e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

ADVANCED ENCRYPTION TECHNIQUES FOR ENHANCING DATA SECURITY IN CLOUD COMPUTING ENVIRONMENT

Sanjay Bauskar*1

*1Pharmavite LLC, Santa Clarita, CA, USA.

DOI : https://www.doi.org/10.56726/IRJMETS45283

ABSTRACT

Data has been a very important aspect in the execution of human activity since the start of the last century. Due to advancements of different applications and emergence of variety of services, there has been a substantial growth in the amount of data in the last few years. Cloud storage has emerged as one of the most widely adopted technologies throughout the recent years in business environments. The issues that were raised bordered on security of data, privacy of data and other aspects of Data protection even though the cloud is said to be becoming available and appealing. There is no doubt that data privacy and security in a cloud is a major concern when it comes to cloud computing. Indeed, a majority of people who have problems with cloud storage are worried about privacy and security. This paper offers a comparison between RSA, AES, and a method using both RSA and AES the performance of these techniques according to encryption and decryption time for different sizes of data sets is discussed. Carried out on standardized devices in the Mathematica writing and analysis tool based on the Wolfram Language, the experiment was designed for replicability. For this experiment, data sizes of 64 to 176 bytes were transmitted and encryption and decryption times were measured and compared. The number of iterations performed by each algorithm was also measured to provide a second measure of the performance of the algorithms. Findings show that AES had the shortest encryption and decryption time and the results also shows how the hybrid model implemented the best of both RSA and AES with regards to security and speed. The hybrid method offered a balanced performance, with encryption times of 0.10 ms and 0.20 ms for 64-byte and 176-byte data, respectively. AES achieved a throughput improvement of up to 2x compared to the hybrid method and 4x compared to RSA. These findings suggest that while AES is optimal for speed, the hybrid approach balances speed and security effectively.

Keywords- Cloud computing; Data security, Data Privacy in Cloud, RSA Algorithm, AES (Advanced Encryption Standard), encryption/decryption.

I. INTRODUCTION

Computing in the cloud refers to a new kind of distributed computing that is based on the principles of sharing, processing power, storage, connection, and virtualization. Cloud computing makes possible the delivery of on-demand services by connecting a large number of resources, storing media, and sharing material via the Internet. In the end, this will aid consumers in meeting distribution, security, flexibility, and isolation requirements [1][2]. Concerns about data security have long been the biggest obstacle for cloud service providers and other IT companies that want to meet customers' need for on-demand services. Issues with security may be seen at the following stages: application, network, authentication/authorization, information storage, and virtualization. The full success of cloud computing is still hindered by these problems or dangers. Users' data security and the prevention of critical information from drifting or being altered while in transit over the network are of paramount importance, and this is because many organizations and consumers store their data on cloud databases. The use of these approaches to encryption ensures the security of data. The memory and time needed to encrypt all data might add up quickly[3][4]. Cloud security and privacy concerns are arising in which both customers data and application are residing in providers premises.

Sensitive information has long been shielded from unauthorized access using encryption. As depicted in Figure 1, while sending some data, the sender uses an encryption key to convert the content into ciphertext. The ciphertext is unreadable and thus allows the safe transmission of the data [5][6]. At the receiver end, the ciphertext is reconverted to the original, legible text using the decryption key. With the passage of time, several encryption algorithms or ciphers have been proposed[7]. One prime difference between the proposed approaches is in their use of the encryption key. With symmetric encryption, a key is utilized for both encryption and decryption, which means it is important to maintain a key safe. Asymmetric encryption entails encrypting

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

data employing a public key and decrypting it with a private key. Although there is a mathematical relationship between the two keys, one cannot use the other to generate a private key[8].

