Custom-Aes: A Novel Framework To Enhance Data Security In Cloud Environment

R.Caroline Kalaiselvi

Associate Professor and Research Scholar, PG and Research Dept. of Computer Science,
Presidency College, Chennai, Tamilnadu, India.
carrie.jonna@gmail.com

Dr. S. Mary Vennila

Associate Professor & Head, PG and Research Dept. of Computer Science,
Presidency College, Chennai, Tamilnadu, India.
vennilarhymend@yahoo.co.in

Abstract

The exponential growth in clouds' computing has an affinity to an all-encompassing computing skill naturally overstretching the already strained client logistics. With all pervading security threats looming large on the horizon, sourcing service from these remote data centers would always remain a constant irritant on both sides of the users. As the morphology of the security threats evolve in leaps and bounds as an ongoing process, a highly sensitive encrypted algorithm woven into the Advanced Encryption Standard (AES) becomes a necessity. This study seeks to introduce a system with key features, including improved protection and data privacy for the owner. To boost the speed of the encryption method, it modifies the 128 AES algorithm. Custom-AES-Algorithm parameters such as block-size, secret-key size, salt-size and cipher-mode have been fine tuned. This is seen to minimize the time taken for both encryption and decryption, less memory used and entropy maximization. Proposed system therefore increases security and minimizes the use of memory. In our paper, most popular algorithms, AES, RSA and blowfish shed light on the overall performance and justification made for Custom AES.

***Keywords**: Custom AES; RSA; Blowfish; Security.*

1.Introduction

Cloud computing today is the most attractive option found to be employed in varied services with diverse approaches to the design of appropriate software [1]. (PaaS),(SaaS), and (IaaS) are service models. Four cloud platform deployments rely on architectural keys for public, private, group and hybrid systems [2].

The advantages of cloud computing are the flexibility and ability in application when they are coupled with conventional computing or storage methods [3]. However, Computational clouds are associated with a plethora of security problems, including i) cloud service provider privacy and security issues and (ii) issues related with customers [4]. There are suggestions in the findings with regard to cyber-attack the AES algorithm have been suggested in the literature [5], for example a variety of fault analyses that corrupt the structures of the AES with intent to retrieve the specific data targeted [6].

Having regard to the fact that each and every activity of mankind in every sphere is nowadays increasingly confined to computer applications, the cloud computing standards suggest improved practices ensuring optimization of computer resources [7, 8]. Moreover, data centre cloud storage is very convenient for users only before their data is stored and accessed remotely at any time [9,10]. On the opposite, stability is the primary issue in data storage in cloud. Consequently, Cloud Data Centers should have specific processes designed to ensure perfection in storage and coherence of data stored on the cloud [11]. One or two attributes at a time are used by existing security systems, that is low level of security and consumption of more time in encrypting/decrypting data.

This entails this method consuming more time and the resultant increase in the network access and power consumption and network breakdown [12-16]. Cloud computing is a type of network that efficiently shares data and resources, so users need to be provided with security, since security the quite essential feature of cloud computing. Therefore, it becomes obligatory on the providers of cloud storage to optimize the security in all features, minimum use of power, latency of network and lesser time taken to accomplish [17-23]. It is a known fact that the techniques that are in use currently do not adequately measure the security enveloping the cloud services.

The cloud computing rests on a solid foundation of a secure framework that paves the way for the ease in managing things like easy accessibility to computing making sure it is cost-effective also. The encryption and decryption process should therefore consume low memory, time and efficient entropy that strengthened the security of aspect of data computing. By putting in place a new encryption/decryption system, this paper contributes to the design of this enhanced framework that can be accessed by the providers of cloud computing integrating the finer details of the components of this architecture for the benefit of certain category of cloud users and providers of cloud computing who use similar security framework. The structure envisaged comprises key features of improved protection and data privacy for the owner. The new enhanced 256 Custom AES algorithm to improve the speed of the encryption and decryption processes.

