DESIGN AN ALGORITHM FOR DATA ENCRYPTION USED IN HYBRID TECHNIQUE TO IMPLEMENTATION IN DATABASE

Satinder* and Dr. Parveen Sehgal**
* 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
drparveensehgal@osgu.ac.in, drparveensehgal@gmail.com 1
satindercse192@osgu.ac.in, satindershanky@gmail.com 2

ABSTRACT
Data encryption has evolved into a crucial component of information security in a time when digital data is proliferating. Strong encryption solutions are required because databases, which frequently hold sensitive and important data, are a prominent target for cyberattacks. In order to secure databases, this research suggests a unique hybrid encryption algorithm that makes use of both symmetric and asymmetric encryption techniques. The suggested method offers an effective and reliable data protection solution by combining the speed of symmetric encryption with the security of asymmetric encryption.

The program divides the data into blocks before encrypting each block individually using a symmetric encryption technique, such as Advanced Encryption Standard (AES). This method makes use of symmetric encryption's speedy ability to encrypt huge volumes of data. Then, taking use of RSA's ability to safely exchange keys across an unsecure network, the symmetric key used for this procedure is encrypted using an asymmetric encryption technique.

Our technique guarantees a robust encryption standard for data kept in databases by utilizing this hybrid approach, so raising the degree of security and safeguarding sensitive data from unapproved access and online dangers. The suggested hybrid method has further shown comparable computational efficiency, making it appropriate for settings with high data volume.

This study presents a detailed examination of the design, implementation, and performance of the proposed hybrid encryption method. Additionally, it looks at possible application fields, scalability, and viability in actual database systems. Last but not least, we go through the algorithm's comparative advantage over competing approaches and possible upgrades for further study.

Keywords: Database, Hybrid Encryption, AES, RSA, Security.

INTRODUCTION

Data security is crucial in today's interconnected digital environment, especially when it comes to database administration. Numerous sensitive pieces of information, including private financial and personal information as well as confidential business insights, are stored and managed by organisations from a variety of industries using databases. Strong encryption measures are required to protect this data from illegal access and potential breaches as its volume and value increase. Data protection is quick and effective using traditional symmetric encryption, which is demonstrated by techniques like the Advanced Encryption Standard (AES). When it comes to safely transmitting encryption keys, it presents a substantial issue. However, asymmetric encryption, as represented by algorithms like the Rivest-Shamir-Adleman (RSA) method, effectively solves the key exchange problem at the expense of efficiency, especially when encrypting large amounts of data.

A hybrid encryption strategy shows promise as a technique to establish a balance between security and effectiveness. In this paradigm, symmetric and asymmetric encryption's advantages are integrated to forge a solid framework for safe data management. In this study, we introduce a novel hybrid data encryption algorithm that is specifically intended for database implementation. It aims to combine secure key exchange with RSA with fast, efficient data encryption with AES to deliver the best of both worlds. Our suggested approach, dubbed the "Hybrid Encryption Algorithm," is focused on safeguarding sensitive data kept in databases while guaranteeing that encryption and decryption procedures are carried out quickly. We intend to provide a comprehensive encryption framework suited for the specific requirements of contemporary database systems by using the speed of AES with XOR operation for data encryption and the secure key distribution of RSA.

In this paper, we present a detailed overview of the Hybrid Encryption Algorithm, discussing its block diagram or working principles, key generation, encryption process, and key management strategies. We address the challenges of securely storing encryption

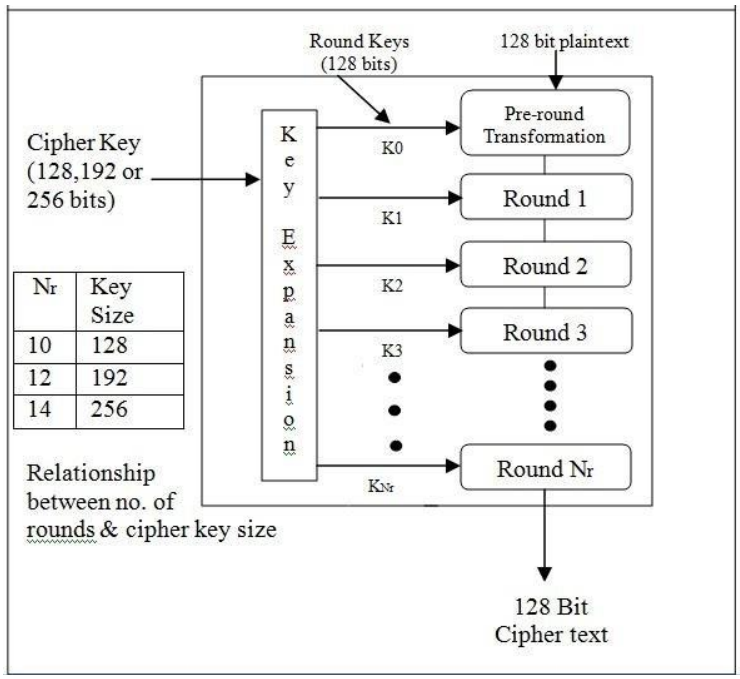
keys, mitigating potential vulnerabilities, and optimizing performance to make the algorithm suitable for real-world database applications.

