Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 4 (2023)

Design a Hybrid Algorithm for Data Encryption to Implementation in Database

Satinder[1] , Dr. Parveen Sehgal[2]

** Research Scholar, Deptt. of Computer Science & Engineering , School of Engineering & Technology, Om Sterling Global University, Hisar, Haryana, India*

*** Professor, Deptt. of Computer Science & Engineering , School of Engineering & Technology, Om Sterling Global University, Hisar, Haryana, India*

Abstract

In an era where digital data is increasingly prevalent, data encryption has become essential to information security. Because databases usually contain sensitive and significant data and are a popular target for cyberattacks, robust encryption solutions are necessary. This research proposes a novel hybrid encryption method that combines symmetric and asymmetric encryption approaches to safeguard databases. The recommended approach combines the security of asymmetric encryption with the speed of symmetric encryption to provide a dependable and efficient data protection solution. This paper introduces hybrid techniques by combining the two most essential algorithms AES and RSA algorithms with XORed Operation. This hybrid encryption algorithm provides more security as compared to existing hybrid algorithms. The implementation and result are also derived in the paper.

Keywords**:** *Hybrid Cryptography, Database Security, AES, RSA,*

Introduction

Data security is essential in today's global digital world, particularly database management. Organizations across various sectors use databases to store and manage sensitive information, including personal and financial data and exclusive company insights. As the amount and value of this data expand, robust encryption techniques will be necessary to safeguard it against unauthorized access and potential breaches. Conventional symmetric encryption provides fast and efficient data security, as evidenced by methods such as the Advanced Encryption Standard (AES). It poses a severe problem in terms of securely communicating encryption keys. However, when encrypting vast volumes of data, asymmetric encryption—represented by algorithms like the Rivest-Shamir-Adleman (RSA) method—effectively overcomes the key exchange problem at the sacrifice of efficiency.

A hybrid encryption scheme is a promising method for balancing efficacy and security. The benefits of both symmetric and asymmetric encryption are combined in this paradigm to create a strong foundation for secure data management. This paper provides a brand-new hybrid data encryption technique designed primarily for database applications. It seeks to provide the best of both worlds by fusing rapid, effective data encryption with AES and safe key exchange with RSA. Our proposed method, the "Hybrid Encryption Algorithm," aims to protect confidential information stored in databases while ensuring that encryption and decryption processes are completed promptly. Using the safe key distribution of RSA and the speed of AES with XOR operation for data encryption, we want to offer a comprehensive encryption framework tailored to the unique needs of modern database systems.

This paper describes the Hybrid Encryption Algorithm, including key generation, the encryption process, working principles, block diagram, and key management techniques. We tackle the difficulties of safely keeping

encryption keys, minimizing possible weaknesses and enhancing efficiency to ensure the method is appropriate for practical database uses.

In addition, we provide a comprehensive analysis of the Hybrid Encryption Algorithm, examining its performance features, security aspects, and efficacy in protecting data. We compare our proposed hybrid approach with existing widely used encryption strategies in database security to show off its benefits and emphasize how it could improve database security across a range of sectors.

Hybrid Cryptography

Asymmetric key encryption, sometimes referred to as public key encryption, and symmetric key encryption are the two encryption techniques that are combined in hybrid cryptography. This combination makes use of both approaches' advantages to offer a safe and effective means of encrypting and decrypting data.

Working of Hybrid cryptography

1. Symmetric-Key Encryption: In symmetric-key encryption, encryption and decryption are accomplished using the same shared secret key. When encrypting huge volumes of data, this approach works faster and more effectively than asymmetric-key encryption. The difficulty with symmetric encryption, though, lies in safely transferring the secret key between the parties involved in communication.

2. Asymmetric-Key Encryption: Asymmetric-key encryption uses a public key and a private key, which are linked but separate keys. While the private key is kept confidential, the public key is made available to everyone. Only the matching private key can be used to decrypt messages that have been encrypted using the public key. Since asymmetric encryption enables secure communication without requiring the transmission of secret keys, it is frequently used for digital signatures and key exchange.