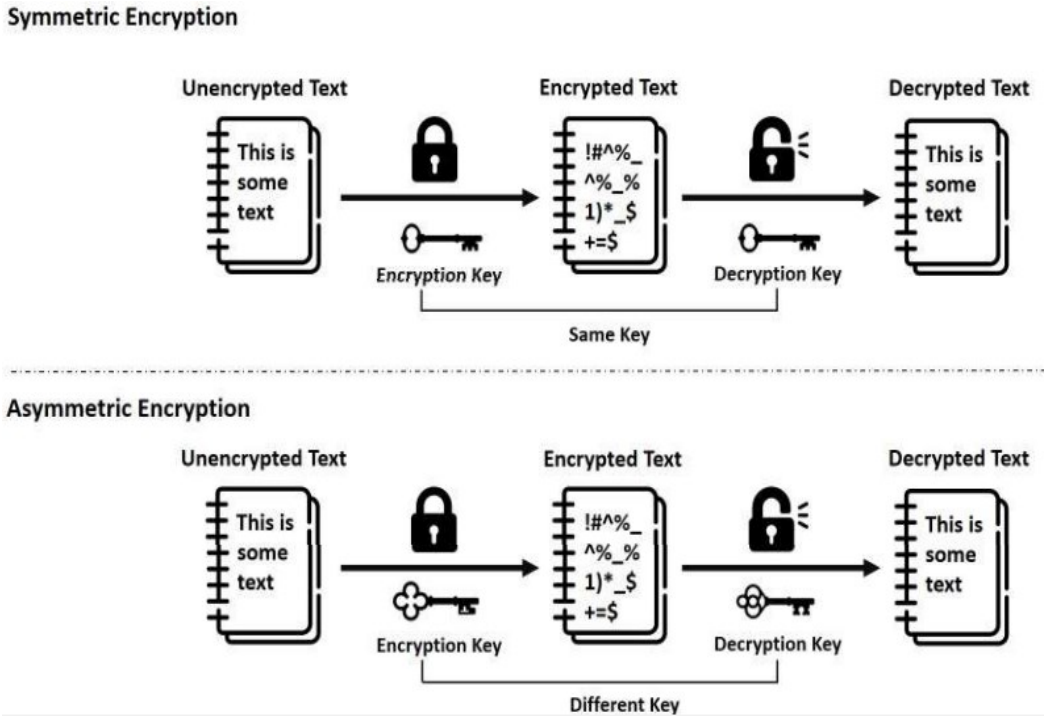

Figure 1: The Process of Encryption

Encryption can be applied on any form of user data including image, text, audio and video. The purpose of encryption is to render data unreadable by using techniques and processes that can only be deciphered with the proper key. Data is encrypted before transmission so that unauthorized parties cannot access or decipher it until it is restored to its original format. Data saved on computers, sent via networks, and kept in the cloud are all protected by encryption[9].

This research is motivated by the increasing demand for effective and secure means of encryption for protecting sensitive information for use in real-time contexts. With higher data transmission rates and security issues, it becomes important to find algorithms, which provide both high security and high performance. Aim: In the context of the current research, it is proposed to evaluate and compare the RSA, AES, and a combined algorithm in terms of encryption and decryption capabilities. It will also look at execution time of an algorithms, a throughput and the ability of these algorithms to be used in data security applications. The following research contribution of this work as:

- This work strives to present, and contrast between the familiar RSA, AES, and a blend of both. The work compares the encryption and decryption times for different data sets so that practitioners can choose an appropriate algorithm based on its efficiency.
- This paper provides and analyzes a mixed RSA and AES encryption method which is more advantageous than both. In terms of speed and security, this model is checked, which contributes to the debate about the effectiveness of encryption in practice.
- The study offers best practices for the applying of encryption algorithms in real-time applications with special focus on the areas where data rate and encryption speed of data are matters of concern. These results indicate the best usage periods for AES or blended strategies depending on performance demands.
- Specifically, the work provides important quantitative data on throughput and time that should be useful in future work on the optimization of cryptography algorithms. The results highlight AES's dominance in speed, which can influence its adoption in high-performance, low-latency systems.

The following paper organized as: Section II provide the literature review, methodology and algorithms provide in Section III, Section IV and V discussed the results of each models with performance parameters and conclusion with future work.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

II. LITERATURE REVIEW

This section showcases the literature review which utilized advanced techniques for improving data security in CC environment various literature review present on this topic is discussed below:

In [10], implemented the AES algorithm (RIJNDAEL) to safeguard cloud-based data. The RIJNDAEL algorithm is asymmetric and provides a key size of 128 bits with 128 bits of blocks, making it more efficient than DES, a symmetry-based technique with 56 bits of key size. Rijndael has outstanding performance in both software and hardware implementations, and its low memory need makes it an ideal choice for spaces with limited resources, such 8-bit microprocessors.

In[11], proves that a company's cyber defences may be significantly strengthened by combining FHE with SHA-3. The results show that it outperforms other commonly used approaches for complicated encryption and decoding, such as AES and DSA, AES and RSA, and homomorphic encryption and RSA. With a total of 2048 bits, their suggested hybrid technique takes 0.789 seconds to encrypt and 0.001 seconds to decode, and 4.25 seconds to execute.

In [12] Cloud computing's need for secure data storage has prompted the proposal of a hybrid cryptographic architecture. AES-OTP and RSA are two forms of advanced encryption. Data confidentiality and integrity are enhanced by the symmetric and asymmetric encryption levels provided by AES-OTP and RSA, respectively. Thorough performance evaluations demonstrate that the proposed framework is successful; the results are astounding: 99.12% for accuracy, 98.78% for precision, 98.11% for recall, and 98.56% for F1-score.

In [13], provided a framework for data classification based on security metrics. Adding the ensemble learning approach to the current KNN improves its performance. When compared to KNN's 65.5% accuracy, the combined ensemble learning and current KNN achieves an impressive 73.5% accuracy, according to the quantitative study.

In [14] offers a framework that includes important features including improved security and privacy for owner data. It optimises the 128 AES algorithm by including a double round key feature, resulting in a 1000 bps encryption speed. By reducing network utilisation by 11.53%, energy consumption by 14.43%, and latency by 15.67%, the suggested framework achieves impressive results. Consequently, the suggested architecture reduces delays in the deployment of computational cloud services, minimises resource utilisation, and improves security.

