International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

AES-RSA: An Innovative Hybrid Security Framework for File Authentication, Integrity, and Data Secrecy Model

Smita Rath[*1], Sushree Bibhuprada B. Priyadarshini[1], Deepak Kumar Patel[2], Prabhat Kumar Sahu[3] Nibedita Jagadev[4], Monalisa Panda[5], Narayan Patra[6], Sipra Sahoo[7]

Submitted: 28/12/2023 **Revised**: 04/02/2024 **Accepted**: 12/02/2024

Abstract: Data secrecy, integrity, and authenticity are the main aims of cryptography. Only the intended receiver may read a confidential communication, and authenticated messages will not change in transit. A guarantee that the communication comes from the intended sender comes through authentication. Symmetric and asymmetric cryptography come within the two categories of cryptography. This study uses the key management issue associated with Advanced Encryption Standard (AES) by incorporating the Rivest-Shamir-Adleman (RSA) algorithm to mask the symmetric key. By utilizing RSA, the security of AES provides a secure method for key exchange and management. In the case of symmetric cryptography, the same key is used, unlike asymmetric cryptography; there are two keys, like a public key and a private key, as in RSA. The AES encryption method shows a strong avalanche effect. This paper proposes a novel dynamic AES-RSA hybrid algorithm that introduces modifications to the traditional AES structure. Specifically, we alter the order of the Mix Columns and Shift Rows operations and increase the number of rounds in the AES algorithm to enhance the cryptographic strength and resistance against attacks. Our experimental results demonstrate that the dynamic AES-RSA algorithm with masked key management offers improved security and resistance against cryptographic attacks. The findings of this research paper highlight the importance of robust encryption techniques in ensuring file security. The proposed modifications to the AES algorithm provide a more secure and safe approach to protecting sensitive data.

Keywords: *Advanced Encryption Standard (AES), Asymmetric Cryptography, Encryption, Decryption, Rivest-Shamir-Adleman (RSA), Symmetric Cryptography*

1. Introduction

The process of encrypting data in plain text into a code that can't be read by anyone who doesn't have the key to decrypt it is known as file encryption. The protection of sensitive data from espionage, theft, and unauthorized access is the goal of file encryption. File encryption uses modified AES algorithms to scramble the data in a file, making it unreadable to someone who does not have the key to unscramble it. This paper includes the importance of file encryption, how it works, and some of the oldest and most popular encryption techniques in use today. With the rise of digital technology, file encryption has become more important than ever before. Hackers are continuously seeking for new ways to steal sensitive data, including social security numbers, credit card numbers, and other personal data, while cybercrime is on the increase. Without file encryption, this data would be easy for hackers to access and use for fraudulent purposes. In addition to protecting against cybercrime, file encryption is also important for protecting intellectual property. Companies and organizations often store valuable information in digital files, such as trade secrets, financial information, and client data. File encryption aids in maintaining the security and privacy of this data.

Data secrecy, integrity, and authenticity are the main aims of cryptography. Only the intended receiver may read a communication that is confidential, and a message that is authenticated has not been changed in transit. A guarantee that the communication comes from the intended sender is provided through authentication.

- A single key is used in symmetric cryptography for both encryption and decryption. The transmitter and receiver must share the same secret key for this approach to work none the less,

**,1Associate Professor, Department of Computer Science & Information Technology, Siksha 'O' Anusandhan Deemed to Be University, India.smitarath@soa.ac.in*
1Associate Professor, Department of Computer Science & Information Technology, Siksha 'O' Anusandhan Deemed to Be University,India. bimalabibhuprada@gmail.com
2Associate Professor, Department of Computer Science & Information Technology, Siksha 'O' Anusandhan Deemed to Be University, India. deepakpatel@soa.ac.in
3Associate Professor, Department of Computer Science & Information Technology, Siksha 'O' Anusandhan Deemed to Be University, India. prabhatsahu@soa.ac.in
4Associate Professor, Department of Computer Science and Engineering, Siksha 'O' Anusandhan Deemed to Be University, India. nibeditajagadev@soa.ac.in
5Associate Professor, Department of Computer Science and Engineering, Siksha 'O' Anusandhan Deemed to Be University, India. monalisapanda@soa.ac.in
6Associate Professor, Department of Computer Science & Information Technology, Siksha 'O' Anusandhan Deemed to Be University, India. narayanpatra@soa.ac.in
7Associate Professor, Department of Computer Science and Engineering, Siksha 'O' Anusandhan Deemed to Be University, India .siprasahoo@soa.ac.in
** Corresponding Author Email: smitarath@soa.ac.in*

it is often quicker than asymmetric cryptography.

- On the other hand, asymmetric cryptography employs a pair of keys—one for encryption and the other for decryption. Due to the complexity of the underlying techniques, this approach can be slower than symmetric cryptography but is more secure.

