University of Ibadan
Journal of Science and Logics in ICT Research

Hybrid Encryption System with Initialization Vector for Secure Data Transmission

[1]✉Chamo Y. Y., [2]Okunade O. A., [3]Dada E. G., [4]Ajao J. F., and [5]Ezeanya C. U.

[1,2 &5]Department of Computer Science, Faculty of Computing, National Open University of Nigeria, Abuja.
[3]Department of Computer Science, Faculty of Physical Sciences, University of Maiduguri, Nigeria.
[4]Department of Computer Science, Faculty of Information and Communication Technology, Kwara State University, Malete.

[1]*yusufchamo@gmail.com;* [2]*aokunade@noun.edu.ng;* [3]*gbengadada@unimaid.edu.ng;* [4]*jumoke.ajao@kwasu.edu.ng;* [5] *cezeanya@noun.edu.ng*

Abstract

A critical challenge facing various industries is how to ensure the security of sensitive data. To transmit sensitive data over insecure channels, secure key management of encryption keys and generating a well-secured ciphertext have become paramount. To address this challenge, this paper provides a Hybrid Encryption System that leverages the strengths of asymmetric and symmetric encryption in terms of key management, encryption speed, and overall usability. The initialization vector ensures the uniqueness of the ciphertext produced. During encryption, a recipient generates two RSA keys (public and private) and then proceeds to share the public key with the sender. A pseudo-random number generator (PRNG) is used to create the initialization vector (IV) that is used alongside the AES key to encrypt the data file. The AES key is then encrypted using the recipients' public key and all these are done during one execution stage. At decryption, the recipient will receive three files namely; an encrypted data file, an encrypted AES key, and IV. The AES key is decrypted using the recipients' private key before decrypting the data file and all these are also done at one execution stage. The Hybrid Encryption System was evaluated against the AES and RSA algorithms. According to the results obtained, the total execution time indicated that the proposed hybrid system was considerably faster than RSA while AES was faster than the proposed hybrid system. The hybrid system also provides the maximum level of data security due to the uniqueness of the ciphertext it produces.

Keywords: *Data Security; Encryption; Asymmetric; Symmetric; Advanced Encryption Standard (AES); Rivest-Shamir-Adleman (RSA)*

1. Introduction

In today's digital age, the need for secure data storage and communication is paramount. With the increasing threat of data breaches and unauthorized access to sensitive information, it has become essential to employ robust encryption techniques to protect valuable data [2]. In recent years, data breaches have become increasingly common, resulting in the loss of sensitive information and damage to organizational reputations. Sending data and storing data through electronic media requires a process that can guarantee the security and integrity of the data sent and for that, we need a process known as cryptography for the encryption of the data sent [4][23].

Encryption is a technique commonly used to protect sensitive data from unauthorized access [6]. File encryption is a means of protecting files so that they can only be accessed by authorized individuals [8]. The rapid growth of digital information and the widespread use of the internet have made file encryption and decryption crucial in order to ensure the confidentiality and integrity of sensitive data [19].Traditional encryption algorithms, such as RSA and AES while having proven to be effective in securing data often lack integration, requiring users to employ separate tools or

Chamo Y. Y., Okunade O. A., Dada E. G., Ajao J. F. and Ezeanya C. U (2024). .Hybrid Encryption System with Initialization Vector for Secure Data Transmission, *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 12 No. 1, pp. 98 - 109

techniques for different encryption algorithms. The absence of a unified system that seamlessly combines the strengths of both asymmetric and symmetric encryption poses challenges in terms of key management, interoperability, and overall usability [29]. The RSA and AES algorithms, while proven to be effective are typically implemented as stand-alone solutions. By combining these two algorithms in a hybrid approach, the system can leverage the benefits of both asymmetric key and symmetric key encryption which provides a comprehensive and efficient solution for data encryption and decryption.

The RSA algorithm, named after its creators Rivest, Shamir, and Adleman is an asymmetric encryption algorithm that is used widely and relies on the mathematical properties of prime numbers for secure key generation, encryption, and decryption [10]. It provides a secure mechanism for exchanging encryption keys between parties without requiring a shared secret. RSA encryption is carried out by using two keys i.e. public and private keys; for encryption and decryption respectively [31]. The perplexity of factoring large prime numbers makes RSA a strong choice for secure communications [30].

The AES algorithm also known as the Advanced Encryption Standard, is a symmetric encryption algorithm well known for its strength and speed in protecting data [13], it uses a secret key for both encryption and decryption [27]. AES is widely regarded as a highly secure and efficient encryption algorithm, making it suitable for various applications, including file encryption [12]. It operates on fixed-size blocks of data and employs a substitution-permutation network (SPN) structure [34].

Hybrid encryption is a mode of encryption that merges two or more encryption systems. It incorporates a combination of asymmetric and symmetric encryption to benefit from the strengths of each form of encryption [9] [11]. This hybrid approach strikes a balance between security and performance, providing a comprehensive solution for file encryption and decryption in the developed system.

Initialization vector (IV) has been added as an extra layer of security for the hybrid encryption system. It is an arbitrary number that is used in combination with a secret key to encrypt data. This number is sometimes referred to as a nonce, or “number occurring once,” as an encryption program uses it only once per session [25]. IV is created from a secure pseudo-random number generator (PRNG) and it is used to avoid repetition during the encryption process, this makes it impossible for hackers that use dictionary attacks to decrypt the encrypted data by discovering a pattern [3] [5].

2.0 Related Works

Francis &Monoth [9] carried out an analysis of Hybrid Cryptographic approaches for Information Security. According to the study, practically every cryptosystem is susceptible to various kinds of cryptographic attacks therefore, it is necessary to use a combination of cryptographic algorithms to protect the system. As a result, there is increased computational complexity.

Santoso, *et al.* [26] carried out research that incorporated two cryptographic algorithms namely AES and Twofish with the Hash function SHA 256 generating 256 bits key. They indicated that combining at least two algorithms for cryptography will enhance data security. However, their work combined two algorithms that are both symmetric encryption algorithms and as such has not addressed the limitation of symmetric encryption.

