A Hybrid Cryptographic Technique for File Storage Mechanism Over Cloud

Shivam Sharma, Kanav Singla, Geetanjali Rathee and Hemraj Saini

Abstract The cloud is a radical platform that conveys dynamic, virtualization asset pools and high convenience. Since distributed computing lays on web, security issues like information security, protection, secrecy, and verification are experienced. In order to get rid of these, an assortment of mechanisms and encryption algorithms are utilized in various blends. On the comparable terms, we made utilization of hybrid encryption with the utilization of crossbreed cryptographic calculations to upgrade the security of information or data file on cloud. We plan at scrutinizing different incorporation of encryption algorithms, in view of various execution constraints to reason a hybrid calculation which can anchor information more effectively on cloud.

Keywords Cloud · Hybrid cryptography · Security · Encryption techniques · Decryption techniques · File security

1 Introduction

Cloud computing was engineered to deliver computing services over the internet. Both the fully developed, ready-to-use applications, hardware resources like network, storage are provided according to user's needs. These resources are hosted at the data centers spread globally. The resources are utilized from a configurable pool of similar resources which can be relaxed and provisioned depending on their requirement [1]. According to the definition of NIST, Cloud computing is a sculpt for

S. Sharma · K. Singla · G. Rathee (✉) · H. Saini
Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat, Solan, Himachal Pradesh 173 234, India
e-mail: geetanjali.rathee123@gmail.com

S. Sharma
e-mail: shivam14.cr7@gmail.com

K. Singla
e-mail: singlaheel203@gmail.com

H. Saini
e-mail: hemraj1977@yahoo.co.in

A. K. Luhach et al. (eds.), *First International Conference on Sustainable Technologies for Computational Intelligence*, Advances in Intelligent Systems and Computing 1045,
https://doi.org/10.1007/978-981-15-0029-9_19

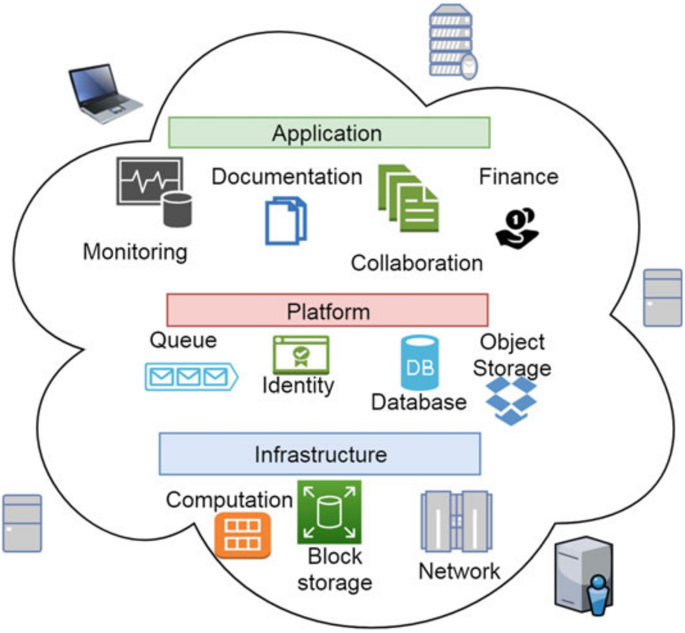


Fig. 1 Wireless mesh network

enabling suitable, on-demand system access to a mutual pool of configurable evaluating resources (e.g., storage, applications, servers, networks, and services) that can be quickly released and provisioned with minimal supervision service provider or effort interaction as depicted in Fig. 1.

- **Need for Security in Cloud Computing**

Although, Cloud computing has grown to be a popular and successful business model due to its attractive characteristics and features. In addition to the reimbursements, its features may result in severe problems specific to cloud security issues [2–4]. The entities whose concern is the security of cloud still hesitate to transfer their business to cloud. The security issues that have been governed barricade of the growth and worldwide use of cloud computing are defined as follows:

– Outsourcing: It refers to providing data to a third-party cloud hosting provider in order to support and deliver IT services that could be handed over in-house. Outsourcing means that the clients physically lose control of their tasks and data. The problem that user's data is with someone else and no longer in proximity of user is a prime security concern.
– Multi-tenancy: It implies that the cloud platform is distributed and shared, exploited by multiple clients. Furthermore, in an environment that is virtual, Different cloud user's data may reside on same physical servers in virtual environment.

A series of security problems and issues such as computation breach, data breach, and flooding attack are experienced.

- **Cryptography**

Converting useful data into unreadable text so that no one else can read it except the pre-determined user is encryption and the techniques used are called cryptographic techniques. It may be accomplished through scrambling words, using code words or using highly efficient mathematical techniques. There are different algorithms classified in two classes [5–8].

a. Symmetric Algorithms—This algorithm uses same key to decrypt and encrypt the message. For instance, if some user desires to convey a message to another and wants nobody else should read it. So, she/he can encrypt the data using a secret key that can be shared with the receiver who will decipher the data using the same key. As can be judged the key can be shared with all the users who ought to retrieve data.
b. Asymmetric Algorithms—These types of algorithms make use of two separate keys in process of decryption and encryption, respectively. Like a user ciphers, the data using one key (known as public key) and receiver decrypts the message using separate key (known as private key). Public Key—This key is nearby to all over the internet and used to cipher the secret text. Private Key—As clear form, the name it is a receiver's private key and only the receiver will be able to use this key to decrypt the required message.