The Advanced Encryption Standard (AES) is one of the cryptographic algorithms that is most extensively used. AES is a symmetric encryption technique that has a changeable key size of 128, 192, or 256 bits and a 128-bit block size. Passwords, sensitive data like credit card numbers, and other private information are secured with it. The Data Encryption Standard (DES), which creates a 56-bit key size and a 64-bit block size, is another well-liked symmetric method. DES is still often used in older systems even though it is less secure than AES. Common applications of asymmetric cryptography include key exchange and digital signatures. The RSA algorithm, which bears the names of its creators Ron Rivest, Adi Shamir, and Leonard Adleman, is the most used asymmetric algorithm.

RSA encrypts data using a public key and decrypts it using a private key. Private keys are not shared and kept as secret, but public keys are freely shared. Online banking, secure email, and other applications that demand secure communication all make use of RSA[1-3]. Cryptography is not fool proof, and there are still vulnerabilities and weaknesses that can be exploited by attackers. For example, the use of weak keys, weak algorithms, or poor implementation can compromise the security of a cryptographic system. Additionally, advances in computing power have made it possible to brute-force attack encryption keys, making it essential to use strong encryption algorithms and key sizes. Cryptography is a critical field that plays a vital role in ensuring secure communication in today's digital world. It is used extensively in various applications and is essential for protecting sensitive information. While cryptography is not perfect, it continues to evolve, and new algorithms and protocols are constantly being developed to improve its effectiveness and security.

2. Methodology

This chapter presents the proposed model, which aims to enhance the security and robustness of the Advanced Encryption Standard (AES) method. The model incorporates modifications to the traditional AES structure, focusing on the dynamic rearrangement of operations and the integration of a masked key management mechanism using the Rivest-Shamir-Adleman (RSA) algorithm. This chapter provides an in-depth explanation of the proposed model, detailing the rationale behind each modification and the expected benefits in terms of protection and efficacy and a comparative analysis in Table 1. The RSA generates a set of private and public keys to encrypt a file. The public key is shared among all the senders while the private key is shared to the owner who will decrypt the file. The overall safety and management of keys abilities of the RSA method are significantly greater than those of the AES approach. This is its main benefit over any other algorithm. Its safety of it has been significantly enhanced since just the matching encryption key is able to unlock a piece of plaintext that has been encrypted with a particular encryption key. It is more challenging to decrypt a file with a larger key length. The RSA still has limitations. Since it deals with large numbers and its execution time is very slow. Prime factorization is one of the major challenges in the field of RSA approach.

In order to protect sensitive information and defend against unauthorized entry or tampering cryptography is crucial. Masking is an important technique within the area of cryptography that extends the security of cryptographic methods and protocols. In this section we will explore the concept of masking in cryptography, its significance, and its role in strengthening security by splitting secrets. Side-channel attacks exploit unintended data leaks from a cryptographic system, such as power consumption, timing, or electromagnetic emanations. Masking helps mitigate side-channel attacks by ensuring that an attacker gains no valuable information from any subset of shares alone. The secret remains concealed unless all shares are combined.

2.1. Need of Masking in AES

Higher Resistance to Physical Attacks such as fault injection or invasive techniques aims to manipulate the behavior of cryptographic devices to extract sensitive information. Masking increases the complexity of such attacks since an attacker would need to manipulate multiple shares simultaneously across different entities to recover the secret. Masking enables the use of a split trust model, where different entities collaborate to perform cryptographic operations while minimizing the trust placed on any single entity. No single entity has access to the complete secret, increasing the difficulty of a system breach by an attacker.

2.2 Working of RSA Algorithm

A common popular public key encryption method for safe data communication is RSA (Rivest-Shamir-Adleman)[4]. Prime factorization and modular arithmetic are concepts at the core of the RSA algorithm. The steps that make up the operation of RSA encryption are as follows:

Key Generation: A public key and a private key are used in RSA encryption. These keys are created in the following way:

a. Select the two large primes p and q.

b. Determine n = pq.

c. Calculate n's totient by using the formula: (n) = (p-1)(q-1).

d. Pick a modest odd number, e, that is comparatively prime to (n). The public key is e.

e. Determine d, which is e modulo (n)'s modular multiplicative inverse. The private key is d.s

The hybrid encryption strategy, which incorporates the benefits of both symmetric and public key encryption, is a typical example. Using a symmetric key approach such as AES, this method first encodes the message content before encrypting the AES key. The recipient receives both the encrypted message and the encrypted AES key, and uses their private RSA key to first decode the AES key before decrypting the message with the decrypted AES key. This approach allows for efficient encryption of large messages while still providing the security of public key encryption.