Murad and Rahouma [20] carried out research that proposed a comparative study for two-tier versus three-tier hybrid cryptographic models applied to secure data on the cloud. Python simulation was used to practically implement the studied hybrid models and an analysis of the performance was carried out in terms of encryption and decryption time, average throughput, and efficiency using relatively larger data files. Their study showed that the two-tier hybrid model is more efficient than the three-tier hybrid model. A limitation of their study was in the performance of the three-tier hybrid model although they indicated it should be utilized in applications where security is the main concern regardless of the performance.

Mamun, *et al.* [18] presented a paper that introduces the estimation and ensuring security of encrypted information by hybrid AES and RSA algorithm with third-party confirmation.

The privacy of messages and the authenticity of communicating parties are supported in this secured authentication system. However, due to third-party confirmation, authenticating the correct beneficiary takes a great deal of time and this can be considered as a significant impediment of this work.

Li and Zhang [16] proposed using RSA and AES algorithms for securing file transfers in IoT networks. They implemented the system in a resource-constrained environment. The study highlighted the effectiveness of the hybrid encryption system in securing IoT communication with limited computational resources. The combination of encryption algorithms in an IoT network increases the computational power to run it and this is a downside to this work.

Priya and Saradha [24] proposed a hybrid cryptographic technique in a cloud environment to improve the security rate and provide privacy preservation of medical data in the cloud environment. This work mainly uses the AES algorithm, Honeypot algorithm, SHA 3 hashing, and OTP in the cloud environment to implement hybrid cryptographic schemes. The proposed work enhances the security of data to a great extent which makes it difficult for intruders to attack the system. However, there are certain limitations such as computational overhead and the complexity of implementing and managing multiple encryption schemes, particularly in diverse cloud environments

Al-Bayati [1] researched to apply a hybrid encryption method that combines two encryption algorithms namely AES and RSA as well as the introduction of the use of quantum encryption BB84 to the three-hybrid encryption method that was created in the research. A hybrid quantum key distribution solution was applied to the hybrid encryption method to verify data integrity before the encryption process started. A limitation of this work is quantum computers are very expensive and are still in their early stages. They are also not widely accessible.

2.1 AES Algorithm in File Encryption

The AES algorithm, also known as the Advanced Encryption Standard, is a widely adopted symmetric encryption algorithm for secure data transmission and storage [22]. It has been rigorously analyzed and it is considered to be highly secure against cryptographic attacks [21]. Recent studies have explored the application of the AES algorithm in file encryption and highlighted its effectiveness in ensuring data confidentiality and integrity.

One notable study by Li *et al.* [15] investigated the performance and security of the AES algorithm in file encryption applications. The researchers conducted a comprehensive analysis of different AES key sizes and encryption modes. Their findings emphasized the importance of using larger key sizes, such as 256-bit, to enhance the security of file encryption. The study also evaluated the impact of different encryption modes, such as Cipher Block Chaining (CBC) and Galois/Counter Mode (GCM), on file encryption performance and security. The results provided valuable insights for selecting appropriate AES configurations for file encryption systems.

In another study, Wang *et al.* [32] proposed an optimized implementation of the AES algorithm for efficient file encryption. They explored parallel computing techniques to accelerate the encryption and decryption processes for large files. The study demonstrated the potential of leveraging parallelism to improve the performance of AES-based file encryption systems. The findings highlighted the importance of considering performance optimizations when implementing AES in file encryption applications.

Furthermore, the research conducted by Chen *et al.* [7] focused on enhancing the security of AES-based file encryption through key management techniques. They proposed a key management scheme that dynamically updated the AES encryption key based on user behaviour and file characteristics. The study emphasized the significance of robust key management practices in maintaining the security of AES-based file encryption systems. The proposed scheme provided an additional layer of protection against unauthorized access and key-related attacks.

Lastly, the work by Kumar *et al.* [14] investigated the integration of AES encryption with secure cloud storage. They proposed a scheme that combined AES encryption with attribute-based access control (ABAC) to ensure the confidentiality and access control of files stored in the cloud. The study highlighted the compatibility of AES with other security mechanisms and its effectiveness in securing files in cloud environments.

In summary, recent research studies have examined the application, performance, and security aspects of the AES algorithm in file encryption. These studies have provided insights into key size selection, performance optimizations, key management techniques, and the integration of AES with other security mechanisms.

2.2 RSA Algorithm in File Encryption

The RSA algorithm is an asymmetric encryption algorithm that is used widely for data protection and secure communication. It has been extensively studied and applied in various encryption systems, including file encryption. A recent study by Wang *et al.* [33] investigated the performance and security of the RSA algorithm in file encryption applications. They conducted a comprehensive analysis of the RSA encryption process, focusing on key size selection, encryption speed, and resistance against attacks. The study provided valuable insights into the practical considerations and trade-offs when utilizing the RSA algorithm for file encryption. Their findings highlighted the importance of selecting appropriate key sizes to ensure strong security while maintaining acceptable performance levels.

Furthermore, the research conducted by Liu *et al.* [17] explored the application of the RSA algorithm in secure file encryption for cloud storage. They proposed a scheme that combined RSA encryption with homomorphic encryption techniques to achieve both confidentiality and integrity of the stored files. The study demonstrated the effectiveness of RSA-based file encryption in protecting sensitive data in cloud environments. It emphasized the importance of proper key management and encryption processes to ensure the security of files stored in the cloud.

Lastly, the work by Zhao *et al.* [35] focused on enhancing the efficiency of the RSA algorithm in file encryption. They proposed an optimized implementation of RSA encryption using parallel computing techniques, significantly reducing the encryption time for large files. The study highlighted the potential of optimizing the RSA algorithm to improve the performance of file encryption systems. Their findings provided valuable insights for developing efficient RSA-based file encryption solutions.

In summary, recent research studies have explored the application and performance of the RSA algorithm in file encryption. These studies have provided insights into the optimization techniques, key size selection, performance considerations, and security aspects of using the RSA algorithm in file encryption systems.

3.0 Methodology

The Python programming language and the Django web framework were chosen for the development of the system due to their versatility, robustness, and extensive community support. Python's readability and simplicitymake it an ideal choice for implementing complex encryption algorithms, while Django's high-level web development framework has the necessary tools and functionalities for building secure web applications. It provides a unified and user-friendly platform that integrates encryption, and decryption seamlessly.

The combination of these technologies ensures that the hybrid encryption system can be developed efficiently and effectively, with a focus on usability, security, and performance.