The following actions are commonly taken in a hybrid cryptography system:

- **Key Exchange:** The communicating parties safely exchange a symmetric key through the use of asymmetric encryption. One party may, for instance, create a random symmetric key, encrypt it using the recipient's public key, and then give it to them. The only person who can decrypt and obtain the symmetric key is the recipient, who possesses the matching private key.
- **Data Encryption:** After the symmetric key has been shared, the data itself is encrypted. Since symmetric encryption is more effective in this context, the majority of data is encrypted using it.
- **Transmission of Data:** Through the communication channel, the encrypted data is transferred.
- **Data Decryption:** To decrypt the data, the recipient needs the symmetric key, which is only known to them.

Some Basic Algorithms used in Hybrid Cryptography.

AES Algorithm

The symmetric key encryption method used as the standard algorithm for sophisticated data encryption is known as AES (Advanced Encryption Standard). In addition to the DES method, a “block encryption” algorithm called AES is used. Although its block length is only limited to 128 bits, the AES method uses keys with lengths ranging from 128 bits to 192 bits to 256 bits. The AES algorithm's grouping and encryption steps, which both employ the same key, are shown in Fig. 1.

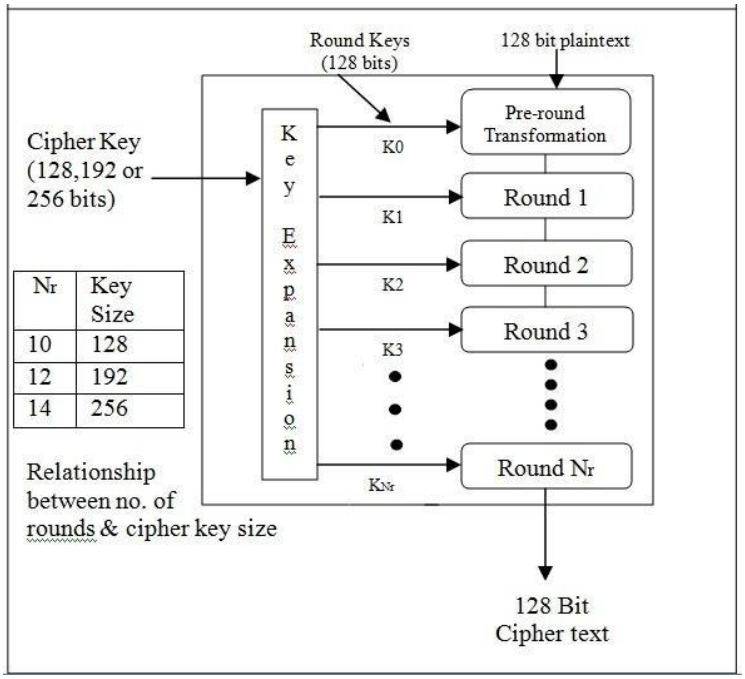

Fig. 1 AES Block Diagram

"AES algorithm" results are astounding. Execution-Speed-wise, the "AES" method is rather straightforward. It has a faster execution speed and inherits the speed of other encryptions such as DES, 3DES etc. It works well to encrypt and decrypt huge volumes of data and has great encryption efficiency. By expanding the key's length from 56 bits to 128,192 and 256 bits, the AES method addresses two issues: resource use and the insufficient length of the DES key. When compared to DES and 3DES, the AES algorithm's security has improved, and while it is still not as high as the RSA algorithm, it is still significantly less.(Satinder, 2023)

RSA Algorithm

A 1978 invention by "Ronald Rivest, Adi Shamir, and Leonard Adleman" that is a de facto industry standard for public key encryption. RSA served as the foundation for several encryption schemes. RSA is a public key encryption algorithm. One of the earliest significant developments in public key encryption, it was the first method that was recognised to be acceptable for both signing and encrypting. There are three steps to it:

- Key Generation,
- Encryption
- Decryption

Step-1 Key Generation Process:

- Select two distinct prime numbers, p and q.
- Process the modulus, n, as the product of p and q (n = p * q).
- Calculate the totient (Euler's totient function), φ(n), as (p - 1) * (q - 1).
- Select an encryption exponent, e, where 1 < e < φ(n), and e is relatively prime to φ(n)
- Calculate the decryption exponent, d, such that (d * e) % φ(n) = 1. This can be generated by "Extended Euclidean Algorithm".
- The "public key" is (n, e), and the private key is (n, d).

Step-2: Encryption Process:

- Convert the plaintext message into a numerical representation (usually using ASCII or Unicode codes).
- Break the plaintext into smaller blocks, if necessary, such that each block is less than or equal to n.