Key management-wise, the RSA method outperforms the DES approach. The RSA algorithm allows for the open distribution of encryption keys, making it very simple to update the encryption keys[5-6]. In contrast, the DES algorithm requires the distribution of a secret key prior to communication, making key replacement more difficult. For each communication object, DES also requires the generation and maintenance of a separate key[7].

2.3 Working of AES

AES is an iterative cypher rather than a Feistel cypher. Based on two popular techniques for encrypting and decrypting data, substitution and permutation networks (SPN), it encrypts and decrypts data as shown in Fig.1. SPN, a set of mathematical operations, is used in block cypher techniques [16]. The fixed plaintext block size supported by AES is 128 bits (16 bytes). AES employs a 16-byte matrix, which is represented as a 4x4 matrix. The quantity of rounds is yet another crucial element of AES. The number of rounds depends on how long the key is. The AES algorithm takes into account one of three key sizes, such as 128, 192, or 256 bits. AES uses 10 rounds for keys with 128 bits, 12 rounds for keys with 192 bits, and 14 rounds for keys with 256 bits. The number of rounds depends on the key sizes [17].

The security of the AES method is stronger than that of the DES and 3DES algorithms, yet it is still considerably less resilient than that of the RSA approach. The issue of a short DES length has been addressed by the AES method [8]. It is still possible to defeat the AES algorithm under particular circumstances due to a variety of management of keys challenges, making key distribution and security management challenging. It has these two characteristics: (1) Since the AES technique utilizes the similar key for both data encrypting and decrypting a file. (2) Each time the receiver and sender use the distinctive key that no one else is aware of. This will result in more keys being needed and complicate management. The RSA encrypted public key algorithm is the most well-known of all them. It is a symmetric encryption method and an enormous integer factorization-based method of encryption [9]. The secret key and the publicly accessible key are used alongside the keys for the RSA technique, that are provided in pairs, to encrypt file.The file uses public key to encrypt a data and subsequently decode it using the private key. The efficiency of the RSA algorithm is considerably stronger than that of the AES technique and it offers superior management of keys capabilities. Its key advantage over other algorithms is this. The RSA algorithm uses two sets of keys. Since only the key that matches the encrypted plaintext may be used to decrypt it, the security of the system is greatly strengthened. The difficulty of factoring really large numbers is what it is known as [10]. An RSA algorithm key is harder and harder to decipher the longer the key length. RSA technique and encryption flowchart Though offering robust security, the RSA algorithm performs badly in many areas. The RSA algorithm runs poorly and slowly because it depends on enormous amounts of data.

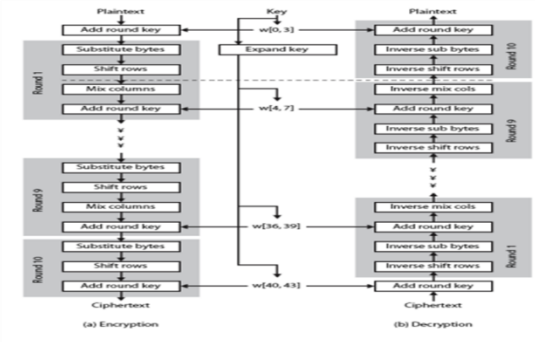

Fig. 1. Basic Structure of AES Encryption Algorithm.

Only small-scale data encryption can be done with it. The majority of RSA algorithm research depends on prime factorization-based mathematical attacks.

2.4 Flowchart of the Proposed Approach AES-RSA Hybrid Model

Although the two above RSA and AES algorithms have distinct properties, they are affected by how they work. The RSA algorithm and AES are the main components of this hybrid approach for encryption and decryption to secure a file[11-15].

The flow diagram of encryption flowchart as shown in Fig. 2. contains a condition checker, public key, private key, RSA, Data.txt, AES, XOR operation between plaintext and IV, padding, add round key operation, iterative operation, and final round. In iterative rounds there are 4 operations:

Substitute bytes, mix columns, shift rows, and add round key. Final round has three operations Substitute byte, mix columns, and add round key. After going through all the operations encryption is complete.

Psuedocode for Encryption

Start

1. Check if Secret/Private key and public key is present in the same directory or not: if not then Step 2 and if present start encryption of file using AES.

2. Generate RSA Key Pair:

2.1. Induce a random prime number p

2.2. Induce one more random prime number q

2.3. Calculate n = p * q

2.4. Calculate φ(n) = (p - 1) * (q - 1)

2.5. Select an integer e such that $1 < e < \varphi(n)$ and gcd (e, φ(n)) = 1

2.6. Calculate d as the modular multiplicative inverse of e modulo φ(n)

2.7. Save the public key (n, e) as "public_key.pem."

2.8. Save the private key (n, d) as "private_key.pem."

2.9. Print "Restart the program."

Exit the program.