3.1 Dataset Description

The data compared and analyzed in this paper were six text files of varying sizes which are listed below:

i. 1.04 MB
ii. 3.14 MB
iii. 6.70 MB
iv. 9.84 MB
v. 12.9 MB
vi. 19.2 MB

Several metrics have also been identified to evaluate the performance of the Proposed Hybrid System.

3.2 Performance Metrics

Performance evaluation was carried out on the following algorithms: AES, RSA and Proposed Hybrid system. The performance metrics used are encryption time, decryption time and throughput.

i. **Encryption time**
The time taken to transform plaintext to ciphertext is known as encryption time. It indicates the speed of the encryption process.

ii. **Decryption time**
The time taken to transform ciphertext to plaintext is known as decryption time. It indicates the speed of the decryption process.

iii. **Throughput**
This is the number of data successfully transported from one location to another via a network system in a certain period. This is calculated as the total plaintext encrypted in Kbytes divided by the encryption time in milliseconds (Singh et al., 2012). The unit of throughput is MB/Sec. A high throughput signifies that the encryption procedure takes less time.

3.3 Proposed Hybrid System

The AES and RSA encryption techniques are used in hybrid to secure the transmission of data files. During encryption, the recipient generates two RSA keys (public and private) and then proceeds to share the public key with the sender. The AES algorithm is the primary mode of encryption in the Hybrid Encryption System. A pseudo-random number generator (PRNG) generates the initialization vector (IV) that is used to encrypt the data file together with the AES key. The recipient’s public key is used to encrypt the AES key and all these are done during one execution stage and not separately.

At decryption, the recipient receives three files which are; the encrypted data file, encrypted AES key, and IV. The recipient will then proceed to use their private key to decrypt the AES key before decrypting the encrypted data file using the AES key and the IV. All these are also done during one execution stage.