We know that classic encryption schemes have long been used for security purposes and they have been successful to some extent. So why is there a need to use hybrid system? The basic answer is security. Basically, what hybrid cryptography does is that it mixes the effectiveness of symmetric encryption with easiness of public-key encryption. It enhances security level and both types of encryption efficiency. It has several advantages. The new technology was fast, dependable, and data sharing was efficient. There were certainly many benefits to it and had its limitations that were harmful to data owner. As the data was stored on outside system not within the physical reach of user there are the concerns for security of data. Data sharing may also lead to breach of sensitive data and privacy of user. There have cryptographic techniques to cipher the data and code it so that the attacker cannot understand it.

Techniques like DES, 3DES, AES, RSA, RC4 and many others have proven to be a success in hiding the data and securing it. However, everything is prone to attack and every individual encryption technique is known to be attack prone due to an increase in computational power days. Any new technique which will be developed will be known to attacks but to increase the security and efficiency hybrid encryption is used. Hybrid encryption provides effectiveness of public-key cryptography and easiness of private key cryptography. In this paper, a hybrid cryptographic mechanism is proposed by incorporating AES, DES, and RC4 techniques to implement hybrid cryptography. The remaining structure of the paper is structured as follows. Section 2 described the previous security issues proposed by different researchers. The hybrid secure mechanism is portrayed in Sect. 3. The test plan of proposed phenomenon is elaborated in Sect. 4. Further, Sect. 5 represented the performance analysis of encryption and decryption time. Finally, Sect. 6 concludes the paper.

2 Comparative Analysis of Previous Approaches

This research paper introduces cloud architecture and characteristics of cloud. Through the paper, authors have identified present security and privacy issues and discuss their proneness to attack and provide currently used defence mechanisms by analyzing their efficiency in doing the required work [9, 10]. The authors have used current strategies including Virtualization, Data centre Techniques, and MapReduce centre that helps the cloud vendor in overcoming those challenges, but each has its shortcomings and not useful. Author gives a cloud security ecosystem to model different attacks and use different mechanisms against them. Further, the paper provided the security challenges that are a big hindrance in the customer trusting the cloud technology for commercial and personal operations. The issues regarding current defence strategies have not been ratified and left open for future resolve of these issues and enhance the user trust on cloud. Various authors have proposed several cryptographic techniques in order to ensure the security overcloud. The below text describes number of security algorithms and techniques proposed by various author's.

- **Secure Data Sharing in Clouds**

In order of preserving the thoughts of challenges in cloud the researcher has designed a new SeDaSC methodology for securing the facts storage and transfer. The authors have analysed that like cl-pre-scheme, certificates-much-less encryption and el-ghamal cryptography schemes with their blessings and cons. Like the cl-pre-scheme generates a public-non-public key pair and uses bilinear method for encryption which growth the cost of encryption. The certificates much less encryption even though tries to enhance upon the cost issue, however, falls short are in trusted facts garage. On the other hand, the el-ghamal scheme uses bilinear and incremental encryption. However, complexities nonetheless exist [11]. Although, it is clear from the picture that key evaluation time changes merely with augment in file size and the encryption time too is commendable with this methodology. However, the time needs to check with file sizes of greater than 1 GB to really test [12, 13]. Additionally, the important element is that it stores key on outside server that can be underneath outside threats, so measures should be taken to make it extra comfy or limit the level of trust in the server.

- **Brief Study of Encryption Algorithms (RC4, DES, 3DES, and AES) for Information Security**

This paper gives a comparative view of different encryption algorithms on parameters like CPU usage, encryption time, ROM utilization, throughput with different file sizes, length of packet and data type [14–16].

Further, in order to ensure the privacy for cloud providers that provide the services to the clients upon request, plenty of security techniques have been proposed. Ahmad el at. [17] have proposed a game-theoretical model that assists the brokers by identifying the malicious providers that manage and create the federations. The proposed mechanism has been analyzed and simulated over a dataset through CloudSim simulator. In order to ensure security over cloud environment, Akshay et al. [18] have proposed a hybrid cryptographic mechanism that includes both asymmetric and symmetric encryption techniques. Author's have provided multilevel encryption for authenticating the clients using hash technique. The proposed technique is validated by computing the results in CloudSim simulator. From service requested clients to cloud providers, the security has been provided through various cryptographic algorithms including AES, DES, hybrid, hash or MAC algorithms. The scope of this paper is to discuss various cryptographic approaches and propose a hybrid encryption algorithm in order to ensure the security during transmission of file from one place to another. Further, author Shefali et al. [19] have used MD5 and AES encryption techniques for ensuring the security during the login and data access by the user over the cloud. The proposed mechanism need not authenticate the user upon login while during the data accessing, the AES and MD5 algorithms require user authentication.