2.Related Work

Theoretical parallels are made with findings and implementations, but not supported. We implemented application's algorithms and calculated efficiency of cost and response time related to performance. Two big algorithms, such as Custom-AES-Algorithm & AES-Algorithm, encrypt experiments performed with 'n' number of files uploaded by end-users. The end-user selects the desired encryption algorithm to be used during the registration itself. In Custom-AES-algorithm, five set of modifications made such as cipher mode, block size, secret key length, IV key length and salt generation length. These are taken into account to estimate performance of Custom-Advanced-Standard-Encryption algorithm and compared with existing works, like Advanced-Standard-Encryption algorithm RSA and Blowfish.

Major aim of this paper is to prove Advanced-Standard-Encryption (AES) algorithm is much more efficient than previous Rivest-Shamir-Adleman and Blowfish-algorithm. Comparison of Custom-AES-algorithm and AES-algorithm encryption-algorithms produce an efficient solution for data-security. This approach has very sound realistic knowledge of crypto-graphic technique and outcome of proposed-research will reduce time-consumption during encryption-process and improved security level during decryption-process too.

The metrics of the time taken for encryption and decryption reveals the effectiveness of this application. Entropy reveals the randomness. And, no entropy metrics to calculate cryptographic power and attack resistance has been used in any experiments so far. Memory usage by all the algorithms indicated the best algorithm.

3.Algorithms used in the Implementation

3.1. Blowfish Algorithm

The advent of the Blowfish algorithm dates back to 1993. Bruce Schneier provided this hugely popular algorithm. It is a variable length key of 64-bit block cypher. A number of research analyses demonstrated the superiority of the Blowfish algorithm over others. Measured for throughput and power consumption Blowfish rated better than other algorithms [24].

Algorithm

```
Divide x into two 32-bit halves: yL, yR
For i = 1 to 16:
yL = yL XOR Pi
yR= F(yL) XOR yR
Swap yL and yR Next i
Swap yL and yR(Undo the last swap.)
yR= yR XOR P17
yL= yL XOR P18
Recombine yL and yR
```

3.2 RSA Algorithm

The RSA algorithm encryption method has been widely in use since 1977 ever since they were authenticated by Ron Rivest, Adi Shamir and Leonard Adelman. Being a fast cipher this algorithm is used for private and public key generation [25].

```
Algorithm
Key Generation: KeyGent(x, y)
Input: Two large primes – x, y
Compute n = x .y
φ (n) = (x - 1)(x - 1)
Choose b such that gcd(b, φ (n)) = 1
Determine e such that b .d ≡ 1 mod φ (n)
Key: public key = (b, n)
secret key= (d, n)
Encryption: ci = mi^b mod n
where ci is the cipher text and mi is the plain text.
```

RSA has a unique property, i.e., by multiplying cypher texts, it is possible to find the product of plain text. The cypher text of the product will be the outcome of the operation.

Given cl = E(ml) = mi^b mod n, then

(ci1 . ci2) mod n = (mi1 . mi2)^b mod n

3.3 AES Algorithm

The Advanced Encryption Standard is the latest for encryption suggested by NIST in the place of DES. The only documented successful breach against it is the Brute Force attack. It is nothing but guessing all possible combinations to break through the encryption. AES and DES are cyphers for bricks, the key length of AES is 128, 192, or 256 bits, 256 by default.

In 10,12 and 14 rounds, the key size determines the encryption of data blocks of 128 bits. The AES encryption is flexible, comfortable with small devices capable of performing on different platforms. AES was carefully checked with regard to several security applications [25][26].

Algorithm

```
Cipher1(byte1[] input1, byte1[] output)
{
 byte1[4,4] State;
copy input1[] into State1[] AddRoundKey
for (round1 = 1; round1 < Nr-1; ++round1)
{
SubBytesShiftRowsMixColumnsAddRoundKey1
}
 SubBytesShiftRowsAddRoundKey1
copy State1[] to output[]
}
```

3.4 Custom AES Algorithm

Custom-AES-Algorithm has been tuned for the following modifications

- Block-size Changed
- Secret-key size Changed
- Salt-size Changed
- Cipher-mode Changed
- IV Key Length Changed

These modifications made to the Custom-AES-Algorithm has resulted in a higher degree of security, precision, performance, etc.