3.4 Architecture of Proposed Hybrid System

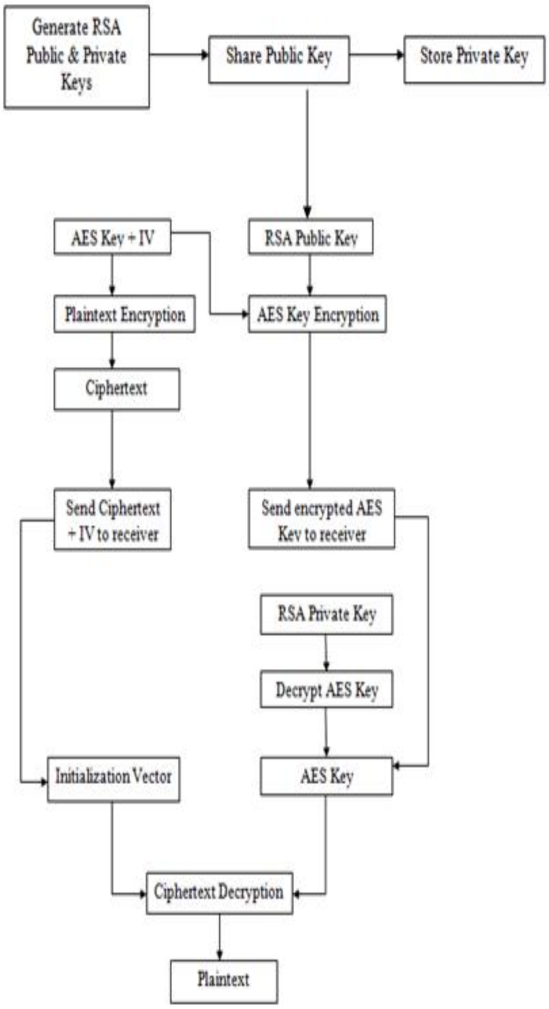


Figure 1: Block Diagram of the Proposed Hybrid System

3.5 Flowchart of the Proposed Hybrid System

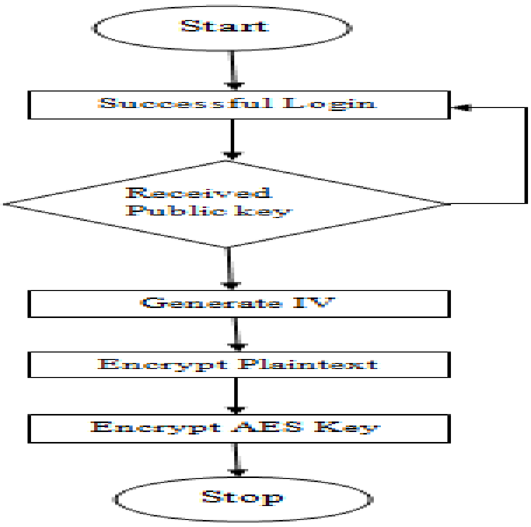


Figure 2: Encryption Flowchart

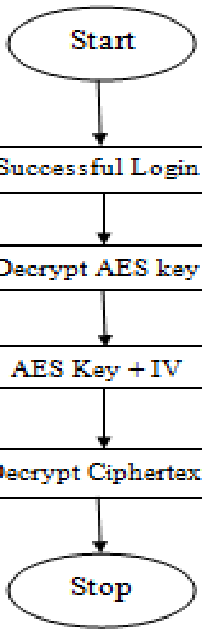


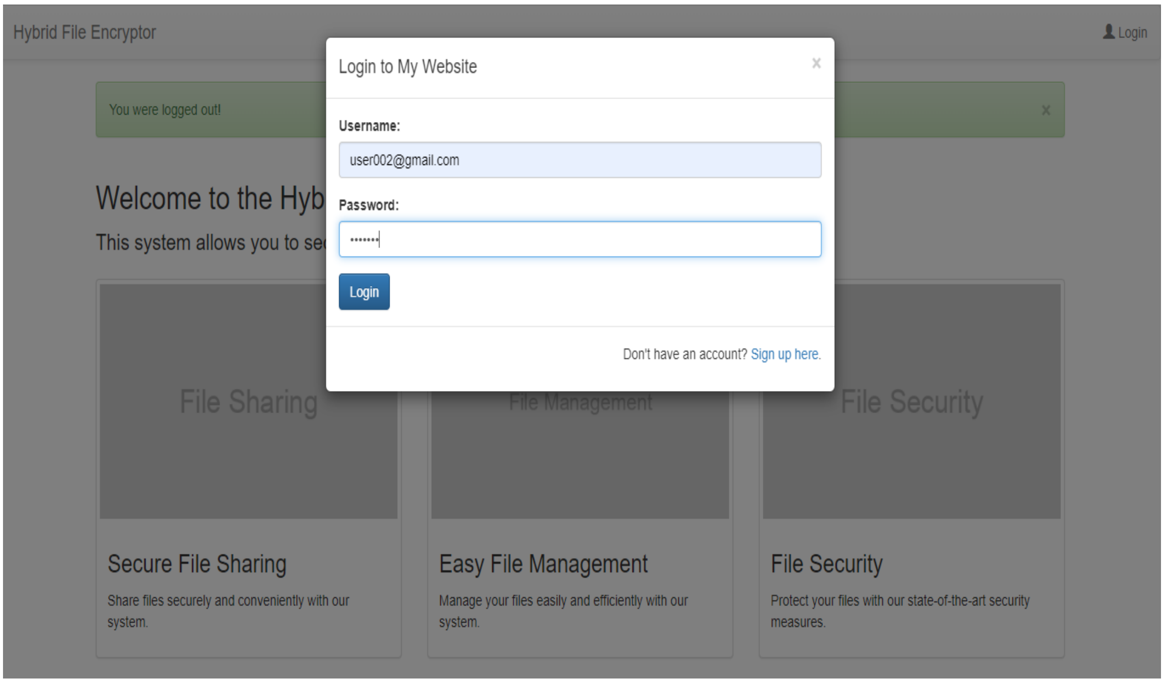
Figure 3: Decryption Flowchart

4.0 Results

Data text files of varying sizes were used in calculating the encryption, decryption time, and throughput of the three different algorithms used, and all the results were discussed, compared, and analyzed.

4.1 Graphical User Interface

Figure 4 depicts the login and sign-up interface to use the system. A registered user is granted access to the system through the interface.



You were logged out!

Welcome to the Hyb

This system allows you to se

Username: user002@gmail.com

Password: •••••••

Don't have an account? Sign up here.

Secure File Sharing

Share files securely and conveniently with our system.

Easy File Management

Manage your files easily and efficiently with our system.

File Security

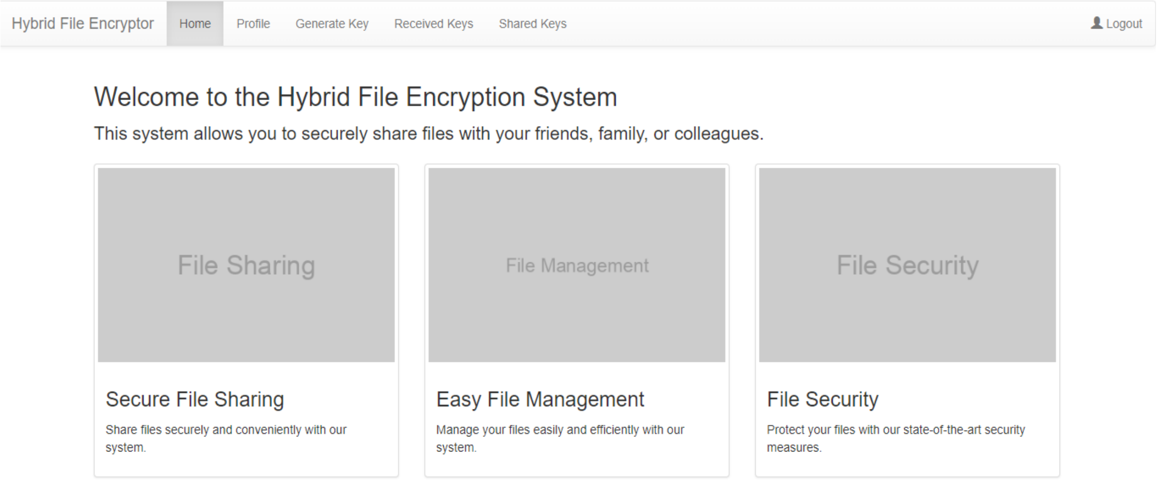
Protect your files with our state-of-the-art security measures.

Figure 4: Login and Signup Page

Figure 5 depicts the system homepage. A user can choose from the different options available depending on what they want to do.

Figure 6 depicts key generation interface. After a user successfully generates the RSA keys for AES key encryption here, they can also further share the public key to another user from here.

Figure 7 depicts key sharing interface with another user. The email of the user that the key will be shared with is entered here.



Welcome to the Hybrid File Encryption System

This system allows you to securely share files with your friends, family, or colleagues.

Secure File Sharing

Share files securely and conveniently with our system.

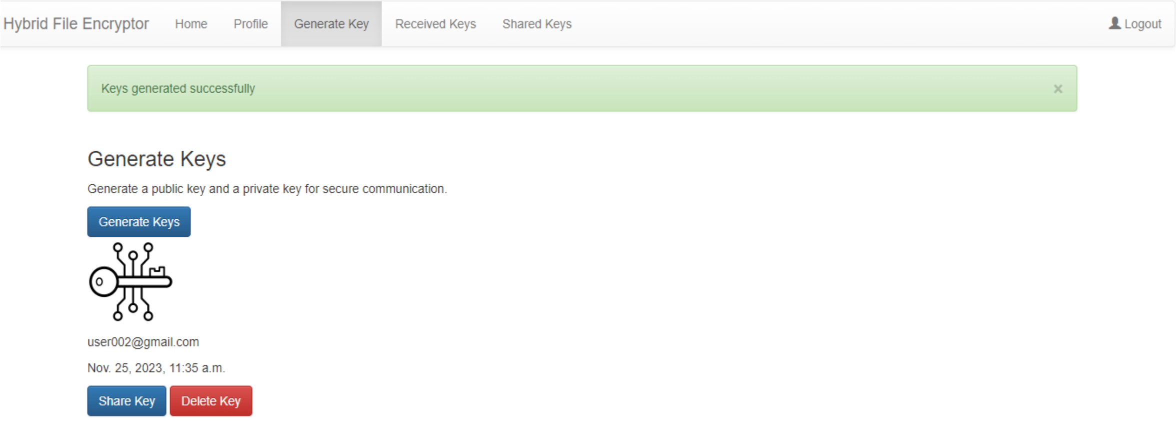
Easy File Management

Manage your files easily and efficiently with our system.