Furthermore, we conduct a thorough evaluation of the Hybrid Encryption Algorithm, analyzing its security aspects, performance characteristics, and effectiveness in safeguarding data. By comparing it with other encryption techniques commonly used in database security, we aim to demonstrate the advantages of our proposed hybrid approach and highlight its potential impact on enhancing database security in various industries.

ABOUT AES AND RSA ALGORITHM

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.



Nr	Key Size
10	128
12	192
14	256

"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.

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).

METHODOLOGY FOR HYBRID ENCRYPTION METHOD

In a hybrid cryptosystem, the receiver performs the following actions to encrypt a message directed to the sender:

- Obtain the sender's public key.
- Create a "symmetric key" for the data encapsulation method.
- Use the data encapsulation method and the freshly produced symmetric key to encrypt the message.
- Encrypt the symmetric key utilising the sender's public key using the key encapsulation technique.
- Send both of these encryptions to the sender.

To decrypt this hybrid cipher text, the sender does the following:

- Use the receiver's private key to decode the symmetric key kept in the key encapsulation section.
- The message of the data encapsulation section can be decoded using this symmetric key.

Proposed Hybrid Algorithm

In order to secure data flow in databases, create a hybrid encryption method in this work that combines the XOR operation with the AES and RSA encryption algorithms: The AES technique is used to encrypt plain text, and the AES key and its block size are XORed using ASCII codes. This secret AES key is additionally encrypted using the RSA technique. In a single execution step as opposed to many execution stages, this approach encrypts data using the secret key (AES-XOR key) and the "secret key" encrypted using the public key of RSA. The original data is the input, while the encrypted data is the output. To complete the decryption procedure, the secret key is first decrypted using the RSA private key and then further decrypted using the XOR operation at the receiver end. The decoding process, which recovers the original text, is the antithesis of the coding process. Figure 2 shows the working diagram of the proposed hybrid Algorithm.