4.Implementation

Custom AES with standard AES, blowfish and RSA have been introduced and compared. Employing Eclipse IDE, we introduced the java algorithms. They used Java security and crypto packages. These include features including encryption, decryption, generation of keys, infrastructure for key management, authentication, and authorization. Blowfish, however, is not part of the setup for Java security and crypto. In Java, Blowfish is introduced, converted into a container, and the Blowfish jar is added to the crypto list externally. Each file's encrypted output is stored as a file which turns out to be input for decryption. The same input files were used in the experiment for the purpose of comparison.

For all implementations and analytical work, we have used the same method to maintain the memory and processor conditions. In the same ECB mode, all block cypher algorithms are set which by default is the Java cryptosystem and security. The classes and interfaces implementing the Java security included in this package.

5.Evaluation Parameters

Each of the encryption techniques has its own strength and weak points. We should be aware of the efficiency and the working pattern of the algorithms to apply it for an application. Therefore, based on many characteristics, these algorithms must be evaluated. This paper analyses the following metrics in which cryptosystems can be correlated with each other. There are strong and weak points in each of the encryption techniques, the shortcomings of which could be overcome with the knowledge of the strength and weakness, efficiency of the cryptographic algorithms that are to be applied. Therefore, based on many characteristics, these algorithms must be evaluated. The following metrics are analysed in this paper so that the cryptosystems could be correlated with each other.

5.1 Encryption time

This is the time duration to transform plaintext into ciphertext. The time consumed is dependent on the key and block size of plaintext and mode. In this paper we estimated milliseconds of encryption time which tells upon the performance of the device, less encryption time renders the device fast and sensitive.

5.2 Decryption time

This is the time duration retrieve plaintext from ciphertext. The decryption time should be less compared to the encryption time and then only it would make the device sensitive and quick as the time taken for decryption impacts the performance of the device. We came across milliseconds of decryption time in our experiment.

5.3 Memory used

Different memory sizes are required for implementation of various encryption techniques. This again rely on the number of operations of the algorithms, the size of the key used, the vector initialization used and the method of operation involved. The memory consumed reveals the expense of the application. A limited consumption of memory would be most appropriate.

5.4 Entropy

In cryptographic processes, randomness is an essential property, since an attacker should not be able to guess data. Entropy in the data is a test of randomness. It tests knowledge ambiguity. We need security algorithms in information security to generate high randomness in the encrypted message, so that there is little or no dependency between key and ciphertext.

The relationship between key and ciphertext becomes complex with elevated randomness. Confusion is also called this property. A high degree of uncertainty is expected to make it hard for an intruder to guess. Using Shannon's formula, we compute entropy.

6.Results and Discussions

6.1 Encryption Time

Captured data will be plotted into a graph against the time taken for encryption and the amount of total files processed by end-users during file encryption processing. Failures are also captured, such as incorrect-key, padding-error, incorrect salt-size, incorrect block-size, etc. But during graph plotting, these kinds of data are not taken into consideration.

In order to show time taken for encryption alone, we only plotted success cases into a graph, since if we plot failure cases, the time taken for encryption will be zero and it looks like it has given more efficiency.

Encryption time of proposed Custom-AES-Algorithm is compared with existing AES, RSA and blowfish in table 1.

Table 1. Encryption Time of Custom AES with AES, RSA, Blowfish

No. of files	**ENCRYPTION TIME**			
	CUSTOM AES	**AE**	**RSA**	**BLOWFISH**
2	**0.1**	**0.1**	**0.1**	**0.13**
4	**0.19**	**0.2**	**0.23**	**0.26**

6	**0.29**	**0.27**	**0.34**	**0.38**
8	**0.28**	**0.28**	**0.31**	**0.39**
10	**0.49**	**0.49**	**0.53**	**0.64**
12	**0.68**	**0.58**	**0.67**	**0.86**
14	**1.04**	**1.14**	**1.12**	**1.25**
16	**1.1**	**1.18**	**1.23**	**1.3**

