. In spite of Data integrity being an essential factor, it was not considered in earlier SE methods [11]. Image quality and Integrity are not assured in fractional wavelet based SE methods [12].

In another method, data packets are checked if required to be split during operation period. It provides security and can guard threats from clouds [12]. Han Qiu et al [13] did DCT on bitmap images and tried to reduce rounding errors and recovery from non-selected DCT coefficients. Another encryption algorithm uses a secret key, a map to change positions of image pixels and a second map to modify intensity of image pixels. This method could enhance security to a large level [14].

Han Qiu Et.al in [14] proposed an image protection with shorter calculation resources but with larger image input. The traditional encryption method is very slow. As it is not fast it consumes a lot of CPU calculation resource. For this issue they came up with a combined selective encryption along with the current GPGPU acceleration. Yulen Sadourny et al [15] proposed selective encryption and impact of signaling information. When the signaling was taken into account there was lot of problems, So they tried to resolve by applying the selective encryption scheme. This was implemented because the image code stream provided extra information to the transcoding application

W.Puech et al [16] incorporated AES cipher to encrypt JPEG images. A major advantage of this method is the reduction of calculation resources for big sized data. Ayoub Massoudi et al [17] proposed a cost effective encryption method for JPEG2000 . Harshitha.Y et.al in [18] proposed a study which is based on keyword and multi-keyword. this compares the term efficiency. Here the performance is calculated based on the speed of search done over the encrypted data. They also tried to improve the time for multi keyword search over the RSA.

A more secure algorithm implemented in VHDL used a digital signature [19]. The usage of both cryptography and steganography at the same time improved security to a large extent [20]. Naga Hemanth et.al in [21] proposed an RSA algorithm for the purpose of security of the information and the key which is used for encrypting the information or data. This methodology is implemented in three steps. In the first step text is been encrypted using playfair cipher which make use of 9x6 matrices. The second step deals with XOR operation carried out between key and encrypted text. At the last step of encryption the key was made using the RSA algorithm and further XOR operation was continued. Finally the encrypted information along with key is received and decrypted to read the message. This algorithm provided by them provides extra security among the existing algorithms. A hybrid encryption algorithm that could protect data in Cloud used three encryption keys [22]. An Enhanced RSA algorithm made it difficult for data stealing and consumed less time [23].

III. System Design

We propose a method that provides high security. The user uploads a file into the cloud which has public and private fragments. The private fragment is supposed to securely protect. As said before we have proposed to use the Double Encryption Technique. For Double Encryption the algorithms that we have used are AES and RSA. Here we first encrypt the private fragment containing the important information with AES128. After the first encryption is over the corresponding key is generated. This encrypted file is again subjected to encryption with another algorithm.

Figure 1 shows the process of how a file is being uploaded. Initially the user first registers and then logs into the profile. The user then selects the file which he wanted to upload into the cloud to keep it safe. After choosing the file some internal process is undergone by the file before it gets uploaded. First the file is being encrypted using the AES algorithm and then by RSA algorithm. Double Encryption is done for security purposes.