File Security

Protect your files with our state-of-the-art security measures.

Figure 5: Proposed Hybrid System Homepage



Keys generated successfully

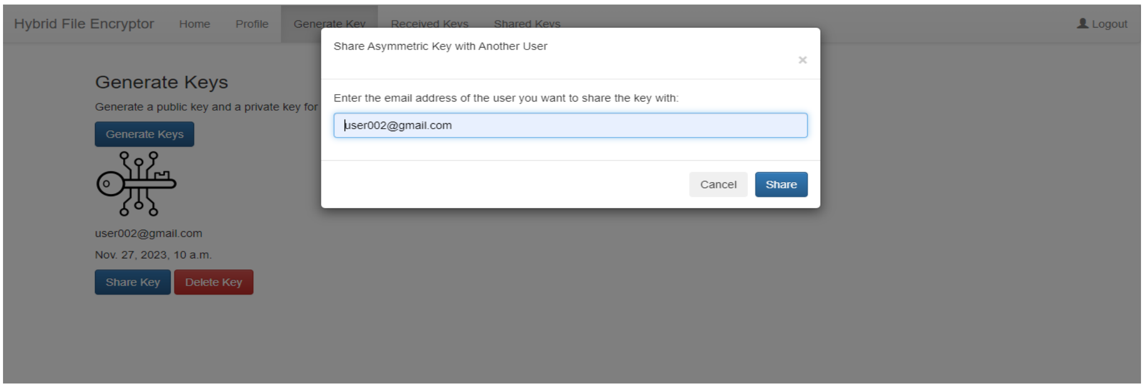
Generate Keys

Generate a public key and a private key for secure communication.

user002@gmail.com

Nov. 25, 2023, 11:35 a.m.

Figure 6: Key Generation Interface



Generate Keys

Generate a public key and a private key for

user002@gmail.com

Nov. 27, 2023, 10 a.m.

Share Asymmetric Key with Another User

Enter the email address of the user you want to share the key with:

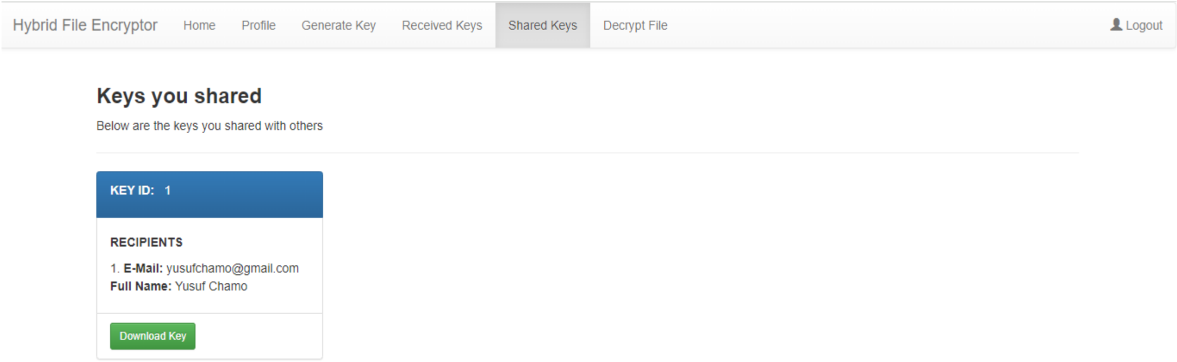
user002@gmail.com

Figure 7: Key Sharing With Another User

Figure 8 depicts the shared keys interface, where a user can see all the keys they have shared with other users and also download the keys they generated i.e. public and private keys.

Figure 9 depicts the received keys interface, where a user can see all the keys that were shared with them i.e. sent from other users.

Figure 10 depicts the file encryption interface, where a user selects the file they want to encrypt along with the public key that was shared with them.



Keys you shared

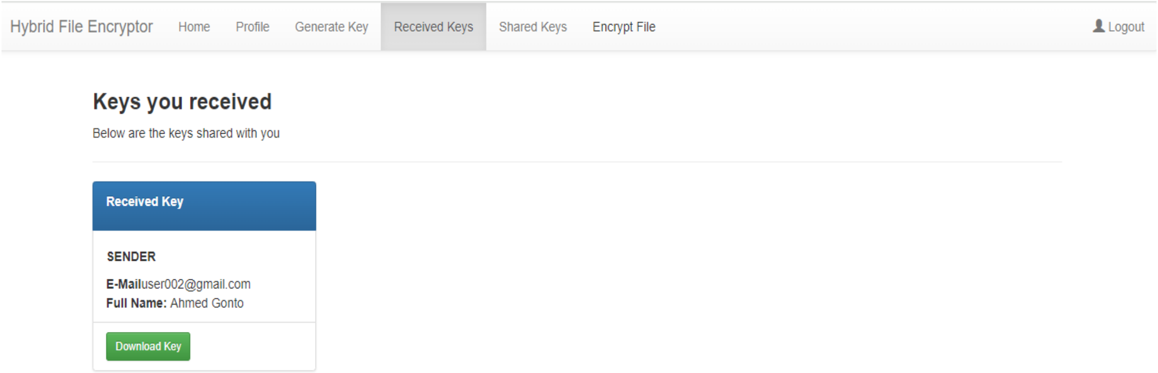
Below are the keys you shared with others