- Each plaintext-block, compute the ciphertext-block using the formula, ciphertext = (plaintext ^ e) % n.
- The encoded message is created from the ciphertext blocks that result.

Step 3: Decryption Process:

- Each ciphertext-block in the encrypted message, compute the corresponding plaintext block using the formula: plaintext = (ciphertext ^ d) % n.
- Combine the numerical plaintext blocks to reconstruct the original plaintext message.
- Convert the numerical representation back to its original form (e.g., characters, words).

Proposed Hybrid Algorithm

Create a hybrid encryption technique in this work that combines XOR operation with AES and RSA to safeguard data flow in databases. The plain text is encrypted using the AES technique. The "Secret Key" is the encrypted key that is produced by XORing the AES key block size with the Initialization vector (IV). The RSA encryption method is also used to encrypt this secret AES key. This method encrypts data using the secret key (AES-XOR key) and the "secret key" encrypted using the RSA public key in a single execution step as opposed to many execution steps. The input is the original data, while the output is the encrypted data. The secret key is initially decrypted using the RSA private key, and it is then further decrypted at the receiving end using the XOR operation to finish the decryption process. The opposite of the coding process is the decoding process, which retrieves the original text. The proposed hybrid algorithm's working diagram is displayed in Figure 2.

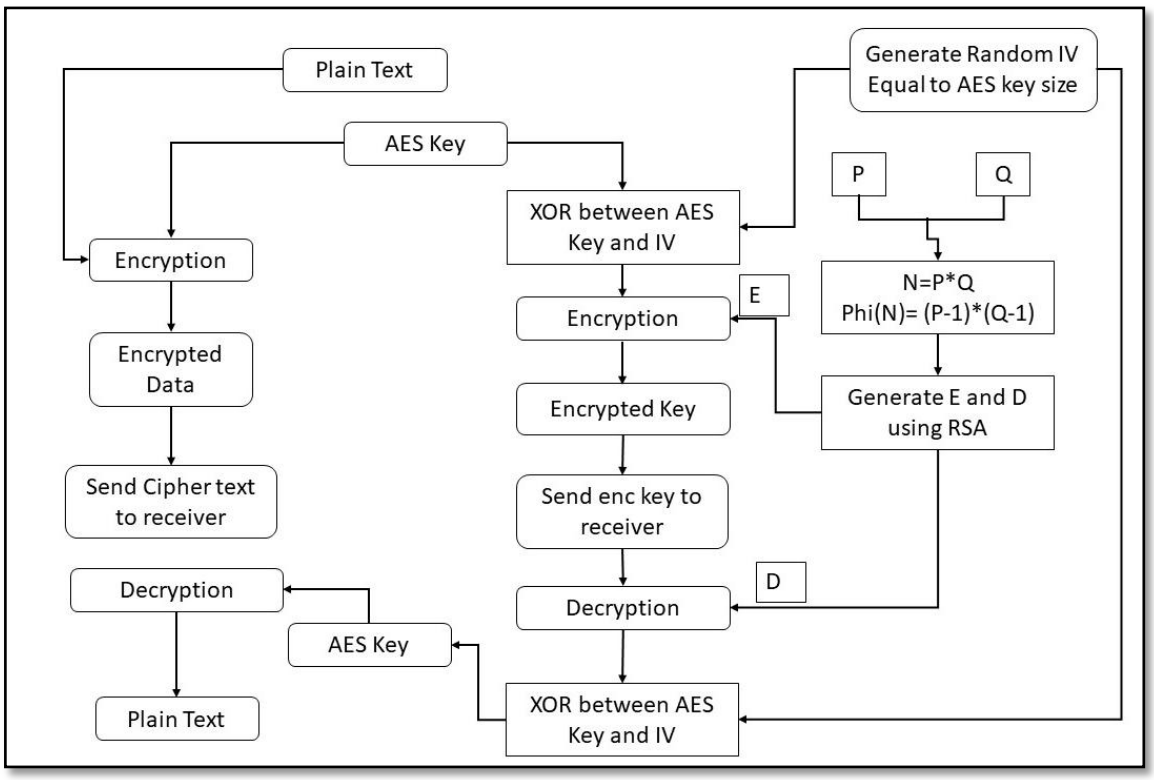

Figure 2 Working Flow Chart of Proposed Hybrid Algorithm

This work provides a hybrid encryption system that uses two different encryption algorithms to safeguard data transit: AES with XOR operation (AES Key) and RSA algorithm.