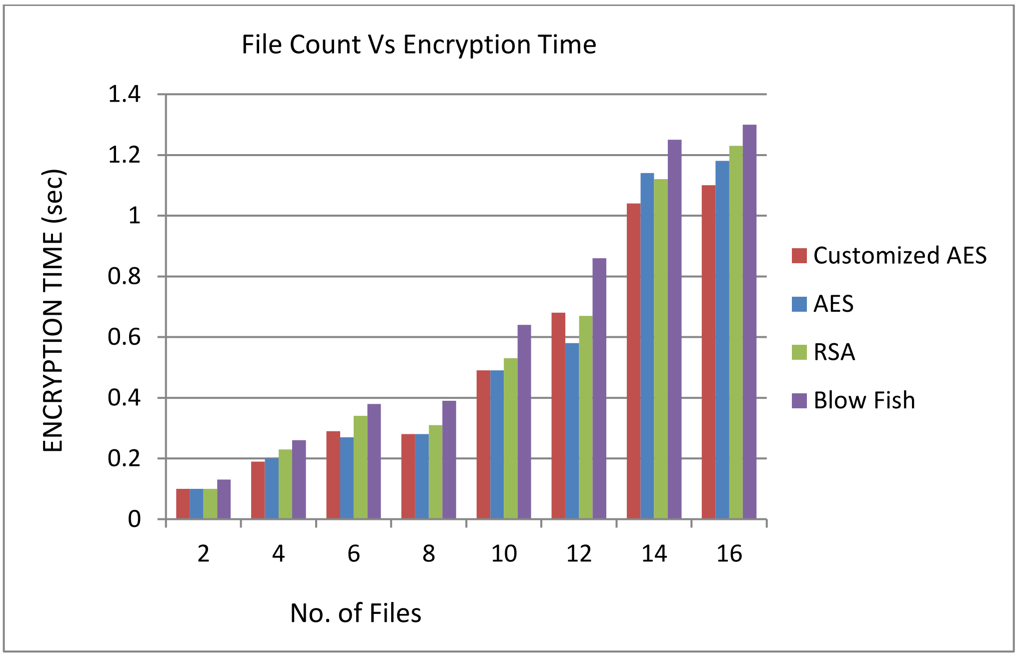

Fig. 1. Encryption time by proposed Custom-AES-Algorithm with existing AES, RSA and Blowfish algorithm.

6.1 Decryption Time

Decryption time of proposed Custom-AES-Algorithm is compared with existing AES ,RSA and blowfish in table 2.

Table. 2. Decryption Time of Custom AES with AES, RSA, Blowfish

	DECRYPTION TIME			
	CUSTOM AES	**AES**	**RSA**	**BLOWFISH**
2	**0.1**	**0.1**	**0.1**	**0.15**
4	**0.2**	**0.22**	**0.23**	**0.25**
6	**0.22**	**0.26**	**0.36**	**0.39**
8	**0.21**	**0.27**	**0.38**	**0.48**
10	**0.41**	**0.43**	**0.54**	**0.68**
12	**0.62**	**0.64**	**0.65**	**0.91**
14	**1.14**	**1.17**	**1.15**	**1.27**
16	**1.12**	**1.19**	**1.22**	**1.29**

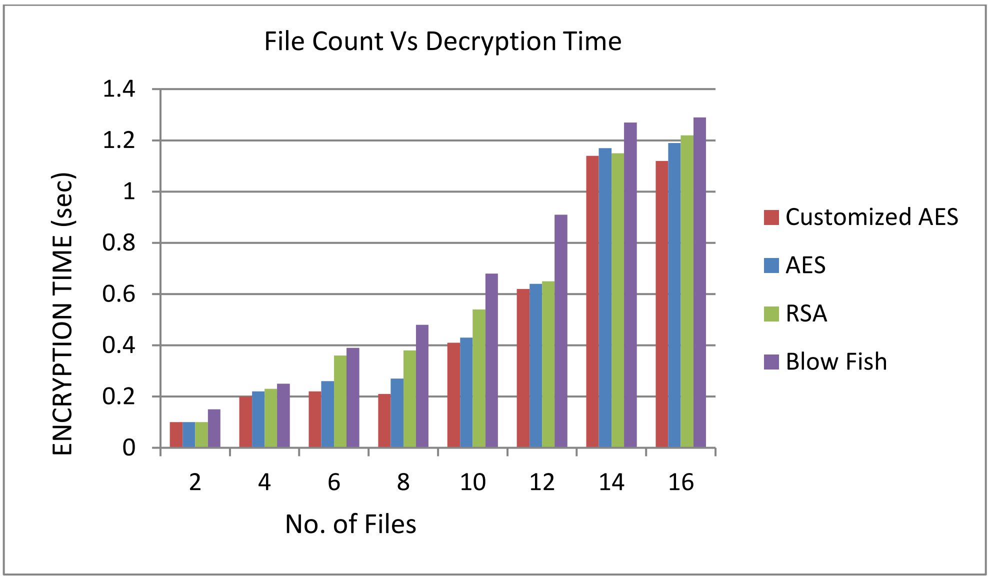

Fig. 2. Decryption time by proposed Custom-AES-Algorithm with existing AES, RSA and Blowfish algorithm.