KEY ID: 1

RECIPIENTS

1. **E-Mail:** yusufchamo@gmail.com
Full Name: Yusuf Chamo

Figure 8: Shared Keys Interface



Keys you received

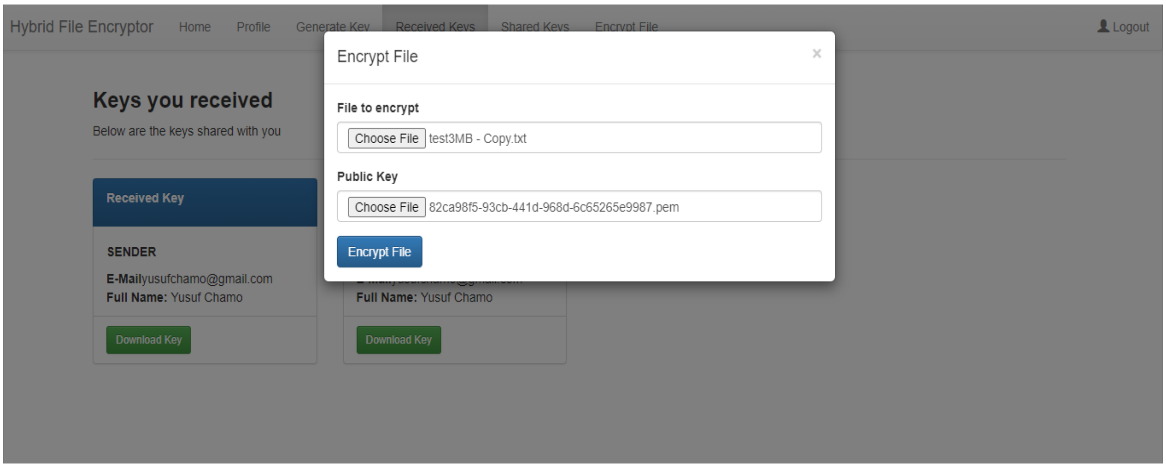
Below are the keys shared with you

Received Key

SENDER

E-Mailuser002@gmail.com
Full Name: Ahmed Gonto

Figure 9: Received Keys Interface



Encrypt File

File to encrypt: test3MB - Copy.txt

Public Key: 82ca98f5-93cb-441d-968d-6c65265e9987.pem

Keys you received

Below are the keys shared with you

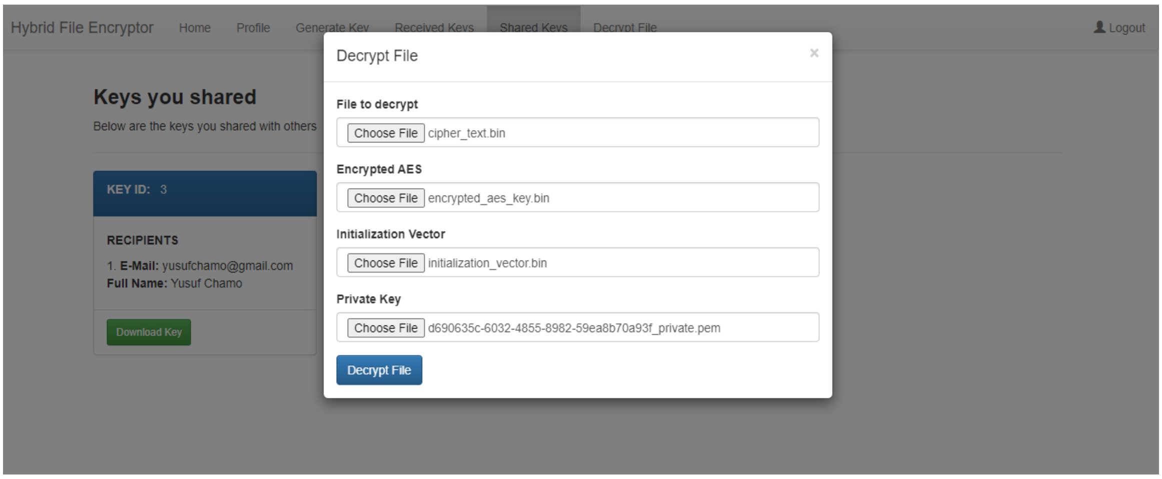
Received Key

SENDER

E-Mailyusufchamo@gmail.com
Full Name: Yusuf Chamo

Full Name: Yusuf Chamo

Figure 10: File Encryption Interface



Keys you shared

Below are the keys you shared with others

KEY ID: 3

RECIPIENTS

1. **E-Mail:** yusufchamo@gmail.com
Full Name: Yusuf Chamo

Decrypt File

File to decrypt: cipher_text.bin

Encrypted AES: encrypted_aes_key.bin

Initialization Vector: initialization_vector.bin

Private Key: d690635c-6032-4855-8982-59ea8b70a93f_private.pem

Figure 11: File Decryption Interface

Figure 11 depicts the file decryption interface, where a user selects the encrypted file they want to decrypt along with the IV they received. The user also adds their private key to decrypt the AES key.

4.2 Performance Evaluation

i) The AES, RSA, and Proposed Hybrid Algorithm were used to encrypt and decrypt the same text files, and the time taken for each algorithm in seconds were compared and evaluated.

ii) The total encryption and decryption time of each algorithm was also calculated.

iii) The throughput of each algorithm was also calculated and the results obtained are as follows:

1. AES: 157.67
2. RSA: 31.66
3. **3.** Hybrid: 107.47

Table 1 shows the processing time obtained from the encryption and decryption processes of the proposed hybrid system in comparison to the other algorithms used.

Table 1: Execution Time for each Algorithm

File size (MB)	AES Encryption (seconds)	AES Decryption (seconds)	RSA Encryption (seconds)	RSA Decryption (seconds)	Hybrid Encryption (seconds)	Hybrid Decryption (seconds)
1.04	0.013	0.02	0.02	0.53	0.019	0.1
3.14	0.04	0.06	0.09	0.77	0.06	0.15
6.70	0.12	0.15	0.18	1.11	0.18	0.24
9.84	0.17	0.20	0.27	1.59	0.23	0.30
12.9	0.24	0.28	0.39	1.96	0.27	0.46
19.2	0.35	0.37	0.52	2.58	0.38	0.56

Table 2: Total Execution Time for each Algorithm

File size (MB)	AES total time	RSA total time	Hybrid total time
1.04	0.03	0.55	0.119
3.14	0.10	0.86	0.21
6.70	0.27	1.29	0.42
9.84	0.37	1.86	0.53
12.9	0.52	2.35	0.73
19.2	0.72	3.10	0.94

Table 2 shows the total execution time i.e. encryption time + decryption time of the proposed hybrid system in comparison to the other algorithms used. The proposed hybrid system is faster than the RSA and slightly slower than the AES.

Figure 12 depicts how each encryption algorithm performs against each other. A high throughput indicates that the encryption process took a lesser time as shown in Figure 13.