If the program is restarted:

1. Load the public key from "public_key.pem".

2. Read the user-provided password.

3. Generate a symmetric key using the password.

4. Masking the symmetric key using RSA algorithm with the public key and stored in data.txt file:

5. Read the file to be encrypted.

6. Perform file encryption using the modified AES algorithm:

6.1. Padding in plain text.

6.2. XOR operation with plain text using initial vector.

6.3. Add Round Key

6.4. Execute Iterative operations of the modified AES encryption operations:

6.4.1. Substitute Bytes

6.4.2. Mix Columns (modified position)

6.4.3. Shift Rows (modified position)

6.4.4. Add Round Key

6.5. Execute the final round of modified AES encryption:

6.5.1. Substitute Bytes

6.5.2. Mix Columns (modified, replaced with shift rows)

6.5.3. Add Round Key

7. Save the encrypted file with (.enc) extension at the end.

8. Print "Encryption complete."

9. Print "Execution time".

10. Remove the original file.

End

Decryption flowchart as shown in Fig. 3. Contains two condition checker, private key form RSA, Data.txt, AES, XOR operation between cipher text and IV, padding, add round key operation, iterative operation, and final round. In iterative rounds there are 4 operations: inverted Substitute bytes, inverted mix columns, inverted shift rows, and add round key. Final round has three operations inverted Substitute byte, inverted mix columns, and add round key. After going through all the operations decryption is complete.

Pseudocode for Decryption

Start

1. Check if private key is present or not if not then exit.

2. Load the private key from "private_key.pem".

3. Read the user-provided password.

4. Decrypt the symmetric key present in data.txt using Private key using RSA algorithm.

5. Check the Decrypted symmetric key is same as user provided password or not

5.1. If both are same then proceed to the next step.

5.2. If both are not same them print "Incorrect Password".

6. Perform file decryption using the modified AES algorithm:

6.1. Execute the final round of modified AES encryption first:

6.2. Add Round Key

6.3. Inverted Mix Columns (modified, replaced with inverted shift rows)

6.4. Inverted Substitute Bytes

6.5. Perform the iterative rounds of modified AES.

6.5.1. Add Round Key

6.5.2. Inverted Substitute Bytes

6.5.3. Inverted Mix Columns (modified position)

6.5.4. Inverted Shift Rows (modified position)

7. Add Round Key.

8. XOR operation of cipher text and Initial Vector.

9. Unpad the cipher text.

10. Save the decrypted plaintext in a text file.

11. Print “Decryption complete.”

12. Print “Execution time”.

13. Remove the (.enc) file.

End

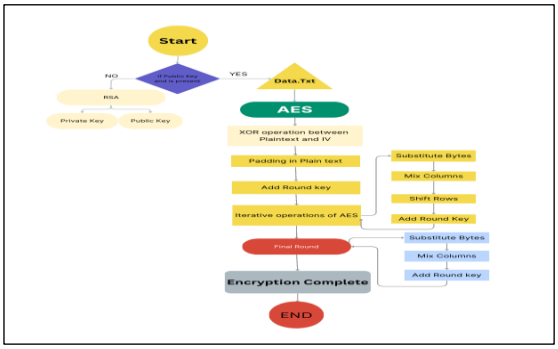

Fig. 2. Hybrid AES Encryption

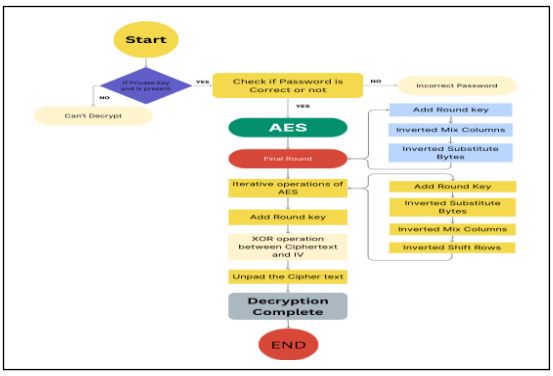

Fig. 3. Hybrid AES Decryption

Table 1: A Brief overview of cryptography algorithms

Algorithm	**Speed**	**Cost (Computation**	**Type (Symmetric/ Asymmetric)**	**Key Size**	**Security Level**	**Best Uses**	**Standardization and Adoption**
RSA	Slow-moving	Max	Asymmetric	1024-8192 bits	High(Depends on key size)	safe exchange of keys, digital signatures, authentication, SSL/TLS handshakes	Widely Adopted, Being phased out
ECC	Moderate	Moderate	Asymmetric	160-512 bits	High	safe exchange of keys, digital signatures, authentication, SSL/TLS handshakes	Increasing adoption, NIST & industry Standard
Two Fish	Fast-moving	Min	Symmetric	256 bits	High	Disk Encryption, secure communications, file encryption	Limited adoption
3DES	Slow-moving	Moderate	Symmetric	168 bits	Moderate	Legacy systems, payment processing, safe communication	Widely used