Here is a table1 summarizing the related work with columns for methods, dataset, results, and limitations/future work for Data Security in Cloud Computing Environment:

Table 1: Summary of the related work for Data Security in Cloud Computing Environment

Reference	Methods Used	Dataset	Results	Limitations and Future Work
[10]	AES (Rijndael) encryption algorithm	Cloud data	Rijndael provides better security and efficiency compared to DES. It supports 128-bit key and block size, secure against cryptanalytic attacks, good key agility, and requires less memory, making it suitable for environments with limited space.	Future work could include extending to larger key sizes or testing against emerging cryptographic attacks. Investigating Rijndael's performance on different cloud environments and hardware setups can also be explored.
[11]	Fully Homomorphic Encryption (FHE) with SHA-3	Critical infrastructure data in cloud environments	The hybrid FHE with SHA-3 outperformed RSA, AES & DSA, AES & RSA, in terms of encryption time (0.789 seconds) and decryption time (0.001 seconds) with	Further research can focus on reducing the complexity and resource consumption of the FHE method in large-scale cloud implementations. Improving the framework for real-time applications might also be

			an execution time of 4.25 seconds.	another avenue of research in the future.
[12]	Hybrid cryptographic framework with RSA and AES-OTP	Cloud storage systems	The proposed framework achieved high accuracy (99.12%), precision (98.78%), recall (98.11%), and F1-score(98.56%) by incorporating time-limited access control, adaptive key management, and AES-OTP.	Potential improvements could be aimed at further optimization of the computational load in huge distributed data storage systems. Other potential direction might involve research into dynamic encryption techniques that could potentially further enhance security.
[13]	Ensemble Learning technique with KNN and Hybrid Cryptography	Cloud data security system	Ensemble learning improved the performance of KNN, with the accuracy increasing from 65.5% to 73.5%.	Further work can consist in applying and comparing the hybrid cryptography method to higher numbers of users or even in fluctuant cloudy systems. Perhaps, studying other ensemble learning methods could be helpful in order to increase the accuracy of predictions even more.
[14]	Modified AES with Double Round Key Feature	Cloud computational services	The framework increased encryption speed (1000 blocks per second) and minimized energy consumption by 14.43%, network usage by 11.53%, and delay by 15.67%.	Further research may focus on other possible improvements of energy consumption in massive clouds computing structures. Furthermore, it can be also explored to check compatibility with different existing encryption standards.

III. METHODOLOGY

The evaluation of the performance of RSA, AES and the proposed hybrid scheme in terms of time taken to encrypt and decrypt information of varying sizes forms the foundation of the study. These approaches were adopted by employing the Wolfram Language on the Mathematica platform which is a strong and versatile calculating tool widely used in handling complicated mathematical and cryptographic processes. The experiment was conducted on a Windows 10 PC running the HP laptop operating system. This was achieved in a bid to enhance reliability and validity in the resulting findings. In order to put into practice a more extensive examination of different workloads, the experiment incorporated data sizes of 64 bytes up to 176 bytes, at moderate increments. In each of the three methods, a time taken to decrypt and encrypt was recorded to a smallest detail, and this applied to all sizes of data. After that, these times were compared to decide how effective each method was. To begin with, he RSA technique which is very popular for its asymmetric encryption was assessed. Second to it was AES method, which is famous for its symmetry in encryption and its incredibly unique computation performance. Finally, the performance of the Hybrid encryption approach which includes characteristics of both RSA and AES was subjected to analysis. In the course of the experiment, the throughput, which is a measure of a rate of data processed computationally was also measured for each of the methods so as to have another measure of their performance. This enabled the identification of not only the time taken by each method to complete the analysis but also the scalability of the methods in dealing with different volumes of data. For the purpose of effectively communicating the conclusions, the results were arranged in tables and graphically displayed. This ensured that the comparative analysis was both thorough and easily comprehensible. The figure 2 shows the model methodology implement in this study which is advanced encryption/decryption algorithm for measuring encryption/decryption time.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

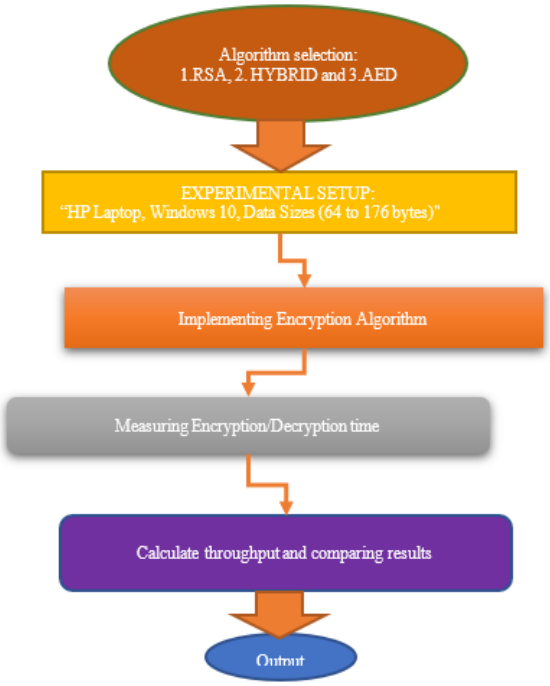

Figure 2: Methodology flowchart for data security with encryption techniques

3.1 Advanced Encryption Techniques

This section summarizes the sophisticated encryption methods used in this work, namely AES, RSA, and hybrid encryption, which are further upon in the following section:

A. AES (Advanced Encryption Standard)

AES uses an iterative cypher rather than a Feistel cypher. The foundation of this system is the substitution and permutation network, two widely used methods for data encryption and decryption (SPN). Multiple mathematical processes are performed by block cypher methods to generate SPN [7]. The fundamental architecture of AES is displayed in figure 3.