6.3 Entropy

Table. 3. Entropy values of Custom AES with AES, RSA, Blowfish

	Custom AES	AES	RSA	Blowfish
Average entropy per byte of encryption	3.98322	3.84024	3.0958	3.93891

Table 3 shows that Custom AES reveals the highest average entropy. Entropy is a measure of the degree to which knowledge is random. A significant and desirable property of cryptographic algorithms is randomness. S boxes and p boxes are extensively used by Custom AES.

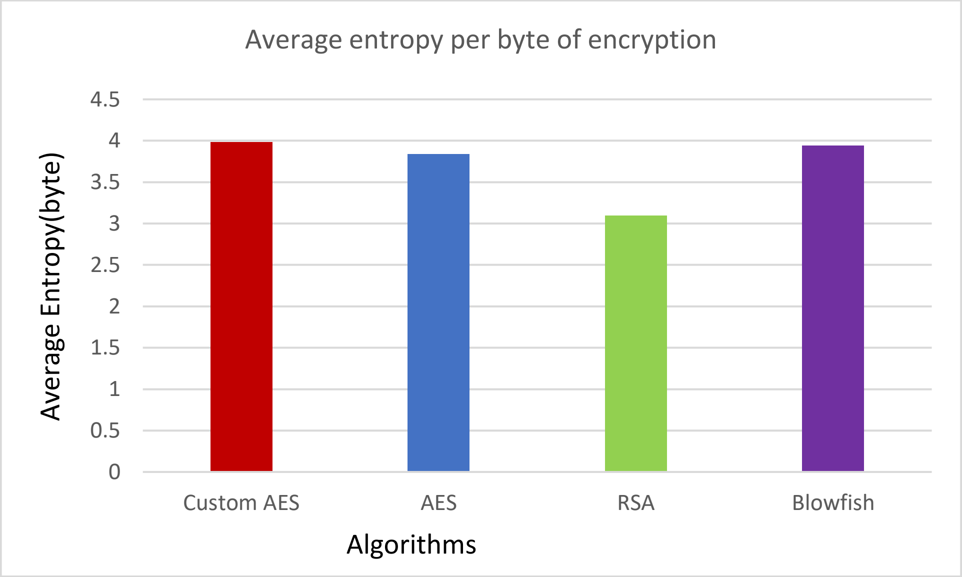

Fig. 3. Average Entropy values of proposed Custom-AES-Algorithm with existing AES, RSA and Blowfish algorithm.

Both Custom AES and Blowfish therefore generate a high degree of randomness in resultant data, rendering the result less vulnerable to attacks.

6.4 Memory used

Table. 4. Memory used by of Custom AES with AES, RSA, Blowfish

Algorithm	Memory required for implementation (KB)
CUSTOM AES	**4.1**
AES	**7.9**
RSA	**14.3**
BLOWFISH	**10.8**

The memory used for unit operations for the mentioned algorithms is shown in Table 4.