AES	Fast-moving	Min	Symmetric	128, 192, 256 bits	High	Bulk data encryption, secure communication, file encryption, wireless network security.	Widely used NIST Standard

3. Results and Discussion

The primary subjects of this study are the common symmetric cryptography technique of the AES representation technique and the asymmetrical technique of the RSA. It has received a lot of attention from experts who are studying the encryption of these two encryption techniques. Although the RSA and AES algorithms are widely used in the field of file encryption, there are still some questions regarding the effectiveness and security of encryption. This study completely utilizes the speed and high integrity of AES encryption, in addition to the potent key management features, built on the most modern RSA and AES methods simply during the execution of specific challenges[16-20]. After that, a hybrid AES/RSA encryption technique is presented and used to encrypt files. The advantages of this hybrid technique are evaluated in terms of how well encrypted and decrypted process.

3.1 Avalanche Effect

The term “avalanche effect” describes a desired property of cryptography in which a minor alteration to an input or key results in a large and unexpected change to the output. This property is necessary for providing the security of encryption algorithms, as it ensures that the encrypted output appears completely different when even a single bit of the input or key is modified. The more pronounced the avalanche effect, the greater the level of security provided by the encryption algorithm.

The avalanche effect prevents an attacker from gaining any useful information about the original input or key by observing the changes in the encrypted output. Even a slight modification in the plaintext or key should result in a completely different cipher text, making it extremely difficult for an adversary to detect patterns or derive any meaningful information about the original data[21-24]. This property is essential for protecting sensitive information and ensuring the confidentiality and integrity of encrypted communications.

To achieve a strong avalanche effect, cryptographic algorithms employ techniques such as substitution, permutation, diffusion, and confusion. These techniques distribute the changes caused by the modification of the input or key throughout the entire encryption process, creating a ripple effect that alters multiple bits in the resulting cipher text. By magnifying the impact of even a small change, the avalanche effect guarantees that encrypted data remains secure even if an attacker attempts to analyze or manipulate the cipher text.

3.2 Execution Time

Execution time refers to the amount of time taken by a program, process, or task to complete its operations. It is a critical metric in the field of computer science and software engineering as it directly impacts the performance and efficiency of systems. Execution time is affected by factors, including the complexity of the algorithm, the executing power of the hardware, the size of the input data, and the efficiency of the implementation.

Reducing execution time is often a desirable goal as it leads to faster and more responsive systems, improved user experience, and increased overall productivity. Developers and engineers employ various techniques to optimize execution time, such as algorithmic optimizations, parallel processing, caching mechanisms, and code profiling and tuning. By analyzing and optimizing execution time, organizations can achieve better system performance, optimize resource utilization, minimize response times, and meet performance requirements. Additionally, in critical domains such as real-time systems or high-frequency trading, where time constraints are crucial, minimizing execution time becomes even more crucial to ensure the timely and accurate processing of data. Therefore, understanding and optimizing execution time is vital in the development of efficient and high-performance software systems.

3.3 Experimental Setup

The Python programming language is employed for execution of the algorithm and Windows 11 operating system support is provided through the V S Code compilation tool. The experiment uses ".txt" filename files that are 7 kilobytes, 518 kilobytes, and 1338 kilobytes in size. In the Hybrid AES paradigm, fixed block size is used for both encryption and decryption, just like in AES. In this experiment, the same sample of data was encrypted and deciphered using three distinct methods. Since the technique used to aggregate the results of numerous runs can be applied and the result with a substantial mistake is omitted, the experiment's results are more trustworthy. Based from the findings, the differences in execution time are assessed for the RSA approach, the AES method, the

modified AES method, and the hybrid model. The scheduling of the three of them, in various file sizes, is as indicated in Tables 2, 3, and 4. Data that can be read and understood by humans, known as plaintext, is transformed into ciphertext, or information that cannot be read or understood by people, through the use of AES encryption. The AES ciphertext, which is the result of the encryption process, cannot be read until a private RSA key is used to decode it. The AES algorithm's design philosophy is based on Shanon's diffusion and confusion theory, and the plaintext of the key is used to calculate the algorithm's round count.

In the iteration of the round function, the reversible non-linear byte substitution is performed, which plays a confusing role, and then performs the linear line shift transformation, which has diffusion effect. Finally, the column mixing transformation and the key addition transformation are produced. The hybrid encryption algorithm exhibits encryption time characteristics like the AES algorithm, maintaining efficiency while providing enhanced security[25-30].

Analyze the experimental data and compare the three encryption times: The RSA algorithm's processing time approximately doubles with increasing file size. AES's growth rate is sluggish, and the encryption time scales with the size of the file. The combined encryption technique of RSA and Modified AES takes the same amount of time to encrypt data as AES does. When the file size is 7 kb, the hybrid technique performs 9.0 times better than the RSA method, but at 1338 kb, the efficiency has grown 65.5 times, and it keeps getting better as the file size grows. Table 2 displays the decryption times for the three techniques. The decryption duration maps of the three models are shown in Fig. 5.