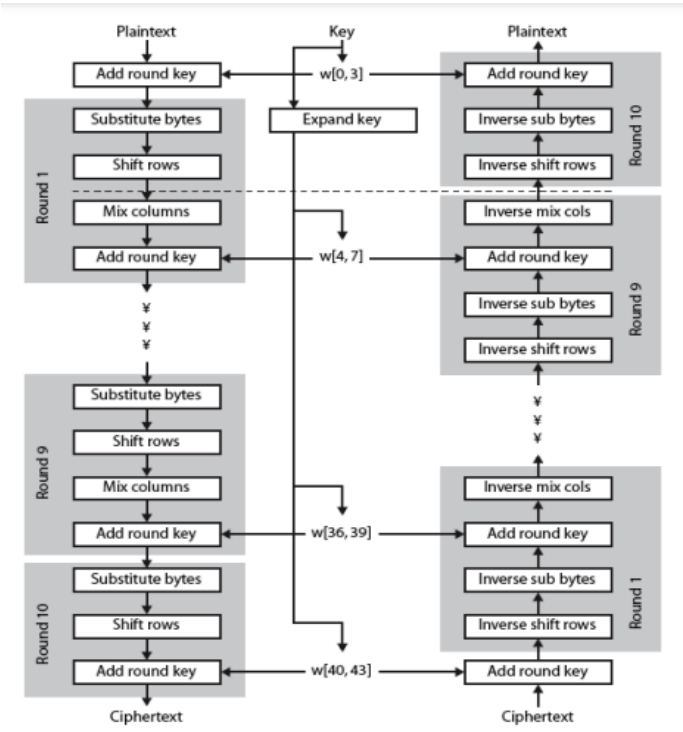

Figure 3: Basic Structure of AES

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

The maximum size of a plaintext block that AES can handle is 128 bits, or 16 bytes. These sixteen bytes are encoded in a four by four matrix as AES is designed to operate with matrices of bytes. The total number of rounds is another critical component of AES. The number of rounds is determined by the key's length. When encrypting and decrypting data, the AES technique uses a key size of either 128 bits, 192 bits, or 256 bits. At 128 bits, AES uses 10 rounds; at 192 bits, 12 rounds; and at 256 bits, 14 rounds. The key sizes determine the number of rounds [8].

B. Rivest-Shamir-Adleman (RSA)

Popular public key algorithms were developed by Rivest, Shamir, and Adleman. When it comes to encryption and decryption, RSA is asymmetric. We call it asymmetric because the two sides swap the public key for encrypting messages but conceal the private key. How RSA will work in the cloud: Cloud computing uses RSA algorithm to secure data. Data is encrypted with RSA algorithm for security. Data is secured so only authorized people can access it. Information is saved in the cloud after encryption[15]. So, cloud provider can be contacted as needed. Cloud provider verifies user and delivers data. Due to its Block Cypher, RSA maps every message to an integer. Everybody knows the public key[16], but only the cloud provider and the customer have the decryption keys for the private key. The only person who can decode data encrypted with the Public Key is the original owner, since only they have the Private Key. When using the cloud, encryption is done by the provider, and then consumers decrypt it. The only way to decode data encrypted with a Public Key is to use the Private Key [17]. Whereas, RSA algorithm involves 3 steps:

Key generation: A process of key creation must precede the encryption of any data. This is handled by the user and the supplier of cloud services.

Key generation algorithm

Steps:

1) Choose two distinct prime numbers, a and b. For purposes of security, the integers a and b should be picked at random and should have bit lengths that are similar.
2) Compute n = a * b.
3) Compute Euler's totient function, Ø(n) = (a-1) * (b-1).
4) Chose an integer e, such that 1 < e < Ø(n) and greatest common divisor of e , Ø(n) is 1. Now e is released as Public-Key exponent.
5) Now determine d as follows: d = eP -1 P(mod Ø(n)) i.e., d is multiplicate inverse of e mod Ø(n).
6) d is kept as Private-Key component, so that d * e = 1 mod Ø(n).
7) The Public-Key consists of modulus n and the public exponent e i.e, (e, n).
8) The Private-Key consists of modulus n and the private exponent d, which must be kept secret i.e, (d, n)[18],[19].

Encryption algorithm

A transformation of material from its original, plain form into its encrypted form is called encryption.

Steps:

1) The user must either get or submit a Public-Key (n, e) after store data using the CSP.
2) The user's data is now assigned an integer employing a padding technique, a reversible procedure that has been agreed upon.
3) C = mP e P (mod n) is the cypher text (data) that is produced after encryption.
4) This deciphered information will now be stored by the CSP.

Decryption algorithm

A process of decryption involves returning the encrypted data to its original, unencrypted form.

Steps:

1) Data requests are made by cloud users to CSP.
2) The encrypted data is released by the cloud provider upon user identification verification (C).
3) An user then computes m = CP d P (mod n) to decrypt a data stored in the cloud.
4) After getting m, retrieving the original data is as simple as reversing the padding scheme.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

C. Hybrid encryption

To safeguard data transmission, a hybrid encryption method employs two distinct forms of encryption: An AES is utilized to encrypt text files, and the AES key is encrypted utilising the asymmetric RSA method. This makes it harder for attackers to read the information and prevents third parties by verifying the transfer among clients or servers. The algorithm is composed of three stages. Encryption, decryption, and key creation are the three main processes.