Step1:

- Produce a random (128 bit) AES key, K1
- “P and Q” are two significant prime numbers.

Step2:

- For AES, create the Initialization Vector (IV).

Step3:

- Calculate N= P * Q.

Step4:

- Find Phi (N) = (P-1)*(Q-1)

Step5:

- Search the number E, such that GCD [E, Φ (N)] =1. Φ (N. Where 1<E< Φ (N)

Step6:

- Calculate D, where E*D = 1 mod Φ (N).

Step7:

- Use the AES Key method to encrypt the message and produce the cypher text C1.

Step 8:

- X-OR between AES key (K1) and IV
- S1= K1 ⊕ IV

Step9:

- Encrypt the XORed symmetric key using RSA public key K2=(S1^E)mod N.

Step 10:

- At receiver side Decrypt the XORed symmetric key using RSA private key K3=(K2^D) mod N.

Step11:

- Receiver side X-OR operation is between K3 and IV,
- S2= K3 ⊕ IV.

Step12:

- Utilize the AES method to decrypt message C1.

Implementation And Results

The technique is implemented in this work, using the C# Windows based programming language(DOT NET 7 Platform), Intel Core i5-7300U processor having 2 core, 4 logical processor with 2.70 GHz processor frequency and the Windows 11 operating system is supported by the Visual Studio 2022 compilation tool and SQL Server 2008 for database.

The GUI was developed using Microsoft .net framework. The programming language is c#. It provides option to choose hybrid encryption type and encrypt the plain text to cipher text and store encrypted data to database. Similarly, it provides the feature to decrypt the stored data in database to plain text. With the help of this implementation, we examine the effectiveness of the selected hybrid type algorithm in manner of encryption/decryption time, memory uses, CPU utilization and power consumption.

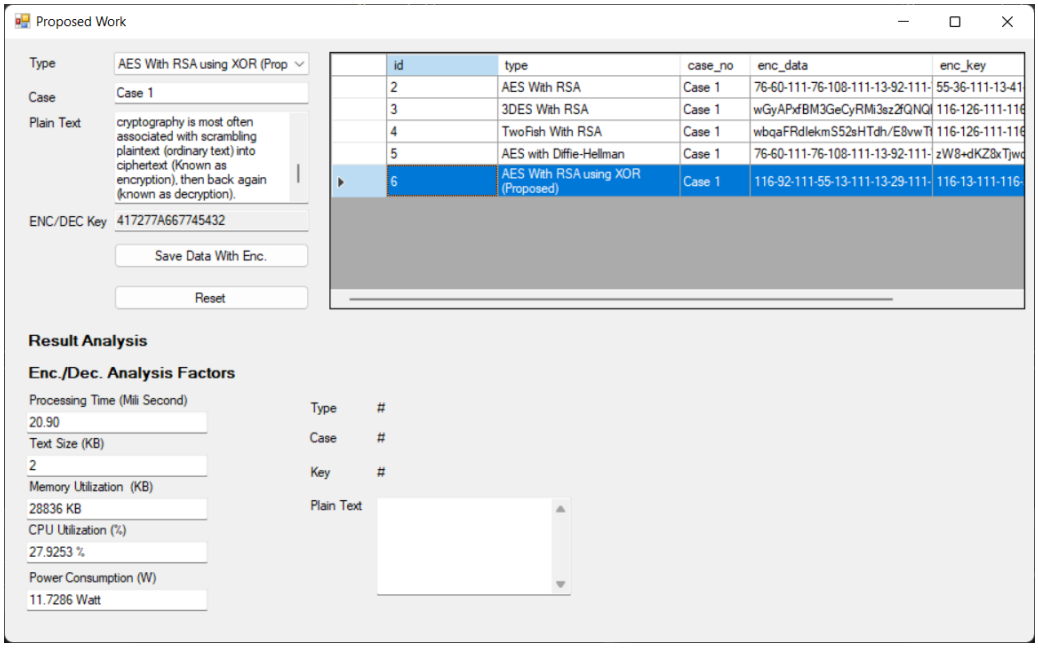

Figure 3(a)Encryption process of proposed hybrid algorithm using GUI.