- **Rivest–Shamir–Adleman (RSA)**

It is a public cryptosystem technique that incorporates block size encryption and variable key size. The steps involved are: Generate two distinct prime numbers, Calculate t as product of two, Now compute phi(t), Find d such that $d * e = 1$, Public Key is $(1, e)$ and Private key is $(1, d)$. It is the most apparent disadvantage is that if two numbers are of massive length then it takes more time and should be of comparable size. The below text elaborated a brief introduction of the standard algorithms.

- **Data Encryption Standard (DES)**

This encrypt algorithm is the most widely used because it works on bits. Length of message at one time to be encrypted is 64 bits and the key size of 64-bits but every $7x$ bit is a check bit which is removed in making of sub-keys. As known, it is one of the best algorithms because no such possible attack is known to crack it other than brute-force which is costly and time-consuming.

- **Triple DES**

Alias 3 DES, it is an extension of DES because it has greater key length. DES was more prone to brute attacks due to increasing computational power. Although it is better version of standard DES, but it can be breached using MITM attacks. So, to defend better against them it can be implemented using secret-key size of 112-bits.

- **Advanced Encryption Standard (AES)**

AES also recognized as Rijndael structure is an iterative algorithm rather being called a fiestel structure. It is iterative because it uses rounds to convert data to ciphertext depending on key sizes.

- **RC4**

RC4 is produced by Bokkos Rivest also called Rivest Cipher four. In this the stream figure is utilized for secret composing of the plain content. On the off chance that the basic square figures don't appear to utilize mackintosh effectively, bit-fluttering assault is doable and furthermore the stream figure assault is moreover powerless on the off chance that they're not legitimately authorized.

- **Blowfish**

Also, a symmetric cipher built as an alternate for more commercial AES and DES with varying key sizes of length from 32 to 448 bits with a block size of 64 bits. It's a 16-round cipher. It is efficient cipher and can be used commercially as alternative to AES. Most possible obvious attacks on this encryption are birthday attacks and should not be used for files of size more than 4 GB.

3 Proposed Solution

The flow graph of proposed approach where the source file is encrypted using the hybrid encryption is shown in Fig. 2. The file is selected and divided into three equal parts using the file system module. Then, each part is encrypted using the AES, DES and RC4 encryption techniques. The encrypted parts are then merged and saved into a single file which, then, can be uploaded on the cloud servers.

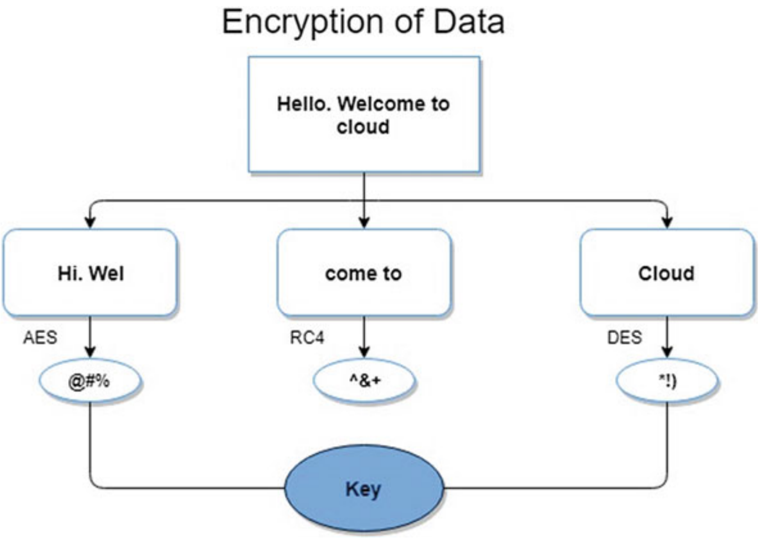


Fig. 2 Process of file encryption

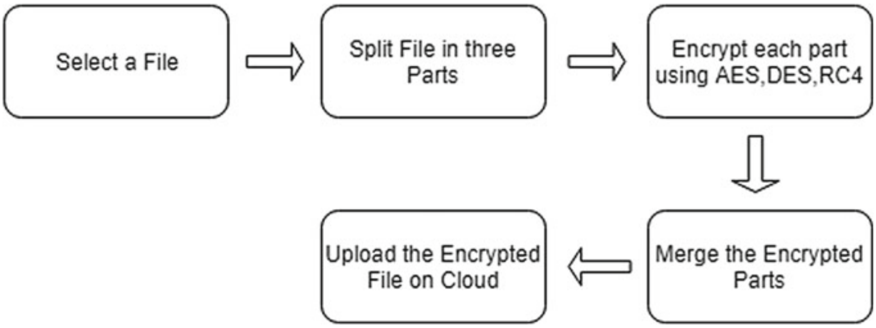

Select a File
Split File in three Parts
Encrypt each part using AES,DES,RC4
Merge the Encrypted Parts
Upload the Encrypted File on Cloud

Fig. 3 Flow diagram of file encryption

File Splitting

The input file is split into three parts so that different encryption algorithms can be applied to each. Split-file module provides an efficient way to split and merge file into multiple parts.