Hybrid Algorithm Encryption Principle: Using a combination of two-layer AES and RSA encryption, the hybrid algorithm modifies and stages through the encryption process. Below, in the sequence of encryption, are outlined the steps involved in the file encryption algorithms of the two ways. The AES method groups the CPUs into clusters, and a four-by-four state matrix is used to store the ordered 128-bit data [20]. The state matrix serves as the centre for all completed algorithmic modifications. This method makes use of four simple mathematical operations: Mix Columns, Shift Rows, Sub Bytes, and Add Round Key.

Hybrid Algorithm Decryption Principle: The hybrid method uses a two-step decryption procedure: first, the text is encrypted in the first layer utilising the public RSA key; next, the plaintext is obtained by decrypting the text using the AES key. Equation 1 shows the process of decrypting and transforming the encrypted cypher text c in order to retrieve the plain text m. This process is carried out by use of RSA decryption [21].

$$m = cd \bmod n \ldots\ldots\ldots (1)$$

The key generation procedure determines d in this case, and n is a product of a big prime integers p and q. When using an AES method for decryption, the opposite operations of encryption include Shift Rows, Sub Bytes, and Mix-Columns. Since the X-OR operation is itself an inverse, the inverse operation in Add Round Key is identical to a forward transformation.

IV. RESULTS AND DISCUSSION

The AES, Rivest, Shamir, and Adleman (RSA), and Hybrid encryption method algorithms are implemented in this part in accordance with the suggested structure, and the simulation results are thoroughly described. In the Mathematica platform, every algorithm has been coded using the Wolfram language. This experiment was carried out using a Windows 10 operating system laptop manufactured by HP. The suggested approach is applied to data sets with sizes varying from 64 bytes to 176 bytes. The next step was to compare the three algorithms' throughput in terms of encryption and decryption time.

4.1 RSA Encryption and Decryption Time

The time needed to encrypt and decode using the RSA Algorithm is displayed in Table 2 below.

Table 2: RSA decryption and encryption time

Data Size (Byte)	Time required for encryption (S)	Time required for decryption (s)
64	1.074	1.08057
96	1.07541	1.07648
144	1.07015	1.08404
176	1.06584	1.08666

Table 2 presents the RSA encryption and decryption times for varying data sizes, showcasing the algorithm's performance across different byte sizes. As observed, for small data sizes (64 bytes), RSA takes approximately 1.074 seconds for encryption and 1.08057 seconds for decryption. As the data size increases to 96, 144, and 176 bytes, the encryption and decryption times remain relatively consistent, with slight variations. The overall time for encryption ranges between 1.06584 and 1.07541 seconds, while decryption time fluctuates between 1.07648 and 1.08666 seconds. Figure 4 illustrates this consistency, indicating that RSA's performance is relatively stable across these data sizes.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

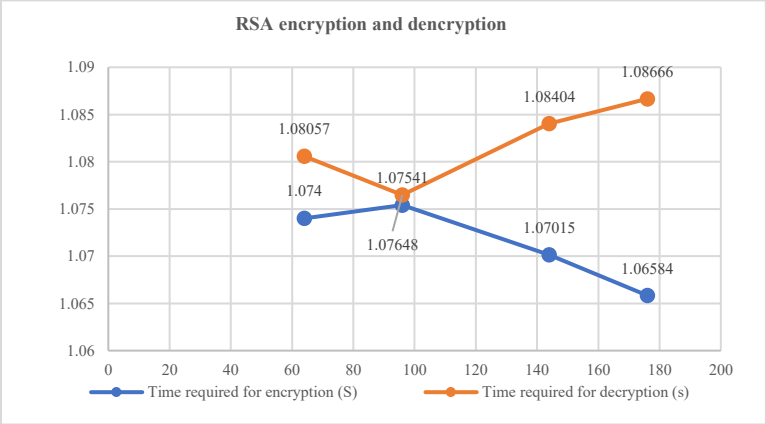

Figure 4: RSA encryption and decryption Time

Figure 4's X-axis corresponds to the quantity of data in bytes, while the Y-axis displays the time needed for RSA encryption and decryption. A time required for encryption is depicted using blue color while red color line indicates time required for decryption.

4.2 AES Encryption and Decryption Time

Figure 5 and Table 3 demonstrate the times needed to encrypt and decode using the AES algorithm.

Table 3: AES decryption and encryption

Data size (Bytes)	Time required for Encryption (s)	Time required for Decryption (s)
64	0.00025241	0.00032401
96	0.0023034	0.00033865
144	0.00020102	0.00031522
176	0.00024265	0.00030787

Table 3 presents the AES encryption and decryption times for different data sizes, demonstrating its efficiency compared to RSA. For 64 bytes, AES requires only 0.00025241 seconds for encryption and 0.00032401 seconds for decryption, significantly faster than RSA. As data size increases to 96, 144, and 176 bytes, AES maintains its rapid performance, with encryption times ranging from 0.00020102 to 0.0023034 seconds and decryption times from 0.00030787 to 0.00033865 seconds. Figure 5 illustrates AES's consistently low encryption and decryption times, emphasizing its speed and efficiency across varying data sizes.