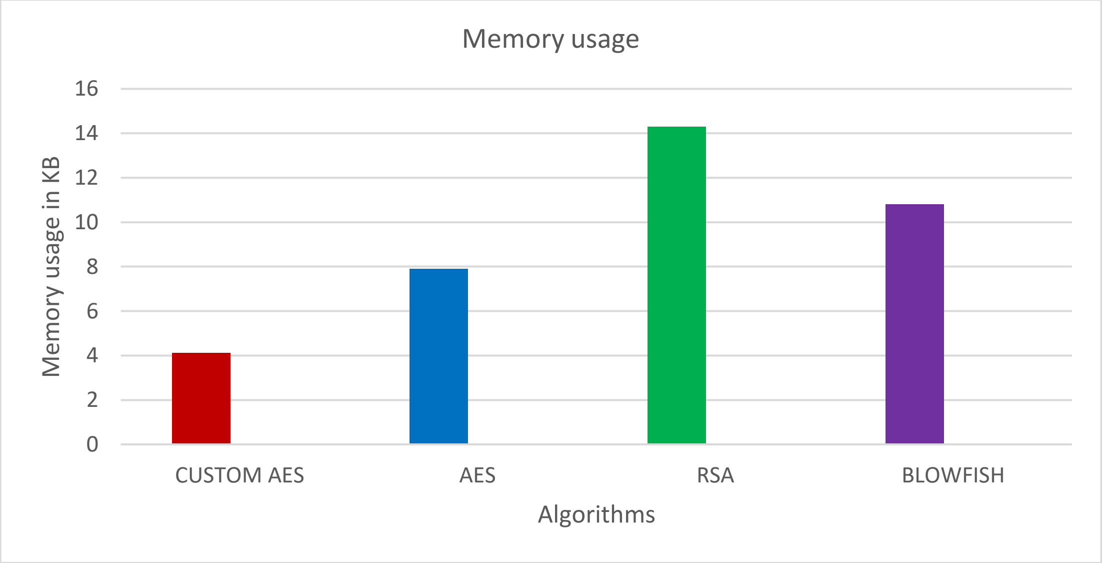

Fig. 4. Memory usage of proposed Custom-AES-Algorithm with existing AES, RSA and Blowfish algorithm.

Fig 4 indicates the memory usage of all the algorithms. Operation Memory space has depended upon program length, that mean memory utilization is directly depend on length of the program; if any program has used large amount of code then we can say that large memory space will be required to execute. In the proposed algorithm we have used simple and small code. From above figure, it is clear that the memory required for implementation is smallest in the proposed algorithm whereas it is largest in RSA. AES and Blowfish require medium size of memory. Therefore, if the demand of any application is the smallest memory size; the proposed algorithm is the best option

7.Conclusion

The method of cryptography has its own dimensions. The shortcomings can be addressed by employing the knowledge gained by the algorithms and choosing the effective cryptographic algorithm. From our experiment, it became apparent that less memory required for implementation, the best entropy metric, the minimized time of encryption and decryption. The results suggest that Custom AES needs the minimum time for encryption and decryption. If time and space happen to be main factors in the application, then the most suitable algorithm could be none other than Custom AES.

References

[1] G.S. Mahmood, J. H. Dong, and B. A. rahman Jaleel, "Achieving an effective, confidentiality and integrity of data in cloud computing," International Journal of Network Security, vol. 21, no. 2, pp. 326–332, 2019.

[2] S. Othman and A. S. Riaz, "A user-based trust model for cloud computing environment," International Journal of Advanced Computer Science and Applications, vol. 9, no. 3, 2018.

[3] A. Firman, A. N. Hidayanto, and P. Harjanto, "Critical components of security framework for cloud computing community: a systematic literature review," International Journal of Pure and Applied Mathematics, vol. 118, no. 18, pp. 3345–3358, 2018.

[4] K. V. Pradeep, V. Vijayakumar, and V. Subramaniyaswamy, "An efficient framework for sharing a file in a secure manner using asymmetric key distribution management in cloud environment," Journal of Computer Networks and Communications, vol. 2019, Article ID 9852472, 8 pages, 2019.

[5] Dr. Ramalingam Sugumar and K. Arul Marie Joycee, "FEDSACE: a framework for enhanced user data security algorithms in cloud computing environment," International Journal on Future Revolution in Computer Science & Communication Engineering, vol. 4, no. 3, 2018.

[6] M. Kpelou and K. Kishore, "Lightweight security framework for data outsourcing and storage in mobile cloud computing," International Journal of Recent Technology and Engineering, vol. 8, no. 2, 2019.

[7] R. Ganga Sagar and N. Ashok Kumar, "Encryption based framework for cloud databases using AES algorithm," International Journal of Research Studies in Computer Science and Engineering, vol. 2, no. 6, 2015.

[8] J. R. Jain and A. Abu, "A novel data logging framework to enhance security of cloud computing," in Proceedings of the SoutheastCon 2016, IEEE, Norfolk, VA, USA, April 2016.