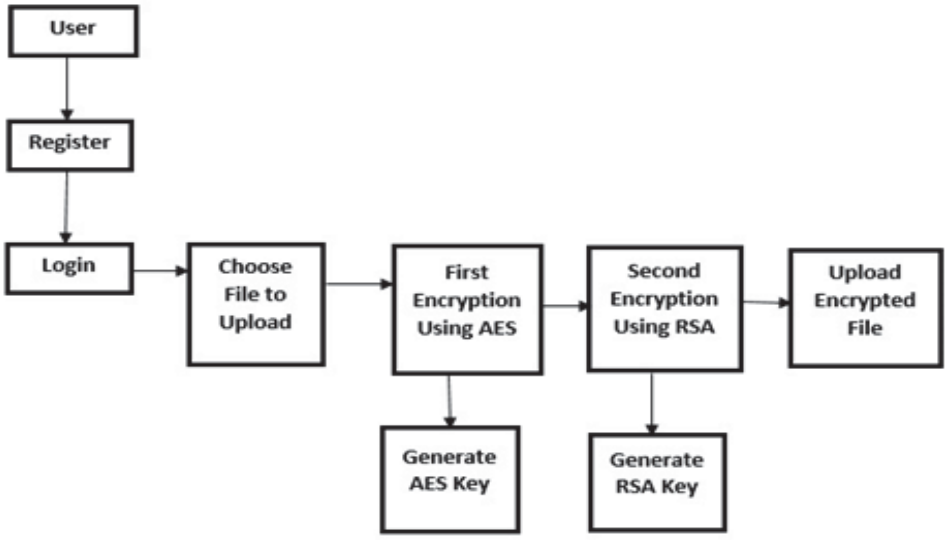

Fig. 1. File Upload Process

Figure 2 shows the system architecture for file download purpose. The user again logs into account. The user views the cloud to check out the files that are being uploaded by others. The user requests the file that he wishes for. This file request is sent to the owner of the file. If the owner of the file wishes to grant access he accepts the request otherwise he deletes it. If the request is accepted, send the key to the user through Email to open the file. The requested user shall make the user of the key to download the file to view or read it. The downloaded file gets stored in the requested user's system

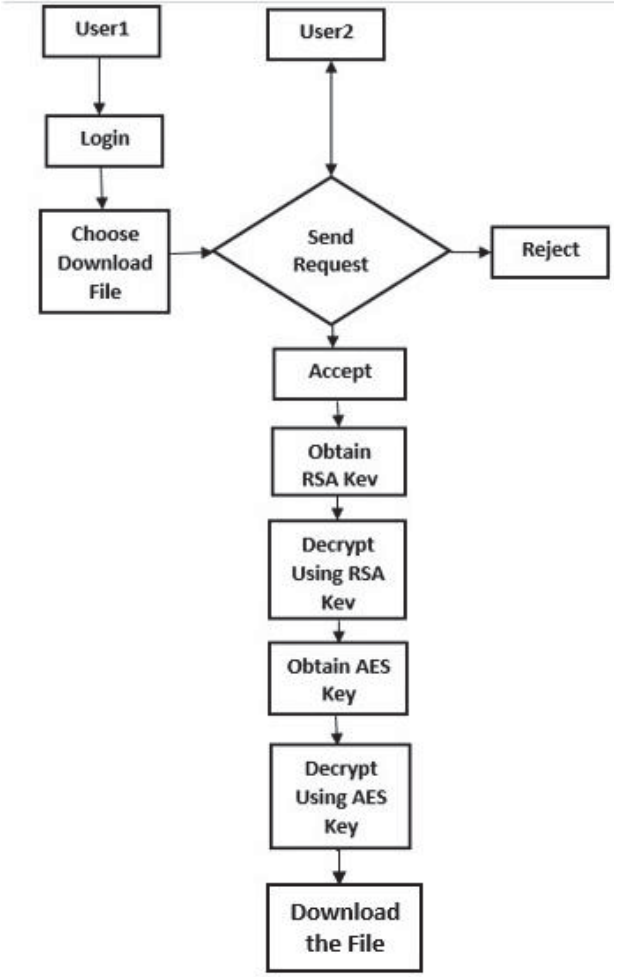

Fig. 2. File Download Process

IV. IMPLEMENTATION

The system has been implemented using AES and RSA algorithms. Both the algorithms are explained here.

A. *Working of AES Algorithm*

1. Obtain the key from cipher key.
2. Assign the plain text to state array.
3. Prefix state array with initial round key.
4. Perform manipulation nine times.
5. Carry out the tenth and last manipulation.
6. Copy cipher text.

Figure 3 represents the working of AES algorithm. AES is an iterative cipher. It is symmetrical block cipher algorithm. It is capable of encrypting 128 bits of plain text. The various keys used by this algorithm are 128,192,256 bits.It is considered as the most secured algorithm.

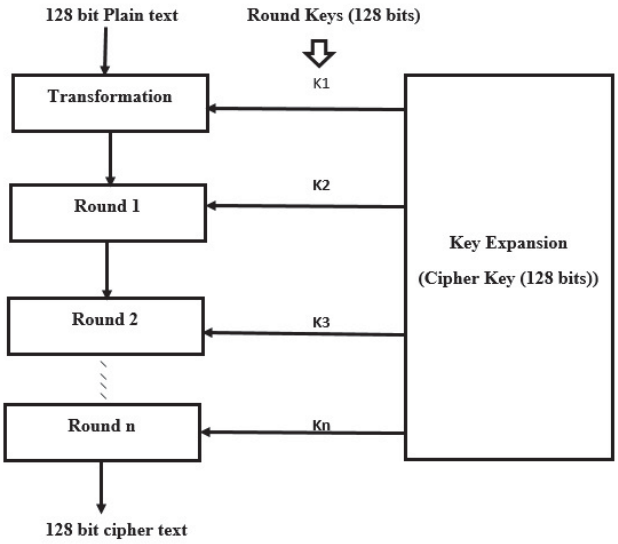

Fig. 3. Working of AES Algorithm

B. *Working of RSA Algorithm [24]*

Step 1: Generating Public Key:

- Select two prime numbers. Suppose p=53 and q=59
- Now we have to compute the public key which is done as follows : we require n and e
- n is computed as n = p*q (3127)
- e is an integer but not a factor of n. e should be like $1< e < \Phi(n)$. So the value of e is taken as 3.