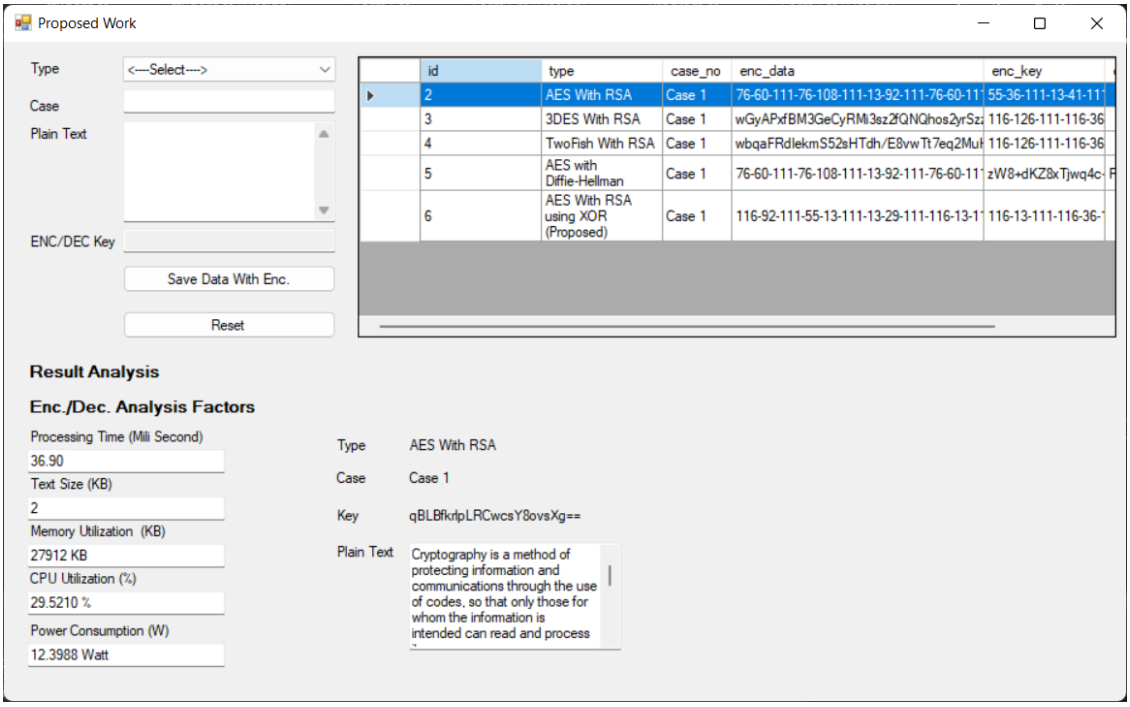

Figure 3(b) Decryption process of proposed hybrid algorithm using GUI.

Encryption and Decryption Time Comparison:

In this experiment, five distinct encryption and decryption methods are used to the same set of data. The experiment analysis result is more accurate when the data from several runs are averaged and the value with the largest error is eliminated. Comparisons are made between the results of the existing algorithms and proposed hybrid algorithms, with an emphasis on the variations in encryption and decryption times. The encryption and decryption schedule for each of them for different file sizes is displayed in Table 1.1 and Table 1.2 simultaneously.

Input Size (KB)	AES + RSA	AES + Diffie Hellman	3DES + RSA	TwoFish + RSA	Proposed Hybrid
	ET (MS)	ET (MS)	ET (MS)	ET (MS)	ET (MS)
2	34.91	68.79	54.85	74.56	20.90
4	64.86	140.06	102.46	187.50	38.14
6	173.63	269.31	221.47	356.86	108.34
8	391.28	432.34	411.81	523.30	278.45
10	702.46	825.46	763.96	915.31	612.20
12	1137.35	1347.05	1235.20	1602.10	1035.63
Total A/Time (Second)	2.50	3.08	2.79	3.66	2.09
Throughput (MB/S)	0.0164	0.0133	0.0147	0.0112	0.0196

Table 1.1 Encryption time of different Hybrid algorithms and Proposed Hybrid algorithm.

According to the above table, the encryption graph of the Hybrid Algorithms is analysed, as shown in figure 4.