Crypto JS

Crypto-js is a JavaScript library of crypto standards with a growing cluster of secure cryptographic algorithms implemented in JavaScript utilizing best practices and patterns. These algorithms are fast and have a simple and consistent interface. The following algorithms from module Crypto-js are used in this project. AES (Advanced Encryption Standard), DES (Data Encryption Standard), RC4 (Rivest Cipher 4), PBKDF2 (Password-Based Key Derivation Function), PKCS7 (Public Key Cryptography Standards).

File Encryption

The file is encrypted using three encryption algorithms. The steps of file encryption are shown in the given Fig. 3.

File Decryption

Further, for decryption at the receiver side as depicted in Fig. 4, the encrypted file is downloaded from the cloud servers and then split into three parts using a certain special character into three parts whereupon each part is then decrypted using the same techniques which were used for encryption, i.e., AES, DES, and RC4. The decrypted parts are merged into one and the retrieved file can then be used.

4 Test Plan

Testing will be done on different files varying from 1 MB till 30 MB.

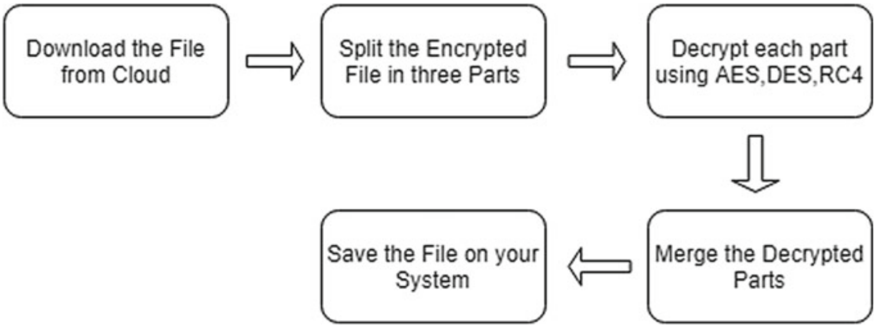

Download the File from Cloud
Split the Encrypted File in three Parts
Decrypt each part using AES,DES,RC4
Merge the Decrypted Parts
Save the File on your System

Fig. 4 Flow diagram of file decryption

Timing for encryption as well as decryption of such files using hybrid algorithm will be noted and compared with the respective encryption and decryption timings of same files using AES encryption alone. Section 5.2 offers a clear view of the time difference among the standard algorithm and the proposed algorithm. This section provides a rough idea of the comparison between the usual algorithm and the proposed algorithm. A file of size 10 MB is taken and is encrypted, firstly, AES and then using the hybrid algorithm. Also, decryption is done of the encrypted file using both, above mentioned, algorithms. The snippet of each algorithm outlines the time required to process the algorithm.

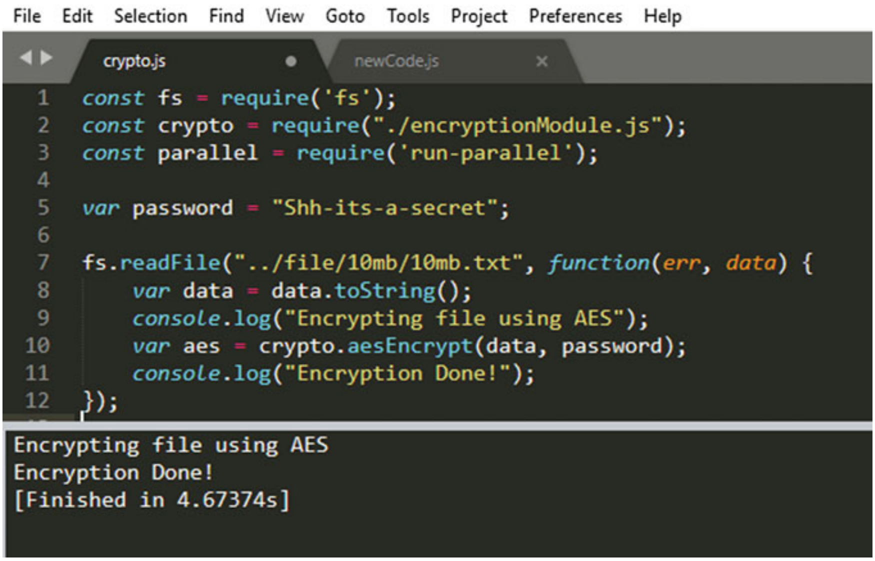
4.1 Encryption and Decryption Time for AES

A file named file.txt of size 10 MB is encrypted using the AES algorithm. Figure 5 shows how much time is required to encrypt the file while Fig. 6 depicts the decryption time.

4.2 Encryption and Decryption Time for Hybrid Algorithm

Now the file is encrypted using the hybrid algorithm (AES, DES, RC4). Figure 7 shows the time taken to encrypt the file.