Experimental studies show that the RSA and AES algorithms take longer to decode files as their sizes grow. The RSA algorithm has grown practically linearly, while the AES method has only slightly grown[31,32]. The hybrid encryption algorithm's decryption time is predictable and stays within a narrow range, comparable to the AES method. Large files noticeably benefit from the faster decryption speed as compared to the RSA technique. It has become an expanding fad. The hybrid method notably upgrades encryption achievement compared to the RSA method while encrypting and decrypting wide information, as seen by the comparison of encryption and decryption above. Dual-layer encryption makes decrypting the file more difficult as it completes, but the hybrid approach fixes the issues of an open AES algorithm key and insufficient verification.

Firstly, the program will check if there is any public key and private key is present or not. If not, it will create public key and private key for future use. Then re-run the program to Encrypt a file. Arguments in the command line will be like “Python [Name of the python program file] encrypt [Password by user] [file to be encrypt].” After passing these commands in command line or terminal a file “data.txt” is being created to store the password by user. When password is going to be stored in “data.txt” file it will be stored after encrypting with public key using RSA algorithm[33-37].

Results can be measured by doing experiments and here we have taken three tables to show experiments. Table 2 shows the execution time taken by the three algorithms which are AES, Modified AES, and RSA + Modified AES or Hybrid AES. Input file for encryption and decryption is of size 7kb in first experiment.

Table 2: Performance Matrix for file 7 kb

Algorithm	Encryption Time(sec)	Decryption Time (Sec)	Key Generation Time (sec)	File Size (kb)	Key size (bits)	No. of rounds
AES	0.077614	0.083019	0.0539351	7 kb	128 bits	10
Modified AES	0.113025	0.11807	0.0550125	7 kb	128 bits	20
Hybrid Model (modified AES + RSA)	0.15137	0.162919	0.1043599	7 kb	128 bits	20

Table 3: Performance Matrix for file size 518 kb

Algorithm	Encryption Time (sec)	Decryption Time (Sec)	Key Generation Time (sec)	File Size (kb)	Key size (bits)	No. of rounds
AES	1.888288	2.5259499	0.05501484	518	128	10
Modified AES	3.579293	4.9901378	0.056602716	518	128	20

Hybrid Model (modified AES+ RSA)	3.748927	5.216840	0.10221409	518	128	20

Table 4: Performance matrix for file size 1338 kb

Algorithm	Encryption Time(sec)	Decryption Time (Sec)	Key Generation Time (sec)	File Size (kb)	Key size (bits)
AES	4.736486	6.396630 5	0.05447912	1329	128
Modified AES	9.229382	13.0161609	0.05747294	1329	128
Hybrid Model (modified AES+RSA)	9.621156	13.4938261	0.13101506	1329	128

Table 3 shows the execution time taken by the three algorithms which are AES, Modified AES and RSA + Modified AES or Hybrid AES. Input file for encryption and decryption is of size 518kb in second experiment. Table 4 shows the execution time taken by the three algorithms which are AES, Modified AES and RSA + Modified AES or Hybrid AES. Input file for encryption and decryption is of size 1338kb in third experiment. The bar graph in Fig. 4, shows the execution time is also increasing to provide more security. It is shown that hybrid modified AES algorithm is more complex and harder to exploit.

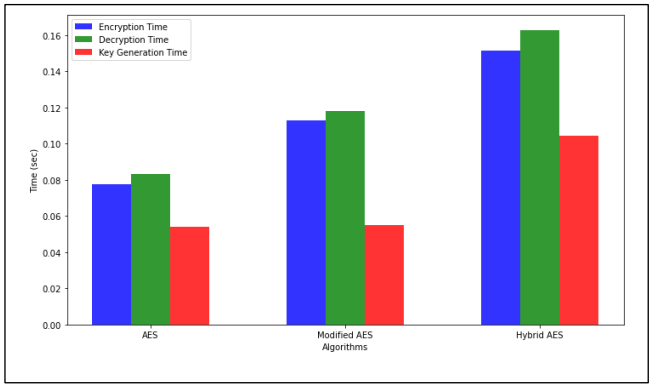

Fig. 4. Bar graph to show the differences in time taken.