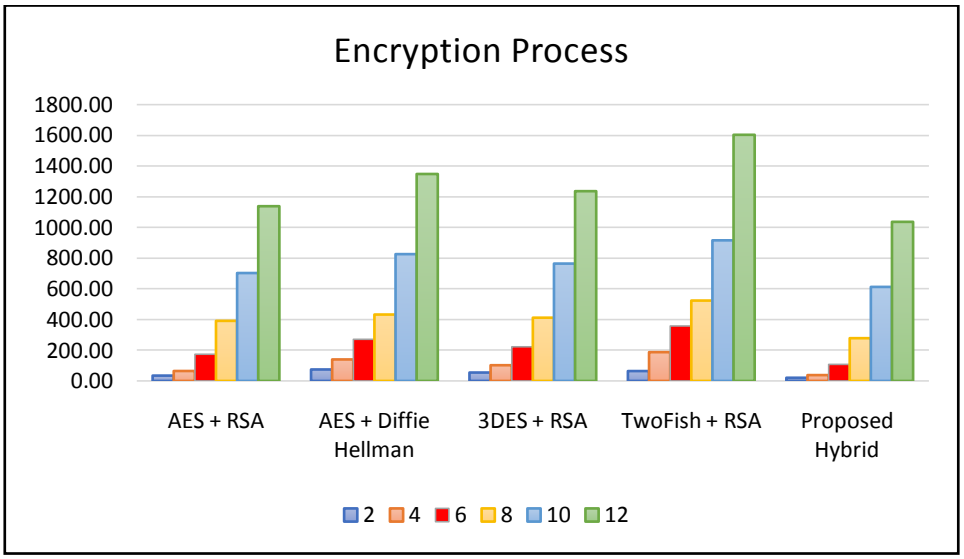

Figure 4 Graphical representation of Encryption time of different Hybrid algorithms and Proposed Hybrid algorithm

Input Size (KB)	AES + RSA	AES + Diffie Hellman	3DES + RSA	TwoFish + RSA	Proposed Hybrid
	DT (MS)	DT (MS)	DT (MS)	DT (MS)	DT (MS)
2	36.89	82.15	72.43	90.67	31.76
4	95.79	179.36	150.13	236.19	56.13
6	225.65	310.23	299.16	419.26	130.46
8	486.12	526.79	510.36	681.82	378.12
10	798.52	912.37	862.44	1104.11	711.56
12	1385.42	1481.26	1451.72	1826.31	1146.83
Total A/Time (Second)	3.03	3.49	3.35	4.36	2.45
Throughput (MB/S)	0.0135	0.0117	0.0123	0.0094	0.0167

Table 1.2 Decryption time of different Hybrid algorithms and Proposed Hybrid algorithm.

According to the above table, the encryption graph of the Hybrid Algorithms is analysed, as shown in figure 5