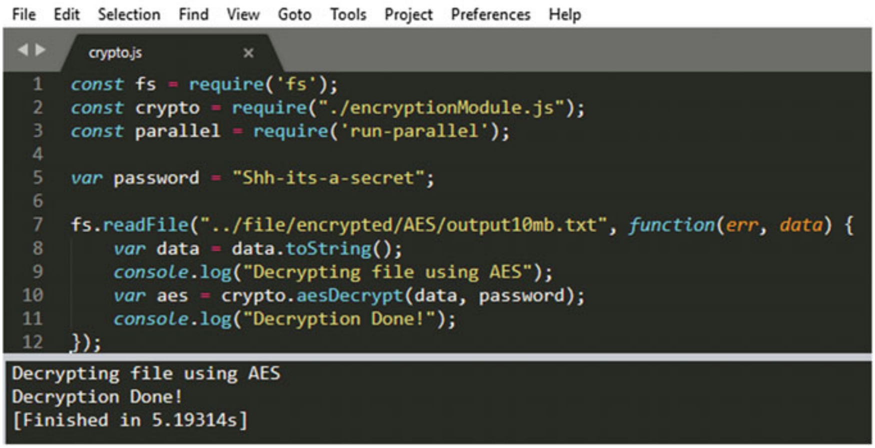
Figure 8 depicts the decryption time of hybrid algorithm. The file of size 10 MB was encrypted and decrypted using the standard algorithm as well as the proposed algorithm. The timings from snippets in section show that the proposed algorithm is both comparatively faster in encryption of the file as well as decryption.


File Edit Selection Find View Goto Tools Project Preferences Help
crypto.js
newCode.js

```
const fs = require('fs');
const crypto = require("./encryptionModule.js");
const parallel = require('run-parallel');

var password = "Shh-its-a-secret";

fs.readFile("../file/10mb/10mb.txt", function(err, data) {
    var data = data.toString();
    console.log("Encrypting file using AES");
    var aes = crypto.aesEncrypt(data, password);
    console.log("Encryption Done!");
});
```

```
Encrypting file using AES
Encryption Done!
[Finished in 4.67374s]
```

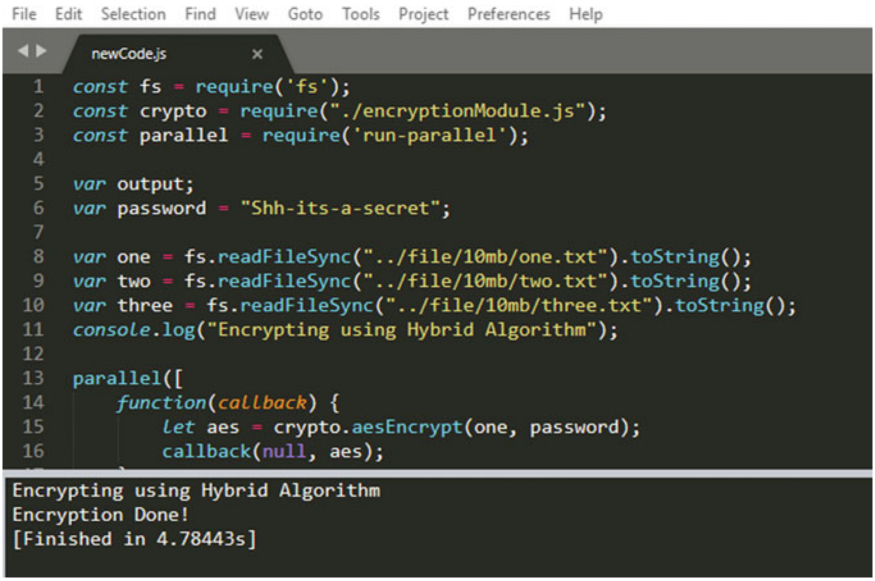
Fig. 5 AES encryption time


File Edit Selection Find View Goto Tools Project Preferences Help
crypto.js

```
const fs = require('fs');
const crypto = require("./encryptionModule.js");
const parallel = require('run-parallel');

var password = "Shh-its-a-secret";

fs.readFile("../file/encrypted/AES/output10mb.txt", function(err, data) {
    var data = data.toString();
    console.log("Decrypting file using AES");
    var aes = crypto.aesDecrypt(data, password);
    console.log("Decryption Done!");
});
```

```
Decrypting file using AES
Decryption Done!
[Finished in 5.19314s]
```