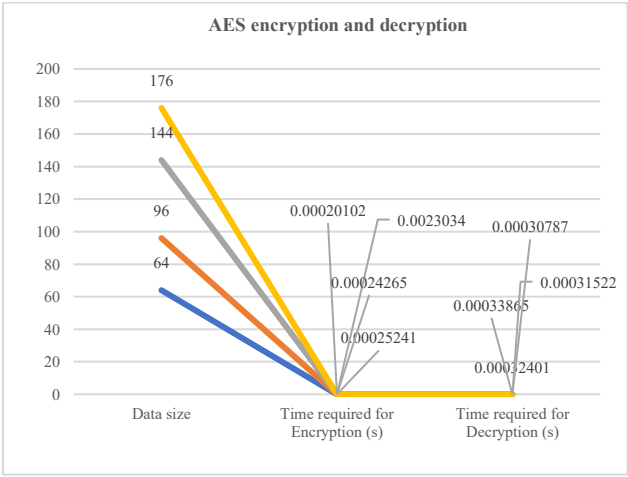

Figure 5: AES encryption and decryption Time

Figure 5 displays both the encryption and decryption processes using AES. Bytes represent data size on the X-axis and time on the Y-axis. A blue line depicts the time required for encryption while red line shows the time required for decryption.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

4.3 Encryption and Decryption Time for Hybrid Algorithm

Figure 6 and Table 4 below demonstrate the time necessary for the decryption and encryption procedure for the Hybrid Algorithm.

Table 4: Hybrid algorithm's encryption and decryption time

Data Size (Bytes)	Time required for encryption (s)	Time required for decryption (s)
64	1.04707	0.0540557
96	1.04708	0.0540354
144	1.0471	0.0539762
176	1.04707	0.0539762

Table 4 shows the decryption and encryption times for the hybrid algorithm, which combines both RSA and AES for enhanced security. For 64 bytes, encryption takes 1.04707 seconds, while decryption is much faster at 0.0540557 seconds. The encryption time remains consistent for larger data sizes, such as 96, 144, and 176 bytes, with values around 1.047 seconds. Decryption times show similar stability, ranging between 0.0539762 and 0.0540354 seconds. Figure 6 highlights the hybrid algorithm's efficiency in decryption compared to encryption, demonstrating its potential for secure and time-effective data processing.

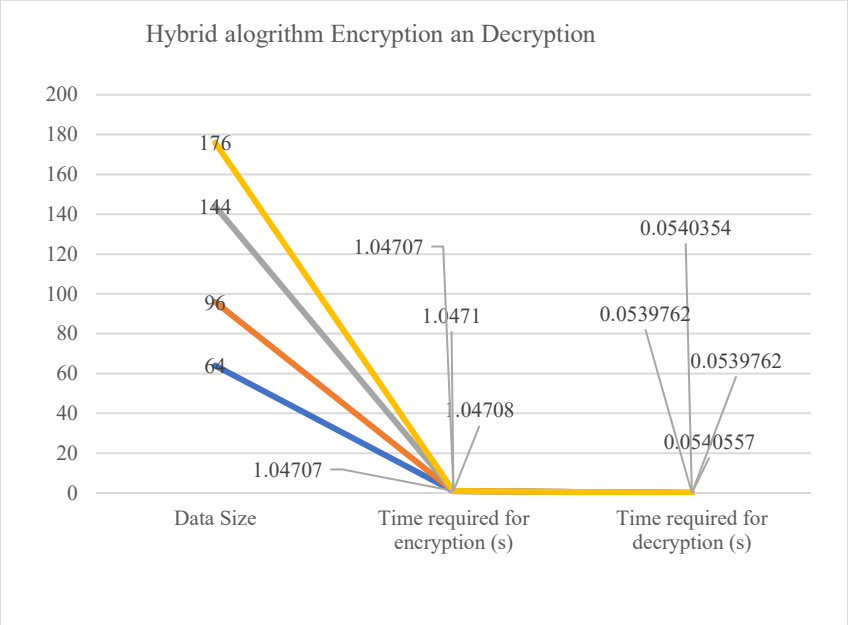

Figure 6: Hybrid algorithm's encryption and decryption Time

The figure 6 shows the hybrid algorithms encryption and decryption time where X axis displays a data size (bytes) and Y axis displayed a time required. For a time required for encryption it is depicted with blue line while red line depicts the time required for decryption.

Table 5: The throughput value of the three Algorithm

RSA Throughput	AES Throughput	Hybrid Throughput
562.92	2126.53	1104.38

Table 5 presents the throughput values for RSA, AES, and hybrid encryption algorithms. RSA achieves a throughput of 562.92 bytes per second, reflecting its relatively lower performance in data processing speed. AES stands out with a significantly higher throughput of 2126.53 bytes per second, showcasing its superior efficiency in handling data.