Now our public key is created using n and e.

Step 2: Generating Private Key: [24]

- Here we need to calculate $\Phi(n)$ in such a way that $\Phi(n) = (p -1) (q-1)$. Here, $\Phi(n)=3016$.
- Now we calculate private key d as $d = (k*\Phi(n) + 1) /$ **e** for some integer k

If we take k as 2 then d is 2011. Now we are ready with our

- Public Key (n = 3127 and e = 3) and
- Private Key (d = 2011)

Step 3: Encryption and Decryption [24] [25]

Now we can encrypt and decrypt using an example. Let the example be "HI"

- Convert the letters to numbers: H=8 and I=9
- The encryption formula is $c = 89^e \bmod n$ (1394 for the example)
- The decryption formula is $m = c^d \bmod n$ (the encrypted data comes out as 89 which is nothing but "HI").

V. EXPERIMENTAL RESULT

Table I refers to the comparison of proposed work with existing tables. In our project we consider parameters such

as Security level, Speed, Data confidentiality, Data integrity and Ciphertext size. These are all considered as important parameters to compare various algorithms. This analysis was done by reading various reference papers. This performance efficiency helped us to get to know that the combination of AES and RSA provides more security for file protection than the convolution methods.

The following performance was observed by considering the various parameters such as security, speed, Data confidentiality, Data Integrity etc. The various results are discussed below.

TABLE I. COMPARISION OF PROPOSED WORK WITH EXISTING METHODS

Parameters	DES[4]	BLOWFISH [23]	RC5 [18]	3-DES [28]	AES+ RSA (PROPOSED)
Security	Not secure	Secure	Partially secure	Better than DES	Very secure
Speed	Very slow	Fast	Slow	Slow	Very Fast
Data Confidentiality	No	Yes	No	No	Yes
Data Integrity	No	Yes	No	No	Yes
Cipher Text	Larger than plain text	Same as plain text	Larger than plain text	Larger than plain text	Same as plain text

The execution time for encryption and decryption is tabulated in table II. It is our experimental result where we have taken various file sizes varying in MB and calculated the encryption and decryption time.

TABLE II. EXECUTION TIME FOR ENCRYPTION AND DECRYPTION

File Size	Execution time for Encryption	Execution time for Decryption
1	0.75sec	1sec
15	1.25sec	1.5sec
25	1.5sec	2sec

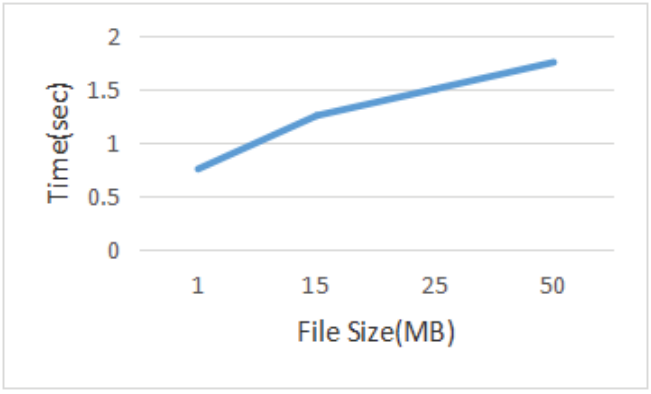

Fig. 4. File Size Vs time (Uploading)

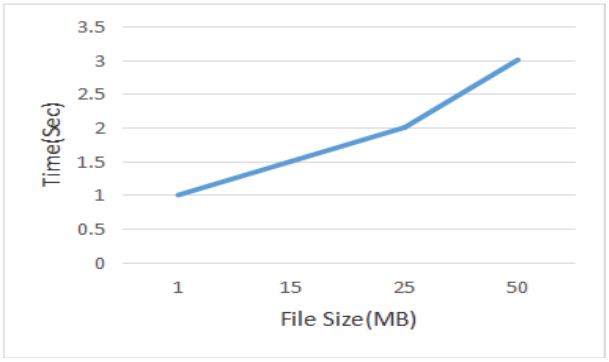

Fig. 5. File Size Vs time (Downloading)

Figure 4 is the graph which shows the time taken to upload the file .The graph shows the time taken by various files of sizes in MB to upload. The time is calculated by first taking the start time from the system before uploading. The time taken for uploading the file followed by encryption and decryption is the end time. The difference between both the times is considered as the upload time. Figure 5 is the graph which shows the time taken to download the file.