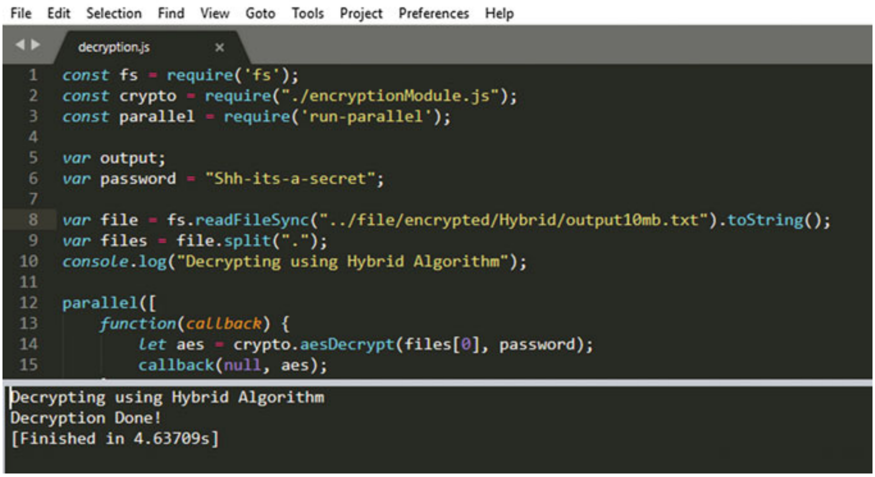
Fig. 6 AES decryption time


File Edit Selection Find View Goto Tools Project Preferences Help
newCode.js

```
const fs = require('fs');
const crypto = require("./encryptionModule.js");
const parallel = require('run-parallel');

var output;
var password = "Shh-its-a-secret";

var one = fs.readFileSync("../file/10mb/one.txt").toString();
var two = fs.readFileSync("../file/10mb/two.txt").toString();
var three = fs.readFileSync("../file/10mb/three.txt").toString();
console.log("Encrypting using Hybrid Algorithm");

parallel([
    function(callback) {
        let aes = crypto.aesEncrypt(one, password);
        callback(null, aes);
```

```
Encrypting using Hybrid Algorithm
Encryption Done!
[Finished in 4.78443s]
```

Fig. 7 Hybrid encryption time


File Edit Selection Find View Goto Tools Project Preferences Help
decryption.js

```
const fs = require('fs');
const crypto = require("./encryptionModule.js");
const parallel = require('run-parallel');

var output;
var password = "Shh-its-a-secret";

var file = fs.readFileSync("../file/encrypted/Hybrid/output10mb.txt").toString();
var files = file.split(".");
console.log("Decrypting using Hybrid Algorithm");

parallel([
    function(callback) {
        let aes = crypto.aesDecrypt(files[0], password);
        callback(null, aes);
```

```
Decrypting using Hybrid Algorithm
Decryption Done!
[Finished in 4.63709s]
```

Fig. 8 Hybrid decryption time

Table 1 Comparision of different encryption algorithms

Factors	DES	AES	RC4
Created by	IBM in 1975	Vincent Rijmen, Joan Daemen in 2001	Ron Rivest in 1987
Key length	56 bits	128, 192 or 256 bits	40–2048 bits
Round(s)	16	10, 12 or 14	1
Block size	64 bits	128 bits	2064 bits (1684 effective)
Speed	Slow	Fast	Fast
Security	Not secure enough	Excellent security	Adequate security

5 Result and Performance

In this section, various cryptographic algorithms are compared based on their key lengths, number of rounds, block sizes and other assets.

5.1 Algorithms Testing

Testing on various files will be done using AES (Standard Algorithm) and proposed algorithm (Hybrid Algorithm—AES, DES and RC4) so that the performance of the algorithm can be recorded in relation to the increasing size of the file. The files used in the tests vary from 1 MB till 30 MB. Graphs of both the algorithms along with their tables are shown below. Table 1 depicts the comparison of different encryption techniques over various parameters.

5.2 AES Testing

Encryption of file sizes ranging 1, 5, 10, 20 and 30 MB is done using AES Encryption. Encryption time of these files is calculated and plotted on a graph depicted in Fig. 9. Similarly, decryption of file sizes ranging 1, 5, 10, 20 and 30 MB is done using AES decryption algorithm. Decryption time of these files is calculated and plotted on a graph depicted Fig. 10.

5.3 Hybrid Encryption (AES, DES, RC4) Testing

Again, encryption of file sizes ranging 1, 5, 10, 20, and 30 MB is done using Hybrid Encryption. As already done on the above process, encryption time of these files is calculated and plotted on a graph depicted in Fig. 11.