4. Conclusion

In the current internet era, where terabytes of data are created every day and online transactions happen almost instantly, information security is challenging. Cryptography plays a major role in modern information security, making the internet a safer place. Using cryptography, one can render information unintelligible to an unauthorized party. This paper presents a hybrid model for file encryption using modified AES with masking and RSA algorithms. An analysis of the advantages, disadvantages, costs, and performance of each algorithm will provide valuable information for choosing cryptographic algorithms. In this paper, the widely used cryptographic algorithms DES, 3DES, AES, RSA, and blowfish are developed and their costs and performances are thoroughly examined to demonstrate an overall performance analysis. The AES algorithm, additionally referred to as the Rijndael algorithm, is a symmetrical block cipher that uses keys of lengths of 128, 192, and 256 bits to convert plain text into cipher text. The AES uses 128-bit symmetric, block cypher or single key.

Furthermore, the research also investigated the execution time of decryption of AES method, the RSA method, and the hybrid encryption method. The results revealed that the execution time of the RSA algorithm increases significantly with larger file sizes, following an almost linear development. In contrast, the AES algorithm demonstrates marginal increases in decryption time. Notably, the hybrid encryption algorithm exhibits reliable decryption times next to that of the AES algorithm, making it highly advantageous for handling large files. The improvement in decryption performance is particularly notable for large files, showcasing the potential impact of the hybrid algorithm. This research paper underscores the importance of file encryption in safeguarding personal and business data. It highlights the limitations of standalone encryption algorithms and presents a hybrid encryption technique that combines RSA and masked AES to address these challenges effectively. Through experimental evaluations, the hybrid approach demonstrates notable enhancements in encryption and decryption performance in comparison to the RSA technique alone. By striking a balance between efficiency and security, this hybrid encryption algorithm presents a promising solution for securing vast amounts of data.

Acknowledgements

The authors are highly grateful to Department of Computer Science & Information Technology, Siksha 'O' Anusandhan Deemed to be University for making this investigation successful.

Author contributions

Smita Rath, Sushree Bibhuprada B. Priyadarshini: Conceptualization, Methodology, Software, Field study **Deepak Kumar Patel, Prabhat Kumar Sahu, Nibedita Jagadev:** Data curation, Writing-Original draft preparation, Software, Validation., Field study **Monalisa Panda, Narayan Patra, Sipra Sahoo:** Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Milanov, E., The RSA algorithm. RSA laboratories, pp.1-11, 2009.

[2] Kalpana, P., & Singaraju, S., Data security in cloud computing using RSA algorithm. International Journal of research in computer and communication technology, IJRCCT, ISSN, pp.2278-5841, 2012.

[3] Saveetha, P., & Arumugam, S., "Study on Improvement in RSA Algorithm and its Implementation", International Journal of Computer & Communication Technology, Vol. 3(6), pp.78, 2012.

[4] Goshwe, N. Y., "Data encryption and decryption using RSA algorithm in a network environment", International Journal of Computer Science and Network Security (IJCSNS), Vol. 13(7), pp.9, 2013.

[5] Kota, C. M., & Aissi, C., Implementation of the RSA algorithm and its cryptanalysis. In 2002 GSW.

[6] Rahman, M. M., Saha, T. K., & Bhuiyan, M. A. A., "Implementation of RSA algorithm for speech data encryption and decryption", IJCSNS International Journal of Computer Science and Network Security, Vol. 12(3), pp.74-82, 2012.

[7] Devi, A., Sharma, A., & Rangra, A., "A review on DES, AES and blowfish for image encryption & decryption", International Journal of Computer Science and Information Technologies, Vol. 6(3), pp.3034-3036, 2015.

[8] *Grabbe, J. O., The DES algorithm illustrated, 2010.*

[9] Adhie, R. P., Hutama, Y., Ahmar, A. S., & Setiawan, M. I., "Implementation cryptography data encryption standard (DES) and triple data encryption standard (3DES) method in communication system based near field communication (NFC)", In Journal of Physics: Conference Series, Vol. 954, No. 1, p. 012009. IOP Publishing, 2018.

[10] Taghipour, M., Moghadam, A., Moghadam, N. S. B., & Shekardasht, B., "Implementation of Software-Efficient DES Algorithm" Advances in Networks, Vol. 3(1), pp.7-22, 2015.

[11] Saikumar, I., DES-Data Encryption Standard. International Research Journal of Engineering and Technology, Vol. 4(3), 2017.

[12] Berent, A., Advanced Encryption Standard by Example. Document available at URL http://www. networkdls. com/Articles/AESbyExample. pdf (April 1 2007) Accessed: June 2013.

[13] Gaj, K., & Chodowiec, P., Fast implementation and fair comparison of the final candidates for advanced encryption standard using field programmable gate arrays. In Topics in Cryptology—CT-RSA 2001: The Cryptographers' Track at RSA Conference 2001 San Francisco, CA, USA, April 8–12, 2001 Proceedings (pp.84-99). Springer Berlin Heidelberg, 2001.

[14] Stallings, W., Cryptography and network security principles and practices, 2006.