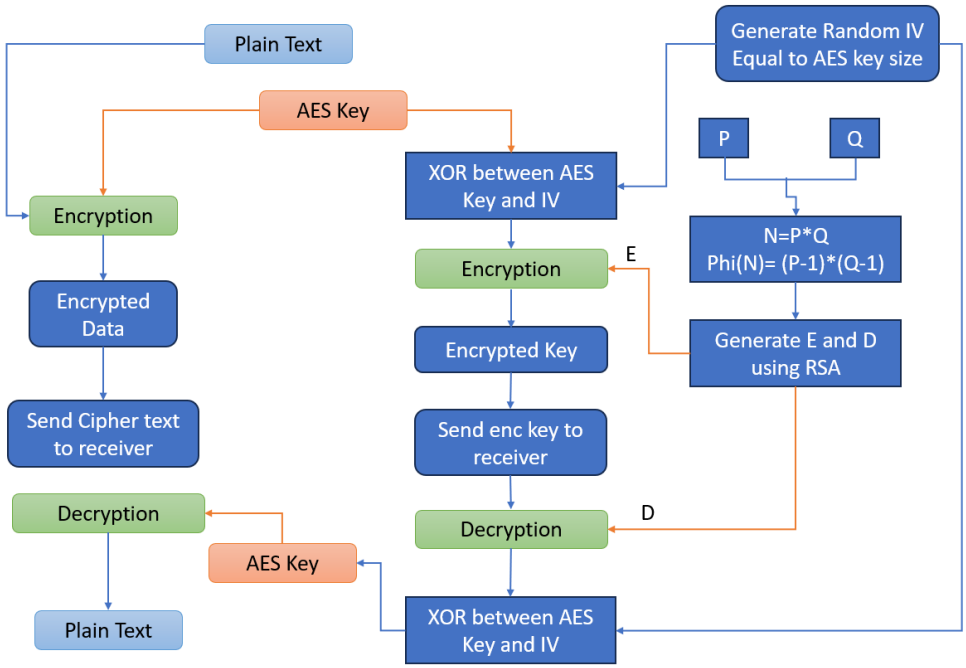


Figure 2 Working Diagram of Proposed Hybrid Algorithm

"Hybrid algorithm"

This work provides a hybrid encryption system that uses two different encryption algorithms to safeguard data transit: AES with XOR operation 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# programming language, and the Windows 11 operating system is supported by the Visual Studio 2022 compilation tool. The experiment's files all have the "Test Case.txt" extension and range in size from 1 to 30 KB.

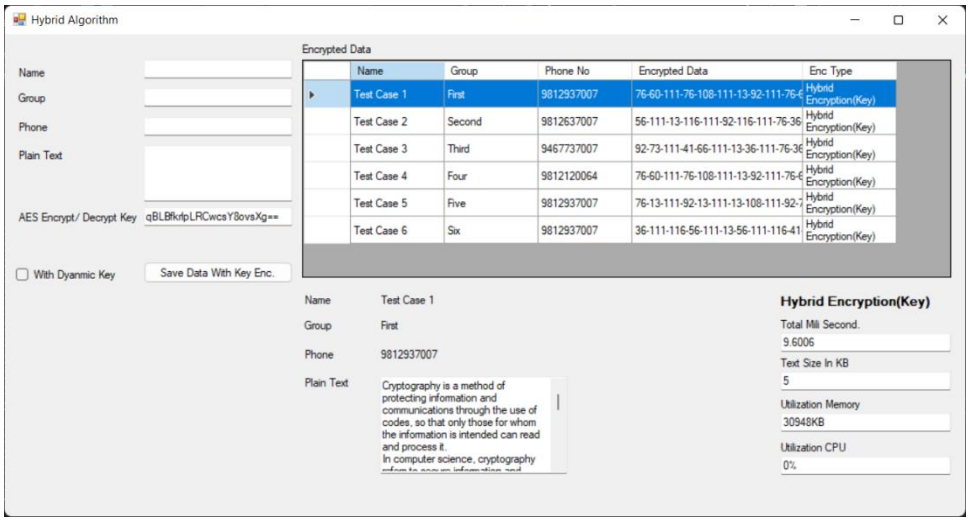


Figure 3 Decryption process of proposed hybrid algorithm using GUI

In the above implementation, the "memory file mapping" method is used to increase output, decrease disc visits, and lessen frequent access to data kept in large files. This is due to the fact that when performing an read/write file operation, two copies of the data are created: one in kernel space and one in user space. This is done by first copying the contents of the file from the hard drive to the kernel space buffer. In contrast, the real data copy in memory mapping occurs when there is a missing page interrupt. Only one data copy is necessary because of the direct mapping between the file and user space; no more data copies are needed. In this experiment, memory file mapping was used to boost file reading and writing productivity and guarantee the consistency of the algorithm test environment.

Three different encryption and decryption techniques are used on the same batch of data in this experiment. Data from multiple runs are averaged, and the value with the highest error is subtracted in order to increase the accuracy of the experimental analysis result. The findings of the RSA, AES, and proposed hybrid algorithms are compared, and the differences in encryption and decryption times are examined. Table 1.1 shows the encryption schedule for each of them for various file sizes.

Input Size (KB)	AES	RSA	Proposed Hybrid
	ET (MS)	ET (MS)	ET (MS)
5	4.13	54.45	1.51
10	10.67	187.51	2.00
15	12.87	356.87	2.79
20	17.97	623.30	4.98
25	22.35	1015.32	6.58
30	24.72	3702.10	7.51
Total A/Time	92.72	5939.54	25.37
Throughput (MB/S)	1.11	0.02	4.04