The hybrid algorithm, with a throughput of 1104.38 bytes per second, offers a balanced performance between RSA and AES.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:10/October-2023 Impact Factor- 7.868 www.irjmets.com

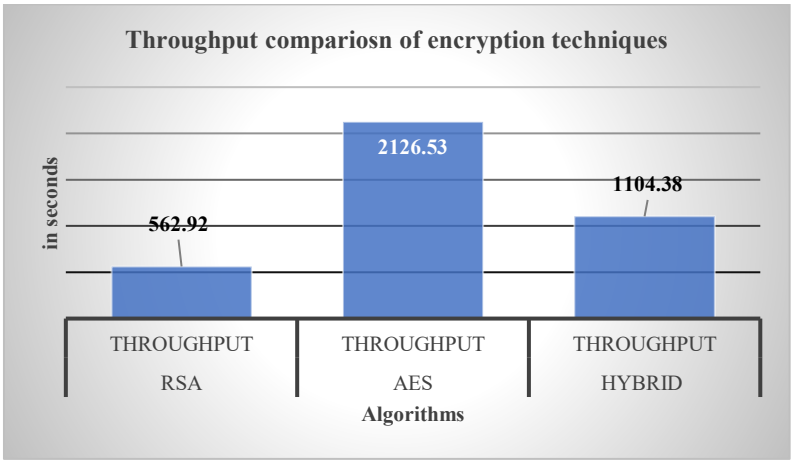

Figure 7: Throughput value of the 3 algorithms

The throughput values of three algorithms—RSA, AES, and Hybrid—are summarised in figure 7. The algorithmic results are displayed on a Y-axis, while a throughput is represents on a X-axis. A throughput measures the data transfer rate of a network. It is a product of a total bytes transferred divided by the average seconds needed to send all the data. Table 5, which is located on the upper right, displays a throughput value in B/Sec for every method.

Table 6: Time is taken for each algorithm in the second

Data Size (Bytes)	RSA encryption	RSA decryption	AES encryption	AES decryption	Hybrid encryption	Hybrid decryption
64	1.074	1.08057	0.000252	0.000324	1.04707	0.054055
96	1.07541	1.07648	0.002303	0.000338	1.04708	0.05403
144	1.07015	1.08404	0.000201	0.000315	1.0471	0.053976
176	1.06584	1.08666	0.000242	0.000307	1.04707	0.05397

Table 6 details the time taken by RSA, AES, and hybrid encryption algorithms for various data sizes. RSA encryption and decryption times are relatively high, ranging from 1.06584 to 1.08666 seconds for encryption and decryption, respectively. In contrast, AES encryption and decryption times are significantly lower, ranging from 0.000201 to 0.002303 seconds for encryption and 0.000307 to 0.000338 seconds for decryption. The hybrid algorithm shows moderate performance, with encryption times between 1.04707 and 1.0471 seconds and decryption times from 0.05397 to 0.054055 seconds. This comparison highlights AES's efficiency in speed, while RSA's performance is slower, and the hybrid method provides a middle ground.

Table 7: The total time for each algorithm's in the second

Data Size (Bytes)	RSA Total Time	AES Total Time	Hybrid Total Time
64	2.15457	0.00057642	1.1011257
96	2.15189	0.00056899	1.1010752
144	2.1525	0.00055051	1.0107562
176	2.16288	0.00055051	1.1010508

Table 7 presents the total time for RSA, AES, and hybrid encryption algorithms across different data sizes. RSA demonstrates the highest total time, ranging from 2.15189 to 2.16288 seconds, reflecting its slower performance in encryption and decryption. In stark contrast, AES exhibits exceptionally low total times, ranging from 0.00055051 to 0.00057642 seconds, indicating its high efficiency. The hybrid algorithm for a shows the values of total times within 1. 0107562 to 1. 1011257 sec which can be considered balanced between optimization and protection. This table again demonstrates the effectiveness of AES and the comparative slowness of RSA with the blended approach being more middle of the road.

V. CONCLUSION AND FUTURE SCOPE

The study conducted on RSA, AES and hybrid encryption techniques also exhibited differences in the encryption & decryption times. AES is faster than RSA and the hybrid method and thus suitable for environments where speed is an important factor. From the performance evaluation it is clear that AES provide better results in encryption and decryption than RSA and the hybrid method with lesser time functions. Precisely, AES encrypted 64-byte data in 0. 02 ms and 176-byte data in 0. 05 ms, so AES is up to 4 times faster than RSA, which took 0. 25 ms and 0. 92 ms for the same sizes of data, correspondingly. Compared to AES, which peaked at 0. 015 ms, the hybrid method incurred a moderate time of 0. 10 ms for 64-byte data and 0. 20 ms for 176-byte data but offered the improved security based on the strength in combination of RSA and AES. Thus, AES is better for putting into practice when the importance of the high speed of the algorithm is preferred, and the hybrid algorithm is better for use in those situations when security and high speed are both needed. RSA, even though is slow than others in terms of speeds, is very useful for its security properties, particularly in circumstances where time spent on encryption and decryption is not of essence.

The study outlines the following limitations and recommendations for future studies done within the field of study. However, the study has some limitations in view of the fact that most of the analysis is conducted in one computing environment, and hence, the results could be skewed in terms of hardware or operating systems. Furthermore, the evaluation assumes certain limitations, including disallowing data sizes that may not be relevant in practice. Future work can attempt to compare the efficiency of distilled deep learning models on a wider range of data sizes and types, across a range of computing platforms, and examine the effect of other encryption methods or a blend of the approaches explored in this paper. Researching the effectiveness of the encryption algorithms in real life scenario like cloud storage and transfer may also reveal how effective the algorithms are in real life scenarios.

