

Optimized Rivest, Shamir and Adleman (RSA) for Network Inter-Layer Communication

Peter Ezekiel, Gregory M. Wajiga, Ibrahim Adamu and Usa John Guli

Department of Computer Science, Faculty of Physical Science Modibbo Adama University, Yola, Nigeria.

Corresponding author: ezekielpeter25@gmail.com

Abstract

This paper examines the use of Optimized Rivest, Shamir and Adleman (RSA) for Network Inter-Layer Communication. RSA is an algorithm used for secure data transmission which is based on the difficulty of factoring large prime numbers. It is an efficient public-key cryptography method used for authenticating messages and digital signatures. We examine the performance of RSA in network inter-layer communication and compare it to other methods. To achieve this appropriate key sizes of AES- 256 bits to balance security and performance was chosen for a good balance between security and performance. Key Generation, load balancing and performance monitoring were also considered to fine tune the implementation using this optimization technique. Our results show that RSA performs well in terms of both security and performance, making it a viable choice for secure network inter-layer communication. The computational complexity of RSA encryption is greatly reduced, according to simulation results, using the suggested approach. Overall, the suggested method enhances the RSA cryptographic protocol's functionality when used for inter-layer communication in networked systems. The findings demonstrated that RSA had a faster encryption time than AES, leading to a high throughput for the RSA approach.

Keywords: Advanced Encryption Standard, Information Authentication, Network Inter-Layer, Security, Algorithms, Encryption, Decryption.

Introduction

Due to the necessity of keeping in touch with friends, family, and coworkers, the internet was created to play a crucial role in our lives today. Through a substantial number of commercial, public, business, academic, and governmental networks, the internet is a system of interconnected networks that facilitates international communication and access to data resources (Lindquist, 2020). People communicate online via email and the World Wide Web for a variety of reasons, such as online buying, data distribution, marketing, education, and research. Due to the Internet's rapid growth, each company with an internal private network that is connected to the Internet must now take network security into account.

Network security is the set of rules and practices used to prevent and monitor unwanted access to, or use of, a computer network or its available resources. On a network, access to data must be permitted, and the network administrator has control over this procedure. By

choosing or receiving an ID, a password, or another kind of authentication, users can access information and programs that fall under their scope. The most common and simple way to protect a network resource is to give it a unique name and a matching password (Marcon, *et al.*, 2021). Security management for networks varies depending on the situation in order to prevent spamming and hacking assaults. In order to reduce vulnerability to malicious attacks from external network risks, businesses typically utilize technologies that perform network security verifications (Perwej *et al.*, 2021).

Statement of the Problem

Consciously or unconsciously, we all exhibit the need for privacy. This is evident in our use of gadgets. We put security features like passwords and PINs on our phones and computers plus a host of other actions to see that our information is not been comprised. Therefore, to makes it computationally infeasible to break the encryption without the private key, and to ensure that data exchanged between different layers of the network remains confidential and protected from unauthorized access. It becomes pertinent to enhance security.

The Objectives of the Study

The objectives are to:

- design a hybrid model based on the Advanced Encryption Standard and Rivest Shamir and Adleman cryptography,
- implement the model using Visual Studio, and
- evaluate the efficiency of the proposed model in terms of encryption and decryption time and throughput.
- perform comparative analysis of the complexity of the hybrid, AES and RSA.

When building a safe network, the physical security of the computer network is prioritized. Shutting down servers is the simplest and most effective method for securing them (Sharma, 2018). To protect data, cryptography changes readable and understandable data into an incomprehensible form. RSA is one of the most important cryptographic algorithms now in use for guaranteeing secure network communication (Nisha & Farik, 2017).

Encryption can be used to further strengthen the network security mechanism. Any encrypted document, data, or information cannot be read by anybody without the key or the formula used to translate the original text while it is in readable form (Sarkar, Chatterjee & Chakraborty, 2021). The network-transmitted encrypted message must be decrypted as part of the decryption process. The receipts program uses the sender's public key to decrypt the text. The messages were not faked if it works. The message is then decrypted and the digest is calculated using the recipient's software and private key. If the computed digest and the digest communicated with the message are in agreement, tempering has not occurred (Nepal, 2010). According to Jirwan, Singh and Vijay (2010) and Jirwan, Singh and Vijay, (2013) after much research, it is still difficult to determine the prime factors of two

numbers, even though it is straightforward to determine whether a number is prime. The RSA encryption keeps its security in this way.

The RSA encryption algorithm, which is often employed, is based on the mathematical characteristics of large prime numbers. Secure communication and data encryption are its main uses. RSA can be used to provide secure channels between the various tiers of a network protocol stack in the context of network inter-layer communication. Therefore, the purpose of this study is to create an optimized RSA for network inter-layer communication with the following goals: design a hybrid model based on Rivest Shamir and Adleman cryptography and the Advanced Encryption Standard, implement the model using Visual Studio, and assess the effectiveness of the suggested model in terms of encryption and decryption time and throughput as well as a comparison of the complexity of the hybrid, AES, and RSA.

This study is of great significance because RSA plays a crucial role in ensuring secure, authenticated, and private communication between different layers of a network. Its strengths lie in providing data confidentiality, integrity, and authenticity while enabling efficient key exchange, making it an essential tool for network inter-layer communication.