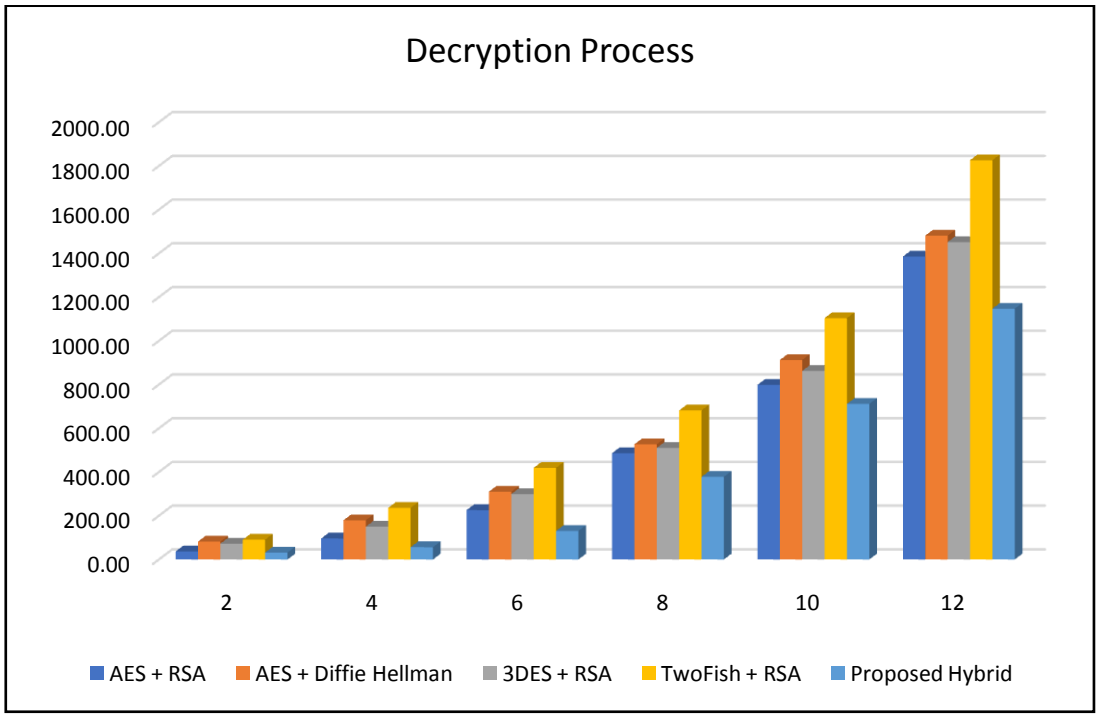

Figure 5 Graphical representation of Encryption time of different Hybrid algorithms and Proposed Hybrid algorithm.

Throughput

A system's throughput is defined as the volume of data that goes across it. This is obtained by dividing the total amount of data sent in Megabytes by typical time needed to transfer all the data in seconds. According to Figure 5, each hybrid algorithm's throughput value in (MB/Sec) and analysis is displayed.

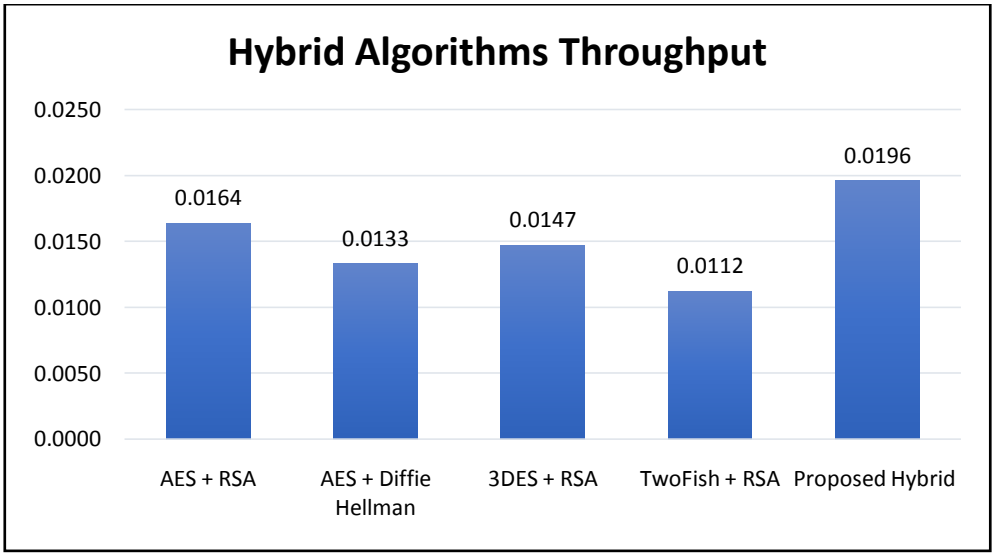

Figure 6 Throughput value of the algorithms

The graph shows comparison of execution (Encryption and Decryption) time between existing hybrid model and new proposed model. When comparing with existing hybrid algorithms, proposed model requires less time for encryption and decryption and proposed model is more secured cryptography algorithm than the other algorithms, because proposed model includes XOR concept, which is more difficult for the intruder to find the plain text from the secrete message.

Conclusion

In this study, to boost security, hybrid key pairs in this study mix symmetric and asymmetric key pairs. In this work, two algorithms are used. The XOR-operated AES method is the first, and the RSA algorithm is the second. The AES algorithm is used to encrypt and decode messages, after which the AES key is XORed with the help of IV to create a secret key. An AES XORed key is encrypted using the RSA method and then sent to the recipient. On the other hand, the AES key is decoded using the RSA private key, offering security from outsiders and attackers. After implementation this hybrid algorithm, we compare this proposed algorithm with existing hybrid algorithm.