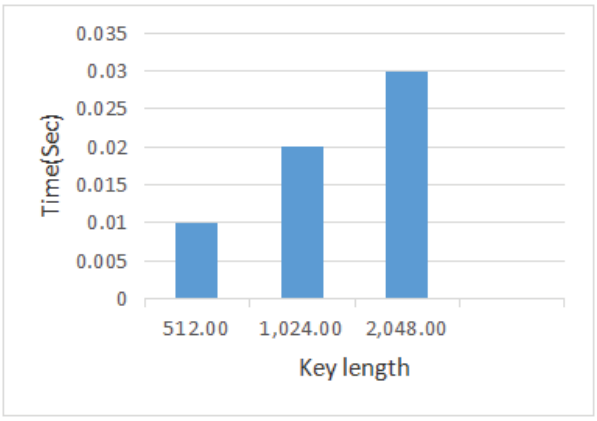

Fig. 6. RSA public key generation graph

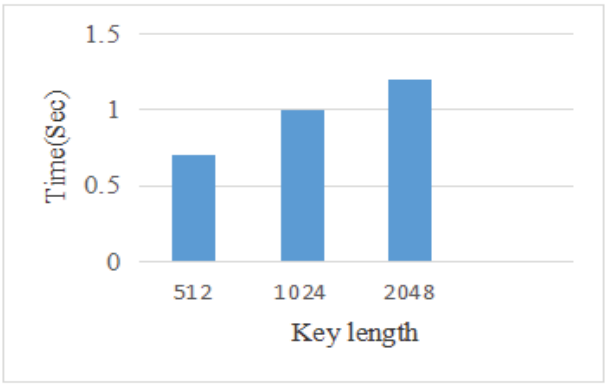

Fig. 7. RSA private key generation graph

The time taken to generate the public key is depicted in Figure 6. Different key length takes different time to generate the public key. Here we have calculated the time taken by 512, 1024 and 2048 key lengths. 512 key lengths take the least time and 2048 key length takes the largest. Figure 7 shows the time taken by the RSA algorithm to generate the private key. Different key length takes different time to generate the private key. Here we have calculated the time taken by 512, 1024 and 2048 key lengths. 512 key

lengths take the least time and 2048 key length takes the largest.

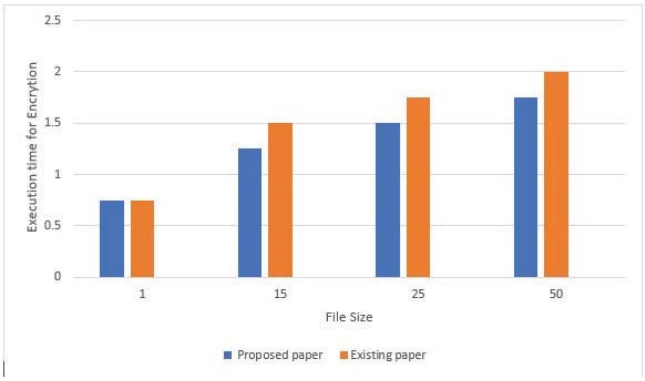

Fig. 8. Time taken for file Encryption

Figure 8 and Figure 9 show the comparison of the results for encrypting and decrypting the various types of files. This shows that the results achieved in our work is much better than previous results.

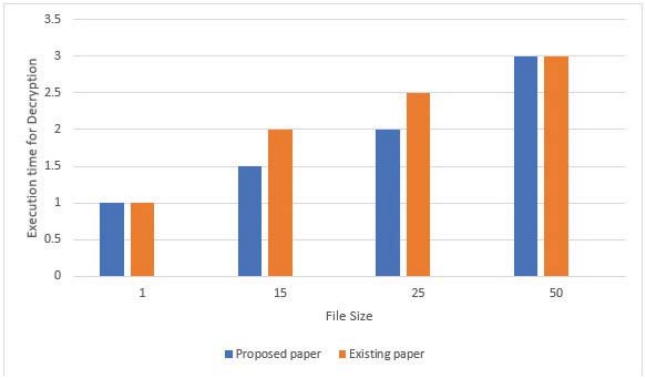

Fig. 9. Time taken for file Decryption

The encryption runtime of text file and decryption runtime of text file are tabulated in TABLE III and Table IV