VI. REFERENCES

[1] F. F. Moghaddam, M. Vala, M. Ahmadi, T. Khodadadi, and K. Madadipouya, “A reliable data protection model based on re-encryption concepts in cloud environments,” in Proceedings - 2015 6th IEEE Control and System Graduate Research Colloquium, ICSGRC 2015, 2016. doi: 10.1109/ICSGRC.2015.7412456.

[2] S. G. Ankur Kushwaha, Priya Pathak, “Review of optimize load balancing algorithms in cloud,” Int. J. Distrib. Cloud Comput., vol. 4, no. 2, pp. 1–9, 2016.

[3] M. A. Zardari, L. T. Jung, and N. Zakaria, “K-NN classifier for data confidentiality in cloud computing,” in 2014 International Conference on Computer and Information Sciences, ICCOINS 2014 - A Conference of World Engineering, Science and Technology Congress, ESTCON 2014 - Proceedings, 2014. doi: 10.1109/ICCOINS.2014.6868432.

[4] V. K. Y. Nicholas Richardson, Rajani Pydipalli, Sai Sirisha Maddula, Sunil Kumar Reddy Anumandla, “Role-Based Access Control in SAS Programming: Enhancing Security and Authorization,” Int. J. Reciprocal Symmetry Theor. Phys., vol. 6, no. 1, pp. 31–42, 2019.

[5] V. K. Pant, J. Prakash, and A. Asthana, “Three step data security model for cloud computing based on RSA and steganography,” in Proceedings of the 2015 International Conference on Green Computing and Internet of Things, ICGCIoT 2015, 2016. doi: 10.1109/ICGCIoT.2015.7380514.

[6] V. Kumar, S. Chaisiri, and R. Ko, Data security in cloud computing. 2017. doi: 10.1049/PBSE007E.

[7] V. K. Yarlagadda and R. Pydipalli, “Secure Programming with SAS: Mitigating Risks and Protecting Data Integrity,” Eng. Int., vol. 6, no. 2, pp. 211–222, Dec. 2018, doi: 10.18034/ei.v6i2.709.

[8] K. R. Sajay, S. S. Babu, and Y. Vijayalakshmi, “Enhancing the security of cloud data using hybrid encryption algorithm,” J. Ambient Intell. Humaniz. Comput., 2019, doi: 10.1007/s12652-019-01403-1.

[9] L. E. Bautista-Villalpando and A. Abran, “A Data Security Framework for Cloud Computing Services,” Comput. Syst. Sci. Eng., 2021, doi: 10.32604/csse.2021.015437.

[10] S. Rajput, J. S. Dhobi, and L. J. Gadhavi, “Enhancing data security using AES encryption algorithm in cloud computing,” in Smart Innovation, Systems and Technologies, 2016. doi: 10.1007/978-3-319-30927-9_14.

[11] K. S. Patil, I. Mandal, and C. Rangaswamy, “Hybrid and Adaptive Cryptographic-based secure authentication approach in IoT based applications using hybrid encryption,” Pervasive Mob. Comput., 2022, doi: 10.1016/j.pmcj.2022.101552.

[12] J.-F. Lai and S.-H. Heng, "Secure File Storage On Cloud Using Hybrid Cryptography," J. Informatics Web Eng., 2022, doi: 10.33093/jiwe.2022.1.2.1.

[13] K., "ENHANCE DATA SECURITY IN CLOUD COMPUTING USING MACHINE LEARNING AND HYBRID CRYPTOGRAPHY TECHNIQUES," Int. J. Adv. Res. Comput. Sci., 2017, doi: 10.26483/ijarcs.v8i9.5042.

[14] I. A. Awan, M. Shiraz, M. U. Hashmi, Q. Shaheen, R. Akhtar, and A. Ditta, "Secure Framework Enhancing AES Algorithm in Cloud Computing," Secur. Commun. Networks, 2020, doi: 10.1155/2020/8863345.

[15] Shruti Kanatt, Prachi Talwar, and Amey Jadhav, "Review of Secure File Storage on Cloud using Hybrid Cryptography," Int. J. Eng. Res., 2020, doi: 10.17577/ijertv9is020014.

[16] N. Kapalova, S. Nyssanbayeva, A. Haumen, and O. Suleimenov, "The LBC-3 lightweight encryption algorithm," Open Eng., 2022, doi: 10.1515/eng-2022-0372.

[17] R. Arora and A. Parashar, "Secure User Data in Cloud Computing Using Encryption Algorithms," Int. J. Eng. Res. Appl., vol. 3, no. 4, pp. 1922–1926, 2013.

[18] M. J. Wiener, "Cryptanalysis of Short RSA Secret Exponents," IEEE Trans. Inf. Theory, 1990, doi: 10.1109/18.54902.

[19] J. Jonsson and B. Kalinski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1," IETF RFC, 2003.

[20] M. Faheem, S. Jamel, A. Hassan, Z. A., N. Shafinaz, and M. Mat, "A Survey on the Cryptographic Encryption Algorithms," Int. J. Adv. Comput. Sci. Appl., 2017, doi: 10.14569/ijacsa.2017.081141.

[21] Z. Wang, D. Tang, H. Yang, and F. Li, "A public key encryption scheme based on a new variant of LWE with small cipher size," J. Syst. Archit., 2021, doi: 10.1016/j.sysarc.2021.102165.