According to the above analysis used in this paper, the hybrid encryption algorithm can be used in database design, system design, and other fields where the exchange of secure data is necessary. The algorithm can effectively protect data while also providing performance and a quick execution time, as the results showed that the hybrid encryption is faster than the compared algorithm i.e. AES with RSA, AES with Diffie Hellman, 3DES with RSA and TwoFish with RSA. By combining encryption and XOR operation in the proposed hybrid encryption algorithm, we can boost the message's security while simultaneously increasing its complexity. In comparison to the standard hybrid technique, the newly suggested model offers greater security and fast execution.

References

[1.] Sun, Hung-Min, Mu-En Wu, Wei-Chi Ting, and M. Jason Hinek. 'Dual RSA and its security analysis.' Information Theory, IEEE Transactions on 53, no. 8 (2007): 2922-2933.

[2.] Chhabra, A., & Mathur, S. (2011, October). Modified RSA Algorithm: A Secure Approach. In Computational Intelligence and Communication Networks (CICN), 2011 International Conference on (pp. 545-548). IEEE.

[3.] Wang Rui; Chen Ju; Duan Guangwen, 'A k-RSA algorithm,' Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on, vol., no., pp.21,24, 27-29 May 2011

[4.] Kaur, Khushdeep, and Er Seema. 'Hybrid Algorithm with DSA, RSA and MD5 Encryption Algorithm for wireless devices.' International Journal of Engineering Research and Applications (IJERA) 2.5 (2012): 914-917

[5.] Al-Hamami, A. H., &Aldariseh, I. A. (2012, November). Enhanced Method for RSA Cryptosystem Algorithm. In Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference on (pp. 402-408). IEEE.

[6.] Pugila, D., Harsh Chitrala, SalpeshLunawat, and PM Durai Raj Vincent. 'AN EFFICEIENT ENCRPYTION ALGORITHM BASED ON PUBLIC KEY CRYPTOGRAPHY.' International Journal of Engineering and Technology (2013).

[7.] Nedjah, A., de Macedo Mourelle, L., Wang, C.: A parallel yet pipelined architecture forefficient implementation of the advanced encryption standard algorithm on reconfigurablehardware. Int. J. Parallel Program.44(6), 1102–1117 (2016).

[8.] Yang, L.T., Huang, G., Feng, J., Xu, L.: Parallel GNFS algorithm integrated with parallelblock Wiedemann algorithm for RSA security in cloud computing. Inf. Sci.387,(2016)

[9.] Moumen, A., Sissaoui, H.: Images encryption method using steganographic LSB method,AES and RSA algorithm. Nonlinear Eng. Model. Appl.6(1), 53–59 (2017).

[10.] Zhang, W., Zhou, R., Gao, Y., Wang, J.: File encryption based on AES algorithm. Softw.Guide16(06), 180–182 (2017).

[11.] Riaz, M.N., Ikram, A.: Development of a secure SMS application using advanced encryptionstandard (AES) on android platform. Int. J. Math. Sci. Comput. (IJMSC)4(2), 34–48 (2018).

[12.] You, Y.: Design and implementation of combined encryption algorithm based on AES andRSA in DOA. Chengdu University of Technology (2018).

[13.] Patel, G. R., & Panchal, K. (2014). Hybrid Encryption Algorithm. Int. J. EngineesingDevelopment Res., 2(2).

[14.] Yang, J.: Design and implementation of an AES algorithm encryption transmission system.Electron. Des. Eng.27(03), (2019).

[15.] Kumar, M. T., Katragadda, R. K., Kolli, V. S. and Rahiman, S. L., (2019). "*A hybrid approach for enhancing security in internet of things (IoT)*". Proc. Int. Conf. Intell. Sustain. Syst. ICISS 2019, pp. 110–114.

[16.] Zou, L., Ni, M., Huang, Y., Shi, W. and Li, X, (2020). "*Hybrid encryption algorithm based on AES and RSA in file encryption".* Springer volume 551 https://doi.org/10.1007/978-981-15-3250-4_68

[17.] Guru, M. A., &Ambhaikar, A. (2021). AES and RSA-based Hybrid Algorithms for Message Encryption & Decryption. Information Technology in Industry, 9(1), 273-279.

[18.] G. Chaloop, S., & Z. Abdullah, M. (2022). ENHANCING HYBRID SECURITY APPROACH USING AES AND RSA ALGORITHMS. Journal of Engineering and Sustainable Development, 25(4). https://doi.org/10.31272/jeasd.25.4.6.