Fig. 9 AES encryption time

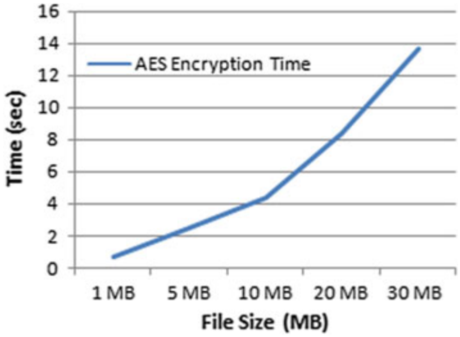


Fig. 10 AES decryption time

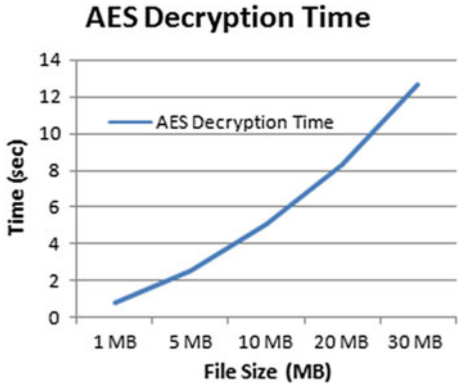
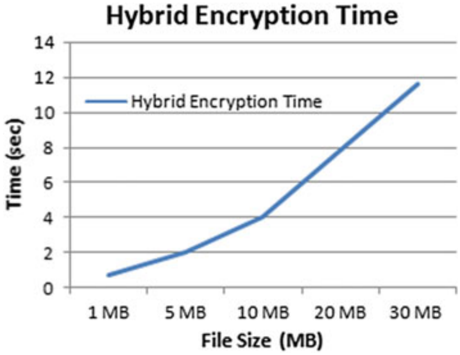


Fig. 11 Hybrid encryption time



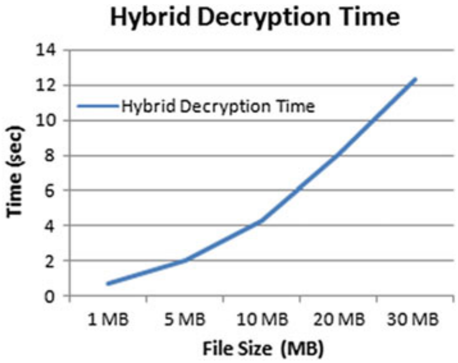


Fig. 12 Hybrid decryption time

Similarly, decryption of file sizes ranging 1, 5, 10, 20, and 30 MB is done using hybrid decryption algorithm. Decryption time of these files is calculated and plotted on a graph depicted in Fig. 12.

5.4 *Comparison of Encryption Algorithms Testing Data*

Bih-Hwang et al. [20] have ensured the cloud data security from third-party data models. Here, the author has used cloud platform know as Heroku that is responsible for managing and integrating the services using cryptographic algorithms. Lee et al. have used AES encryption technique to provide Heroku as cloud platform and provide data security from third parties. The authors have analyzed and validated the proposed mechanism by analyzing the encryption time over various file sizes. In this paper we have used this as over base paper and compared the proposed hybrid encryption over AES with different file sizes. Comparisons of the encryption time as well as decryption time of the files ranging 1–30 MB is shown in Table 2. Hybrid algorithm needs 10–15% less time for file to be encrypted in comparison to other encryption techniques. With single encryption algorithm such kind of data security cannot be provided.

The Graphs presented in Figs. 13 and 14 depicts the encryption time and decryption time of files with various files. As depicted in Fig. 13, AES approach ensures the

Table 2 Comparision of different algorithms

File size (MB)	AES encryption	Hybrid encryption	AES decryption	Hybrid decryption
1	0.717	0.732	0.789	0.685
5	2.544	2.076	2.499	2.034
10	4.457	4.021	5.043	4.232
20	8.429	7.875	8.325	8.063
30	13.710	11.618	12.645	12.328

Fig. 13 Time comparison between AES and hybrid encryption algorithm

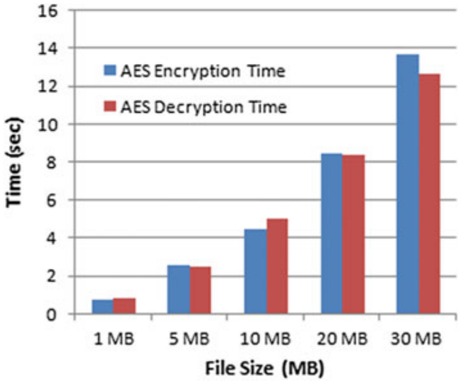
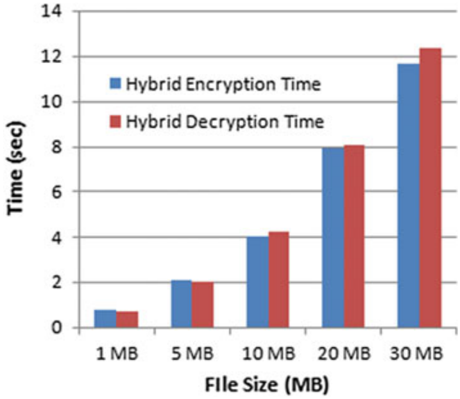


Fig. 14 Time comparison between AES and hybrid decryption algorithm



file security with the increase of file size, however, proposed approach gives better results as compare to AES because of hybrid encryption process. As the file size increases, proposed mechanism performs better. Similarly, the proposed algorithm performs better for the decryption process as presented in Fig. 14 with increasing file size, the decryption time taken for the hybrid algorithm gradually decreases in contrast to the AES algorithm.