This shows that the AES has the highest throughput followed by the proposed hybrid system and then the RSA which has the least throughput

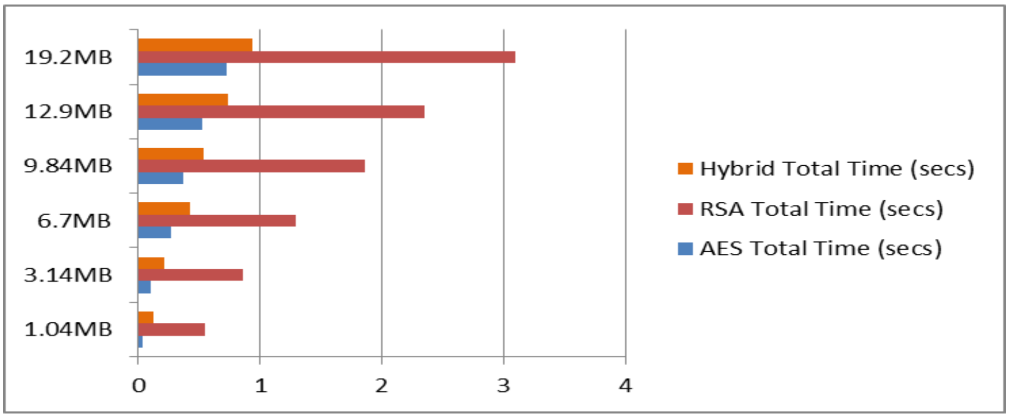


Figure 12: Total Execution Time for each Algorithm

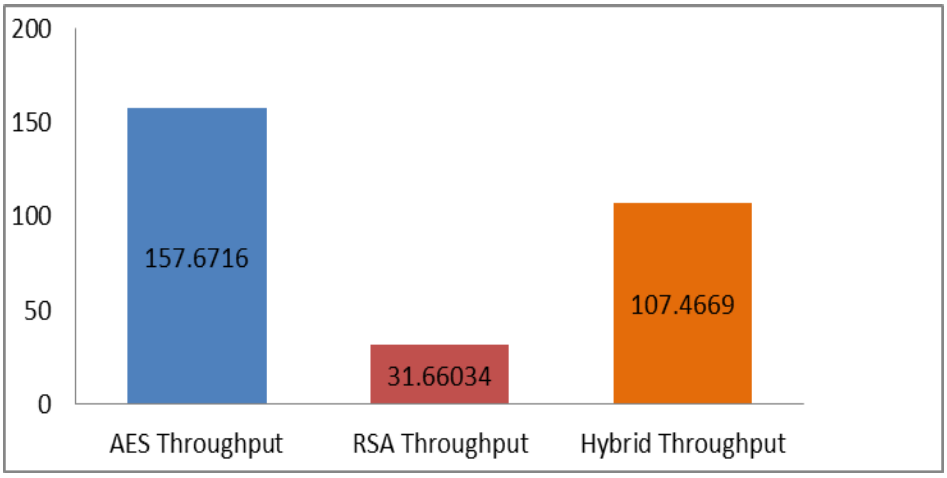


Figure 13: Throughput of each Algorithm

4.3 Discussion

According to the results obtained as shown in Figure 12, the total execution time showed that the proposed hybrid system was considerably faster than RSA while AES was faster than the proposed hybrid system. The proposed hybrid system provided the maximum level of data security due to the uniqueness of the ciphertext it produces and it is more reliable because of the secure key exchange and AES key encryption it provides. The Hybrid Encryption System demonstrated high levels of security and integrity in protecting sensitive data. The RSA algorithm ensured secure key exchange and encryption of the AES key, while the AES algorithm provided strong symmetric encryption for efficient and fast data processing. The initialization vector (IV) proved to be an extra security layer as it ensures the uniqueness and randomness of the ciphertext produced and secures it against dictionary attacks.

In comparison to the work of Santoso, *et al.* [26] the proposed hybrid system presented in this paper offers secure key exchange which is an added security measure whereas theirs is a hybrid symmetric encryption and does not offer a secure key exchange. When also compared to the work of Murad &Rahouma [20] their hybrid 2-tier AES-RSA model was the fastest and also had the highest throughput. This shows that the AES-RSA hybrid model is the most efficient. This efficiency is crucial for ensuring timely and seamless file operations and also maintaining a maximum level of security.

5.0 Conclusion and Future Work

This paper presents a Hybrid Encryption System with an Initialization Vector for secure data transmission that provides high level of security in protecting sensitive data. The performance evaluation of the hybrid encryption system revealed the system's efficient performance, with minimal overhead and fast processing times. The Hybrid Encryption System leverages the strengths of asymmetric and symmetric encryption in terms of key management, encryption speed, and overall usability.

The evolving nature of cybersecurity threats makes continuous monitoring and updating of the system's security measures to be a top priority. Regular security audits, vulnerability assessments, and best practices are essential for maintaining the system's resilience.

Further research can be carried out by making use of other encryption algorithms and carrying out performance evaluation on the developed model.

References

[1] Al-Bayati, A.S. (2023). Enhancing Performance of Hybrid AES, RSA and Quantum Encryption Algorithm. *Anglia Ruskin Research Online (ARRO)*, 30-Aug-2023. [Online]. Available: https://hdl.handle.net/10779/aru.23768127.v1. [Accessed: 18-May-2024]

[2] Awad., A.I., & Fairhurst, M. (2018). Information Security - Foundations, Technologies and Applications. *Institution of Engineering and Technology (The IET).* ISBN: 978-1-84919-974-2.

[3] Awati, R. (2022). Initialization Vector. [Online]. Available: *https://www.techtarget.com/whatis/definition/initialization-vector-IV* [Accessed: 13-Mar-2024]

[4] Azizah, L, N. (2020). How Can Playfair Cipher Secure Data? *Proceedings on the International Conference on Science and Engineering.* Vol. 3, 273 -277. doi: 10.14421/icse.v3.512

[5] Baeldung (2022). Initialization Vector for Encryption. [Online]. Available: *https://www.baeldung.com/java-encryption-iv*[Accessed: 13-Mar-2024]

[6] Burnett, M. (2004). Hacking the code, *Syngress, 1st Edition*. ISBN: 978-1-932266-65-8.