TABLE III. DATA TABLE FOR ENCRYPTION RUNTIME OF TEXT FILE

File(MB)	DES (in sec)	Blowfish (in sec)	RC5 (in sec)	3-DES (in sec)	AES+RSA (in sec)
0.1	2.5	1.2	1.5	2	1
0.5	3	1.6	1.8	2.5	1.5
0.75	4.5	4	4.2	4.5	3.5
1	5.5	4.5	4.8	5	4
Average time	15.5	11.3	13.8	14	10
Throughput(MB/sec)	1	1.8	1.6	1.25	2

TABLE IV. DATA TABLE FOR DECRYTION RUNTIME OF TEXT FILE

File(MB)	DES (in sec)	Blowfish (in sec)	RC5 (in sec)	4-DES 5-(in sec)	AES+RSA (in sec)
0.1	2.0	1.2	1.5	1.8	1
0.5	2.5	1.8	2	2.3	1.5
0.75	3	2.3	2.5	2.7	2
1.0	4	3.5	3.5	3.8	3
Average time	11.5	8.8	9.5	10.6	7.5

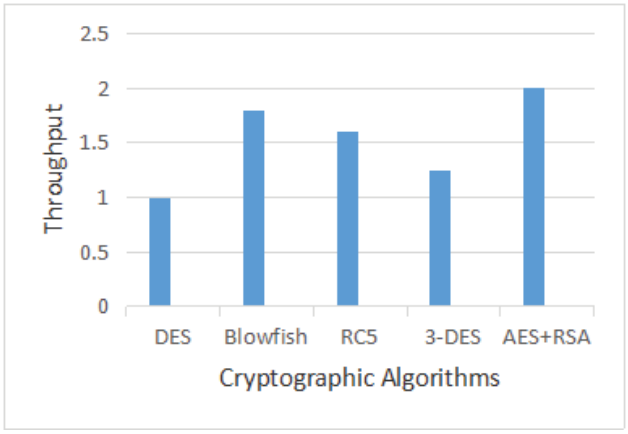

Fig. 10. Graph for Encryption runtime of text files

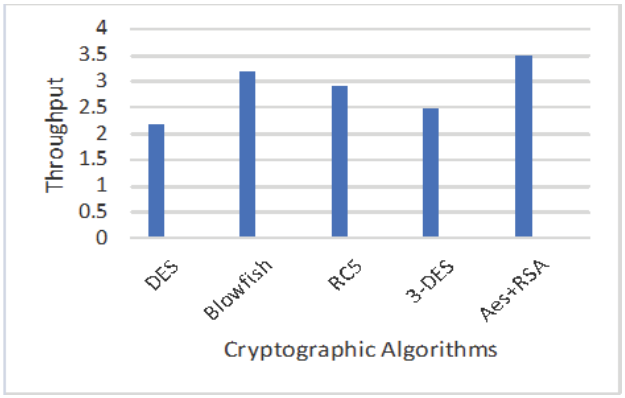

Fig. 11. Graph for Decryption runtime of text files

VI. CONCLUSIONS AND FUTURE WORK

Here, we propose a method to provide high data security while using Cloud storage services. We make use of the Double Encryption Technique to increase the security of the file. From the results obtained, our method provides high security with resistance against propagation errors. The runtime of our algorithm is less compared to the existing algorithms, hence it is fast. Therefore, we propose a secure and cost effective data protection method for cloud service end-users. Our system efficiency in terms of runtime with secure protection of text data over cloud compared with existing encryption and decryption methodologies like DES, Blowfish, RC5,3-DES. Our proposed methodology produces the best result compared with existing methods. In the future machine learning and deep learning may be used in efficient and secure file transfer in the cloud. The encryptions made using machine learning are most welcoming as they are the future technology. As technology advances so does our ability, now a day's neural networks are well capable of learning to keep the data safe.

REFERENCES

[1] Li, Yibin, et al. "Privacy protection for preventing data over-collection in smart city." IEEE Transactions on Computers 65.5 (2015): 1339-1350.

[2] Kuang, Liwei, et al. "Secure tensor decomposition using fully homomorphic encryption scheme." IEEE Transactions on Cloud Computing 6.3 (2015): 868-878.