6 Conclusion

By the investigation of the accompanying outcomes, we could reason that the hybrid algorithms of AES, DES, and RC4 gave us better execution time in contrast with that of AES algorithm alone. We saw that for moderately little document sizes AES individually produced better throughput in bytes per millisecond when contrasted with that of hybrid algorithm including AES, DES, and RC4. However, as the measure of record augmented, proposed algorithm indicated better outcomes. Mulling over, the substantial measure of information that business applications will in general store on the cloud, record sizes can fluctuate to extensive numbers, thus the utilization of hybrid algorithm calculation is proposed to actualize staggered security on cloud information storage. Based on the examination of execution of single encryption calculations and multiple encryption calculations, it is concluded that hybrid encryption involving multiple encryption algorithms (AES, DES, and RC4) provided much better results than single algorithm (AES) implemented alone. In the future, we plan to use certain other encryption algorithms such as Blowfish and other public-key encryption techniques like RSA to be implemented in the project.

References

1. Ali, M., Dhamotharan, R., Khan, E., Khan, S.U., Vasilakos Athanasios V., Li, K., Zomaya Albert, Y.: SeDaSC: secure data sharing in clouds. IEEE Syst. J. **11**(2), 395–404 (2017)
2. Xiao, Z., Yang, X.: Security and privacy in cloud computing. IEEE Commun. Surv. Tutor. **15**(2), 843–859 (2013)
3. Meng, S., Wang, Y., Jiao, L., Miao, Z., Sun, K.: Hierarchical evolutionary game based dynamic cloudlet selection and bandwidth allocation for mobile cloud computing environment. IET Commun. **13**(1), 16–25 (2018)
4. Esposito, C., Castiglione, A., Pop, F., Choo, K.-K. R.: Challenges of connecting edge and cloud computing: a security and forensic perspective. IEEE Cloud Comput. **4**(2), 13–17 (2017)
5. Park, J.-E., Park, Y.-H.: Fog-based file sharing for secure and efficient file management in personal area network with heterogeneous wearable devices. J. Commun. Netw. **20**(3), 279–290 (2018)
6. Zhang, H., Zhou, Z., Ye, L., Du, X.: Towards privacy preserving publishing of set-valued data on hybrid cloud. IEEE Trans. Cloud Comput. **6**(2), 316–329 (2018)
7. Thakur, J., Kumar, N.: DES, AES and Blowfish: symmetric key cryptography algorithms simulation based performance analysis. Int. J. Emerg. Technol. Adv. Eng. **1**(2), 6–12 (2011)
8. Li, J., Li, Y.K., Chen, X., Lee, P.P.C., Lou, W.: A hybrid cloud approach for secure authorized deduplication. IEEE Trans. Parallel Distrib. Syst. **26(**5), 1206–1216 (2015)
9. Seo, S.-H., Nabeel, M., Ding, X., Betrino, E.: An efficient certificateless encryption for secure data sharing in public clouds. IEEE Trans. Knowl. Data Eng. **26**(9), 2107–2119 (2014)
10. Chen, D., Li, X., Wang, L., Khan, S.U., Wang, J., Zeng, K., Cai, C.: Fast and scalable multi-way analysis of massive neural data. IEEE Trans. Comput. **64**(3), 707–719 (2015)
11. Shiu, Y-S., Chang, S.Y., Wu, H.-C., Huang, S.C.-H., Chen, H.-H.: Physical layer security in wireless networks: a tutorial. IEEE Wirel. Commun. **18**(2), 66–74 (2011)
12. Wei, , Zhu, H., Cao, Z., Dong, X., Jia, W., Chen, Y., Vasilakos, A.V.: Security and privacy for storage and computation in cloud computing. Inf. Sci. **258**, 371–386 (2014)
13. Singh, S.P., Manini, R.: Comparison of data encryption algorithms. Int. J. Comput. Sci. Commun. **2**(1), 125–127 (2011)
14. Singhal, N., Raina, J.P.S.: Comparative analysis of AES and RC4 algorithms for better utilization. Int. J. Comput. Trends Technol. **2**(6), 177–181 (2011)
15. Sanaei, Z., Abolfazli, S., Gani, A., Buyya, R.: Heterogeneity in mobile cloud computing: taxonomy and open challenges. IEEE Commun. Surv. Tutor. **16**(1), 369–392 (2014)
16. Rong, C., Nguyen Son, T., Jaatun, M.G.: Beyond lightning: a survey on security challenges in cloud computing. Comput. Electr. Eng. **39**(1), 47–54 (2013)
17. Hammoud, A., Otrok, H., Mourad, A., Wahab, O.A., Bentahar, J.: On the detection of passive malicious providers in cloud federations. IEEE Commun. Lett. **23**(1), 64–67 (2019)

18. Arora, A., Khanna, A., Rastogi, A., Agarwal, A.: Cloud security ecosystem for data security and privacy. In: 7th IEEE International Conference on Cloud Computing, Data Science & Engineering-Confluence, pp. 288–292 (2017)
19. Ojha, S., Rajput, V.: AES and MD5 based secure authentication in cloud computing. In: IEEE International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp. 856–860 (2017)
20. Lee, B.-H., Dewi, E.K., Wajdi, M.F.: Data security in cloud computing using AES under HEROKU cloud. In: 27th IEEE Conference on Wireless and Optical Communication Conference (WOCC), pp. 1–5 (2018)