Table 1.1 RSA, AES, and proposed hybrid algorithm encryption Execution time

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

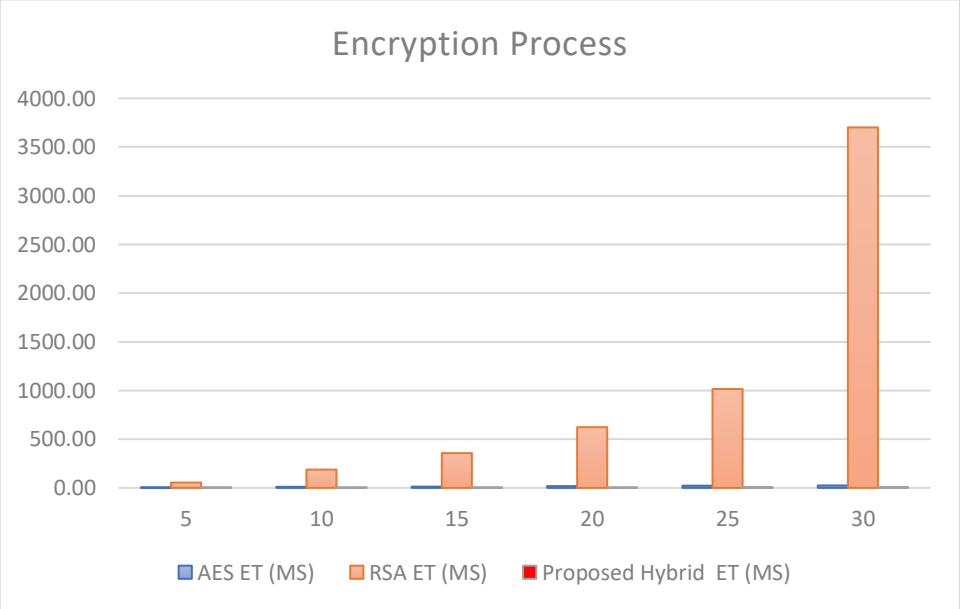


Figure 4 AES(X-OR), RSA and proposed hybrid encryption

Compare the three encryption times after analysing the experimental data: AES requires more time to encrypt data as they grow, and the rate of growth is slow; the RSA algorithm's execution time nearly doubles with growing file sizes; The proposed hybrid encryption method encrypts data faster than the AES algorithm. The decryption process schedule for each of the three methods is shown in Table 1.2

Input Size (KB)	AES	RSA	Proposed Hybrid
	DT (MS)	DT (MS)	DT (MS)
5	10.63	12.7	9.60
10	11.55	16.76	10.87
15	14.00	25.53	14.01
20	18.53	21.88	16.86
25	19.04	30.41	17.03
30	23.58	41.98	18.49
Total A/Time	97.33	149.26	86.86
Throughput (MB/S)	1.05	0.69	1.18

Table 1.2. RSA, AES, and proposed hybrid algorithm decryption Execution time

The experimental data analysis demonstrates that the RSA decryption time nearly doubles with growing file size. On the other hand, the AES decryption time increases more slowly as the text size increases. The proposed hybrid (AES and RSA) encryption algorithm's decryption times are comparable to those of AES, showing that it successfully balances security and efficiency.

The decryption execution time of the three techniques are examined using the information from the decryption process schedule, as shown in Fig. 5