[3] Wu, Jun, et al. "Big data analysis-based secure cluster management for optimized control plane in software-defined networks." IEEE Transactions on Network and Service Management 15.1 (2018): 27-38.

[4] Krikor, Lala, et al. "Image encryption using DCT and stream cipher." European Journal of Scientific Research 32.1 (2009): 47-57.

[5] H.Qiu,G.Memmi,X.Chen,andJ.Xiong,"DCcoefficientrecovery for JPEG images in ubiquitous communication systems," Future Generation Computer Systems,2019.

[6] Pommer, Andreas, and Andreas Uhl. "Selective encryption of wavelet-packet encoded image data: efficiency and security." Multimedia Systems 9.3 (2003): 279-287.

[7] Abdmouleh, Med Karim, Ali Khalfallah, and Med Salim Bouhlel. "A novel selective encryption DWT-based algorithm for medical images." 2017 14th International Conference on Computer Graphics, Imaging and Visualization. IEEE, 2017.

[8] Gai, Keke, et al. "Spoofing-jamming attack strategy using optimal power distributions in wireless smart grid networks." IEEE Transactions on Smart Grid 8.5 (2017): 2431-2439.

[9] Qiu, Han, and Gerard Memmi. "Fast selective encryption methods for bitmap images." International Journal of Multimedia Data Engineering and Management (IJMDEM) 6.3 (2015): 51-69.

[10] Shahid, Zafar, and William Puech. "Visual protection of HEVC video by selective encryption of CABAC binstrings." ieee transactions on multimedia 16.1 (2013): 24-36.

[11] Xiang, Tao, Chenyun Yu, and Fei Chen. "Secure MQ coder: An efficient way to protect JPEG 2000 images in wireless multimedia sensor networks." Signal Processing: Image Communication 29.9 (2014): 1015-1027.

[12] Li, Yibin, et al. "Intelligent cryptography approach for secure distributed big data storage in cloud computing." Information Sciences 387 (2017): 103-115.

[13] Qiu, Han, Nathalie Enfrin, and Gerard Memmi. "A case study for practical issues of DCT based bitmap selective encryption methods." 2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC). IEEE, 2018.

[14] Qiu, Han, and Gerard Memmi. "Fast selective encryption method for bitmaps based on GPU acceleration." 2014 IEEE International Symposium on Multimedia. IEEE, 2014.

[15] Sadourny, Yulen, and Vania Conan. "A proposal for supporting selective encryption in JPSEC." IEEE Transactions on Consumer Electronics 49.4 (2003): 846-849.

[16] Puech, William, and José M. Rodrigues. "Crypto-compression of medical images by selective encryption of DCT." 2005 13th European signal processing conference. IEEE, 2005.

[17] Massoudi, Ayoub, et al. "Secure and low cost selective encryption for JPEG2000." 2008 Tenth IEEE International Symposium on Multimedia. IEEE, 2008.

[18] Harshitha, Y., S. Seema, and P. Apoorva. "Comparative study on RSA algorithm of multi-keyword search scheme over encrypted cloud data." 2017 International Conference on Intelligent Computing and Control (I2C2). IEEE, 2017.

[19] Viney Pal Bansal and Sandeep Singh "A Hybrid Data Encryption Technique using RSA and Blowfish for Cloud Computing on FPGAs", in Proceedings of 2015 RAECS UIET Panjab University Chandigarh ,2015

[20] Shubhi Mittal, Shivika Arora and Rachna Jain "PData Security using RSA Encryption Combined with Image Steganography" ,2016.

[21] Naga Hemanth P, Abhinay Raj N, Nishi Yadav "Secure Message Transfer using RSA algorithm and Improved Playfair cipher in Cloud Computing" in 2nd International Conference for Convergence in Technology,2017.

[22] Mahalle, Vishwanath S., and Aniket K. Shahade. "Enhancing the data security in Cloud by implementing hybrid (Rsa & Aes) encryption algorithm." 2014 International Conference on Power, Automation and Communication (INPAC). IEEE, 2014.

[23] Dr.D.I.GeorgeAmalarethinam,H.Leena "Enhanced RSA Algorithm with varying Key Sizes for Data Security in Cloud" in World Congress on Computing and Communication Technologies,2017.

[24] https://www.geeksforgeeks.org/rsa-algorithm-cryptography

[25] https://www.google.com/search?source=univ&tbm=isch&q=Liverpool+Community+College+%2B+Encryption+and+Decryption%