[15] Yenuguvanilanka, J., & Elkeelany, O., Performance evaluation of hardware models of Advanced Encryption Standard (AES) algorithm. In IEEE SoutheastCon 2008 pp.222-225. IEEE, 2008.

[16] Mohamed, A. A., & Madian, A. H., A Modified Rijndael Algorithm and it's Implementation using FPGA. In 2010 17th IEEE International Conference on Electronics, Circuits and Systems pp.335-338. IEEE, 2010.

[17] Pramstaller, N., Gurkaynak, F. K., Haene, S., Kaeslin, H., Felber, N., & Fichtner, W., Towards an AES crypto-chip resistant to differential power analysis. In Proceedings of the 30th European Solid-State Circuits Conference pp.307-310. IEEE, 2004.

[18] Jain, R., Jejurkar, R., Chopade, S., Vaidya, S., & Sanap, M., AES algorithm using 512 bit key implementation for secure communication. Int. J. Innov. Res. Comput. Commun. Eng, Vol. 2(3), pp.3516-3522, 2014.

[19] Selmane, N., Guilley, S., & Danger, J. L., Practical setup time violation attacks on AES. In 2008 Seventh European Dependable Computing Conference pp.91-96. IEEE, 2008.

[20] Benvenuto, C. J., Galois field in cryptography. University of Washington, Vol. 1(1), pp.1-11, 2012.

[21] Lee, H., Lee, K., & Shin, Y., Aes implementation and performance evaluation on 8-bit microcontrollers. arXiv preprint arXiv:0911.0482, 2009.

[22] Padate, R., & Patel, A., Encryption and decryption of text using AES algorithm. International Journal of Emerging Technology and Advanced Engineering,

Vol.4(5), pp.54-9, 2014.

[23] Wright, C. P., Dave, J., & Zadok, E., Cryptographic file systems performance: What you don't know can hurt you. In Second IEEE International Security in Storage Workshop pp.47-47. IEEE, 2003.

[24] Kretzschmar, U., Aes128–ac implementation for encryption and decryption. TI-White Paper, 2009.

[25] Deshpande, A. M., Deshpande, M. S., & Kayatanavar, D. N., FPGA implementation of AES encryption and decryption. In 2009 international conference on control, automation, communication and energy conservation pp.1-6. IEEE, 2009.

[26] Pletka, R., & Cachin, C., Cryptographic security for a high-performance distributed file system. In 24th IEEE Conference on Mass Storage Systems and Technologies (MSST 2007) pp.227-232. IEEE, 2007.

[27] Ors, S. B., Gurkaynak, F., Oswald, E., & Preneel, B., Power-analysis attack on an ASIC AES implementation. In International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. Vol. 2, pp.546-552. IEEE, 2004.

[28] Han, S. J., Oh, H. S., & Park, J., The improved data encryption standard (DES) algorithm. In Proceedings of ISSSTA'95 International Symposium on Spread Spectrum Techniques and Applications Vol. 3, pp.1310-1314. IEEE, 1996.

[29] Zhou, X., & Tang, X., Research and implementation of RSA algorithm for encryption and decryption. In Proceedings of 2011 6th international forum on strategic technology Vol. 2, pp.1118-1121. IEEE, 2011.

[30] Zhou, X., & Tang, X., Research and implementation of RSA algorithm for encryption and decryption. In Proceedings of 2011 6th international forum on strategic technology Vol. 2, pp.1118-1121. IEEE, 2011.

[31] Karthigaikumar, P., & Rasheed, S., Simulation of image encryption using AES algorithm. IJCA special issue on "computational science-new dimensions & perspectives" NCCSE, pp.166-172, 2011.

[32] Priyadarshini, S. B. B., Rath, S., Patel, S. M., Udgata, A., Mohanta, A., Ali, S. R., & Sahu, P. (2023). A Hybrid Random Image Generation Strategy (Hr-Igs) For Securing Plain Text Data In Networks. Journal of Theoretical and Applied Information Technology, 101(6).

[33] Dumbere, D. M., & Janwe, N. J., Video encryption using AES algorithm. In Second International Conference on Current Trends in Engineering and Technology-ICCTET 2014 pp. 332-337, IEEE, 2014.

[34] Heron, S., Advanced encryption standard (AES). Network Security, 2009(12), pp.8-12, 2009.

[35] Radhadevi, P., & Kalpana, P. (2012). Secure image encryption using AES. International Journal of Research in Engineering and Technology, Vol.1(2), pp.115-117, 2012.

[36] Abdullah, A. M., Advanced encryption standard (AES) algorithm to encrypt and decrypt data. Cryptography and Network Security, 16, pp.1-11, 2017.

[37] Zeghid, M., Machhout, M., Khriji, L., Baganne, A., & Tourki, R. (2007). A modified AES based algorithm for image encryption. International Journal of Computer and Information Engineering, Vol. 1(3), pp.745-750, 2007.