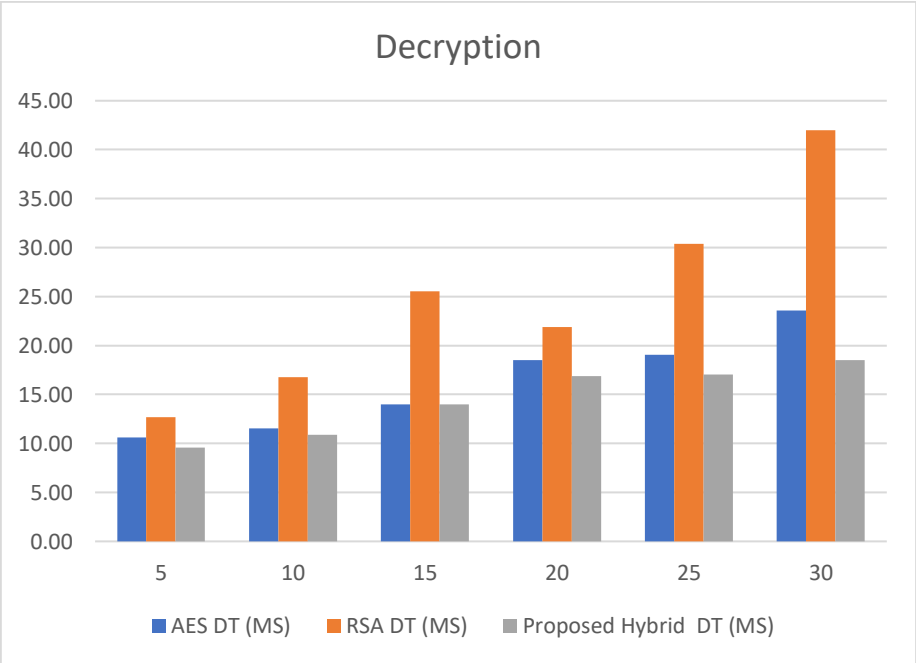


Figure 5 RSA, AES and hybrid algorithm decryption time chart

The graph contrasts the overall decryption times (MS) for RSA, the newly proposed hybrid approach, and the regular AES algorithm. According to the results, the suggested model performs faster at decrypting data than both AES and RSA. The suggested model is also thought to be more secure than RSA since it uses the XOR idea, which makes it more difficult for hackers to extract the plaintext from an encrypted message.

Throughput

The amount of data that passes through a network system is referred to as its throughput. It is calculated by dividing the total data delivered in Megabytes by the usual time required to transport the entire amount of data, expressed in seconds. The throughput value in (MB/Sec) for each algorithm is shown in Table 1.3. Throughput analysis is shown in Figure 6.

AES	RSA	Proposed Hybrid
1.05	0.69	1.18

Table 1.3 Throughput value of the algorithms

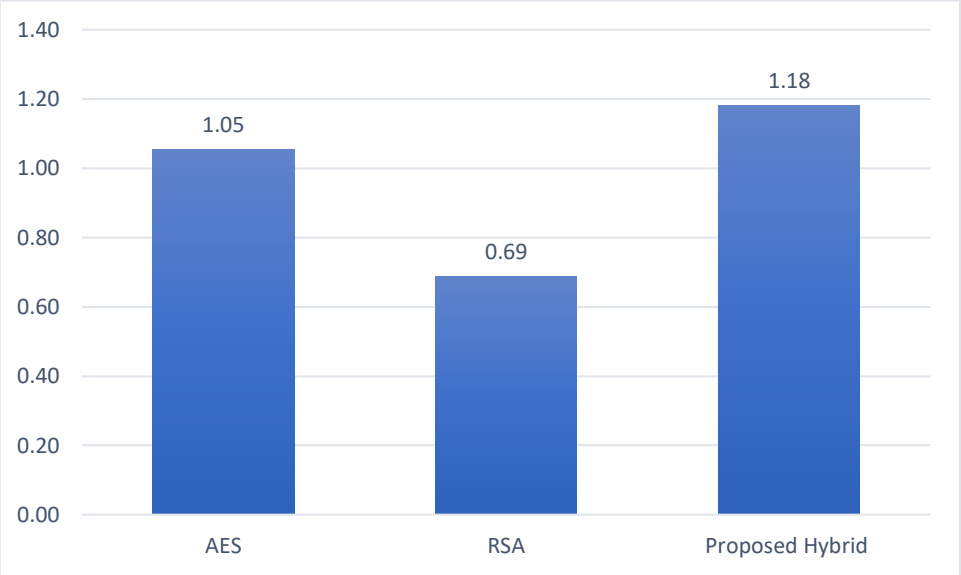


Figure 6 Throughput value of the algorithms

The proposed hybrid encryption technique can be employed in software applications, system design, and other sectors where safe data sharing is required, according to the aforementioned experiments and analyses used in this study. The results showed that the hybrid encryption is substantially faster than the RSA and more secure than the AES methods, demonstrating that the algorithm can successfully safeguard the data in addition to the performance and rapid execution time.

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.

According to the experiments 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 RSA algorithm and AES algorithm. 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 RSA technique, the newly suggested model offers greater security.

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, Salpesh Lunawat, 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). Dr Asha Ambhaikar, M. A. G. (2021). AES AND RSA-BASED HYBRID ALGORITHMS FOR MESSAGE ENCRYPTION & DECRYPTION. INFORMATION TECHNOLOGY IN INDUSTRY, 9(1). https://doi.org/10.17762/itii.v9i1.129

[13.] 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

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

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

[16.] 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.

[17.] 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