[7] Chen, X., Li, D., Huang, C., & Ma, C. (2019). A Key Management Scheme for Enhancing the Security of AES-based File Encryption. *Security and Communication Networks*, 1-9.

[8] Eludire, A.A., Okesola, J.O., &Osang, F.B. (2022). CIT 855: Advanced Cyber security.ISBN: 978-058-557-5. *National Open University of Nigeria (NOUN) Course Material.*

[9] Francis, N., &Monoth, T. (2018). An Analysis of Hybrid Cryptographic Approaches for Information Security. *International Journal of Applied Engineering Research.* 13(3).

[10] Franklin, R. (2022). AES vs. RSA Encryption: What Are the Differences? [Online]. Available:*https://www.precisely.com/blog/data-security/aes-vs-rsa-encryption-differences*[Accessed: 23-Nov-2023]

[11] Gupta, P., & Sharma, A. (2019). Hybrid Encryption Technique using RSA and AES Algorithms for Secure Transmission of Medical Data. *Journal of Advanced Research*

in Dynamical and Control Systems. 11(6), 2687-2694.

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

[13] Jena, B.K., (2023, February 9). *What is AES encryption and how does it work?* https://www.simplilearn.com/tutorials/cryptography-tutorial/aes-encryption

[14] Kumar, A., Goyal, S., & Gupta, A. (2021). Attribute-Based Secure Data Storage in Cloud Using AES Encryption. *International Journal of Cloud Computing.* 10(3/4), 265-279.

[15] Li, Q., Tang, S., Zhang, S., & Gong, Q. (2020). Performance and Security Analysis of AES Algorithm in File Encryption. *Journal of Physics: Conference Series,* 1663(1), 012014.

[16] Li, Z., & Zhang, L. (2021). Hybrid Encryption Algorithm Based on RSA and AES in IoT Networks. *Journal of Physics: Conference Series,* 1883(1), 012095.

[17] Liu, Y., Liu, C., & Zhang, Y. (2021). Secure File Encryption for Cloud Storage Based on RSA Algorithm and Homomorphic Encryption. *Journal of Physics: Conference Series,* 1822(1), 012066.

[18] Mamun, S.A., Mahmood, M.A., & Amin, M.A. (2021). Ensuring Security of Encrypted Information by Hybrid AES and RSA algorithm with third-party confirmation. *5th International Conference on Intelligent Computing and Control Systems (ICICCS).* 337-343, 2021. doi: 10.1109/ICICCS51141.2021.9432174

[19] Mattord, H. & Whitman, M. (2018). Management of Information Security. *6th Edition, Cengage Learning.* ISBN: 9781337405713.

[20] Murad, S.H., &Rahouma, K.H. (2021). Implementation and Performance Analysis of Hybrid Cryptographic Schemes Applied in Cloud Computing Environment. *Procedia Computer Science.* 194 (19), 165 – 172. doi: 10.1016/j.procs.2021.10.070

[21] Mushtaq, M.F., Jamel, S., Disina, A.H., Pindar, Z.A., Shakir, N.S.A., Deris, M.M. (2017). A Survey on the Cryptographic Encryption Algorithms. *International Journal of Advanced Computer Science and Applications (IJACSA).* 8 (11). doi: 10.14569/IJACSA.2017.081141

[22] Mustafeez, A.Z., (2023). What is the AES algorithm? [Online]. Available: https://www.educative.io/answers/what-is-the-aes-algorithm [Accessed: 24-Nov-2023]

[23] Muttaqin, K. &Rahmadoni, J., (2020). Analysis and Design Of File Security System AES (Advanced Encryption Standard) Cryptography Based. *Journal of Applied Engineering and Technological Science.* 1(2), 113-123. doi: 10.37385/jaets.v1i2.78

[24] Priya, A., &Saradha, S. (2021). Implementation of Hybrid cryptographic schemes in a cloud environment for enhanced medical data security. *International Journal of Nonlinear Analysis and Applications.* No.2, 1785-1800. doi: 10.22075/IJNAA.2021.5316

[25] Rouse, M. (2011). Initialization Vector. [Online]. Available: *https://www.techopedia.com/definition/26858/initialization-vector*[Accessed: 15-Dec-2023]

[26] Santoso, K.I., Muin, M.A., & Mahmudi, M.A. (2020). Implementation of AES cryptography and Twofish Hybrid algorithms for cloud. *Journal of Physics: Conference Series.* 1517(2020) 012099. doi: 10.108/1742-6596/1517/1/012099

[27] Semwal, P., & Sharma, M.K. (2017). Comparative Study of Different Cryptographic Algorithms for Data Security in Cloud Computing. *International Journal on Emerging Technologies (Special Issue NCETST-2017).* 8(1): 746-750.

[28] Singh, G., Singla, A., &Sandha, K. (2012). Superiority of Blowfish Algorithm in Wireless Networks. *International Journal of Computer Applications. 44. 23-26.* doi: 10.5120/6308-8632

[29] Smith, A., Johnson, B., & Williams, C. (2021). Addressing the Challenges of Modern Encryption and Decryption Systems. *Journal of Data Security.* 10(3), 215 -230.

[30] Smith, J., & Johnson, M. (2018). A Hybrid Encryption System using RSA and AES Algorithms. *International Journal of Computer Science and Information Security.* 16(8), 18-24.

[31] Verma, A., Guha, P., &Mishra, S. (2016). Comparative Study of Different Cryptographic Algorithms. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 5(2), 58-63.*

[32] Wang, H., Hu, J., Zhang, H., & Gao, Y. (2018). Research on the Optimized Implementation of AES Algorithm for File Encryption. *Journal of Physics: Conference Series.* 1067(7), 072017.

[33] Wang, T., Zhang, X., Li, Z., & Wei, Y. (2020). Performance and Security Analysis of RSA Algorithm in File Encryption. *Journal of Physics: Conference Series.* 1674(1), 012035.

[34] Waybhase, S.K., &Adakane, P. (2022). Data Security using Advanced Encryption Standard (AES). *International Journal of Engineering Research & Technology (IJERT).* 11 (06). doi: 10.17577/IJERTV11IS060338

[35] Zhao, X., Zhang, J., Ma, T., &Xie, Y. (2018). Optimized Implementation of RSA Encryption Algorithm Based on Parallel Computing. *Security and Communication Networks, 1-9.*