[9] J. Singh, "Framework for client side AES encryption technique in cloud computing," IJIRMPS, vol. 6, no. 5, 2018.

[10] J. Y. Gudapati Syam Prasad, S. sunil kumar, and A. Keerthi, "Integration of searching and AES encryption in cloud computing," International Journal of Engineering and Advanced Technology (IJEAT), vol. 8, no. 4, 2019.

[11] I. A. Elgendy, W.-Z. Zhang, C.-y. Liu, and C.-H. hsu, "An efficient and secured framework for mobile cloud computing," IEEE Transactions on Cloud Computing, 2018.

[12] R. Saha, G. Geetha, G. Kumar, and T.-h. Kim, "RK-AES: an improved version of AES using a new key generation process with random keys," Security and Communication Networks, vol. 2018, Article ID 9802475, 11 pages, 2018.

[13] O. C. Abikoye, A. D. Haruna, A. Abubakar, N. O. Akande, and E. O. Asani, "Modified advanced encryption standard algorithm for information security," Symmetry, vol. 11, no. 12, p. 1484, 2019.

[14] K.-L. Tsai, Y.-L. Huang, F.-Y. Leu, I. You, Y.-L. Huang, and C.-H. Tsai, "AES-128 based secure low power communication for LoRaWAN IoT environments," IEEE Access, vol. 6, pp. 45325–45334, 2018.

[15] M.V. C. Suana, A. M. Sison, C. Aragon, and R. P. Medina, “Enhancement of advanced encryption standard (AES) cryptographic strength via generation of cipher key-dependent S-box,” International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 6, no. 4, 2018.

[16] S. NurRachmat, “Performance analysis of 256-bit AES encryption algorithm on android smart phone,” IOP Conf. Series: Journal of Physics: Conf. Series, vol. 1196, 2019.

[17] J. Silki and V. Abhilasha, “An improved security framework for cloud environment using ECC algorithm,” International Journal for Research in Applied Science & Engineering Technology, vol. 6, no. 1, 2018.

[18] A. Oussama and Z. Abdelha, “A security framework for cloud data storage (CDS) based on agent,” Applied Computational Intelligence and Mathematical Methods, Springer, Berlin, Germany, 2019.

[19] H. J. Muhasin, R. Atan, M.A. Jabar, and S. Abdullah, “Cloud computing sensitive data protection using multi layered approach,” in Proceedings of the 2016 2nd International Conference on Science in Information Technology (ICSITech), pp. 69–73, Balikpapan, Indonesia, October 2016.

[20] K. Ravi and K. B. Rajesh, “Quality based cloud service broker for optimal cloud service provider selection,” International Journal of Applied Engineering Research, vol. 12, no. 18, pp. 7962–7975, 2017.

[21] M. Adelmeyer, M. Walterbusch, B. Peter, and T. Frank, Trust Transitivity and Trust Propagation in Cloud Computing Ecosystems, Lawrence Erlbaum Associates, Mahwah, NJ, USA, 2018.

[22] F. Meng, R. Lin, Z. Wang, H. Zou1, and S. Zhou, “A multiconnection encryption algorithm applied in secure channel service system,” EAI Endorsed Transactions on Security and Safety, vol. 5, no. 15, 2018.

[23] H. A. Al Essa and A. S. Ashoor, “Enhancing performance of AES algorithm using concurrency and multithreading,” ARPN Journal of Engineering and Applied Sciences, vol. 14, no. 11, 2019.

[24] Mr. Gurjeevan Singh, Mr.AshwaniSingla and Mr. K S Sandha “Cryptography Algorithm Compassion for Security Enhancement InWireless Intrusion Detection System” International Journal of Multidisciplinary Research Vol.1 Issue 4, August 2011.

[25] Uma Somani, “Implementing Digital Signature with RSA Encryption Algorithm to Enhance the Data Security of Cloud in Cloud Computing”, 2010 First International Conference On parallel, Distributed and Grid Computing (PDGC-2010).

[26] Gurpreet Singh, SupriyaKinger “Integrating AES, DES, and 3-DES Encryption Algorithm for Enhanced Data Security” International Journal of Scientific & Engineering Research, volume 4, Issue 7, July-2013.