Literature Review

The number of Internet users has significantly increased recently. As a result, digital data has taken the top spot in our time because the majority of businesses and governments now completely rely on the Internet to transmit their consumer data and financial operations. In order for absolute dependence on digital data to be protected against theft and modification and to ensure that it reaches the intended user intact and undisturbed, this data must be secured using encryption and its many protocols (Al Busafi & Kumar, 2020). Data security is a prerequisite for data transmission today, which relies primarily on digital methods in order for data to reach its intended audience. As a result, both symmetric and asymmetric cryptography approaches are used to provide data security. Different asymmetric cryptography methods, including RSA, Diffie-Hellman, DSA (Digital Signature Algorithm), and ECC (Elliptic Curve Cryptography), are examined in this review study (Jirwan, Singh & Vijay, 2013). Sharma (2018) described cryptography as a means for hiding message content confidentiality to ensure that only the person with the decode key can understand the message received; Authentication and Data Integrity to guarantee that the message received has not been tampered with; Non-Repudiation: to confirm that the sender actually sent this communication and that it was received by the intended recipient, and Access Control: limits on allowed access and the level of access authority. Abdeldaym et al. (2019) presented a modification to the RSA cryptographic system with a double public and private key pair and the usage of the Chinese Remainder Theorem (CRT) in an intriguing piece of work. P and Q are formed in the first round of the key generation process, followed by r and s in the second round, to produce the public and private keys.

There are many different kinds of cryptographic algorithms, including the RSA algorithm. According to Sihotang, Efendi, Zamzami, and Mawengkang (2020), the RSA method is a contemporary asymmetric cryptographic methodology that employs two keys: a public key for encryption and a private key to make the process descript. According to Zaw et al. (2019), there are six techniques to secure databases using the Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC). Random AES-256 bits key were utilized for the encryption in the first (rows-level encryption), the second (columns-level encryption), and the third (elements-level encryption). Rahim and Ikhwan (2016) described the encryption method as a set of rules that converts readable plain text—meaningful data—into an unintelligible form called cipher text. The encrypted text must be converted into the original plain text during the decryption procedure. In order for the cipher text to be a function of both the plain text and the encryption key, an additional piece of data known as a key is used in both procedures. The various cipher texts will be obtained if the key is altered.

Subedar and Araballi (2020) built AES, ECC, and RSA separately and then as hybrids (AES-RSA hybrid and RSA-ECC hybrid), to demonstrate that hybrid algorithms outperform individual algorithms. Their proposed method involves first hashing the plaintext with SHA-256, then encrypting it with AES, and finally re-encrypting the ciphertext with RSA. The plaintext for the second hybrid is also hashed using SHA-256, encrypted using RSA, and then the ciphertext produced is encrypted with ECC. Their investigation revealed that RSA-ECC outperformed AES-RSA with an encryption time of 10.25 seconds compared to 8.33 seconds for RSA-ECC. ECC is less secure than AES, though. Saho and Ezin (2020) carried out a comparative study of ECC protocols (ECNR, ECDSA and ECIES) and RSA. The ECC algorithms and RSA algorithm were run and compared side-by-side and the result showed that ECC algorithms are more advantageous over RSA cryptosystem in terms of computing speed and memory consumption. To make emphasis on the maximum speed, better performance and effectiveness of algorithms, Jintcharadze and Iavich (2020) proposed a hybrid implementation of Twofish, AES, ElGamal and RSA cryptosystems. In the first hybrid (Twofish-RSA), the plaintext is first encrypted with Twofish followed by RSA and in the second hybrid (AES-RSA), the plaintext is first encrypted with AES, followed by RSA.

The final hybrid (AES-ElGamal) encrypts the plaintext using AES first, then ElGamal cryptography. Java programming language was utilized to implement the results analysis. The outcomes demonstrate how significantly secure the suggested AES-RSA hybrid algorithm is. The Two fish-RSA hybrid, however, has advantages in terms of computation time, memory usage, and cipher text size. A comparison of various well-known cryptographic methods (RSA, ElGamal, and ECC) is presented by Mallouli et al. (2019), with an emphasis on the speed at which these algorithms execute. The standard encryption structure was used to create these methods, which encrypts the plaintext using a predetermined algorithm and a randomly generated key. As they noted, RSA suffers from slow execution time because to its high primes, whereas ElGamal is semantically insecure.

As a result, ECC provided greater performance yet is acceptable for the smaller and less capable devices that people use today for network access services. The number of people using email services and the number of assaults on these services both expand along with the development and use of technology. Sari et al. (2020) suggested employing the RSA method to optimize the data encryption model in order to increase the security of email data. According to their proposal, the email message is first encrypted using RSA (with big primes). Their work was executed utilizing the java programming language on the NetBeans IDE. The production and use of large primes will lengthen the execution time even if this might increase the security of email contents.

Methodology

This research aims to design an optimized RSA algorithm for network inter-layer communication. This is done by adopting the Advanced Encryption Standard (AES) for plaintext encryption and the RSA cryptographic technique for key encryption due to its suitability for signing and encryption.

Algorithm for Optimized RSA for Network Inter-Layer Communication

The private key here is a pair of numbers ($\mathbf{N}$, d), and the public key is a pair of numbers ($\mathbf{N}$, e). Note that $\mathbf{N}$ is common to the private and public keys.

The sender uses the following algorithm to encrypt the message:

$$\mathbf{C} = \mathbf{P}^e \bmod \mathbf{N}$$

In this algorithm, $\mathbf{P}$ is the plain text, which is represented as a number; $\mathbf{C}$ is the number that represents the ciphertext. The two numbers e and $\mathbf{N}$ are components of the public key. Plaintext $\mathbf{P}$ is raised to the power e and divided by $\mathbf{N}$. the mod term indicates that the remainder is sent as the ciphertext.

The receiver uses the following algorithm to decrypt the message:

$$\mathbf{P} = \mathbf{C}^d \bmod \mathbf{N}$$

In this algorithm, $\mathbf{P}$ and $\mathbf{C}$ are the same as before. The two numbers " d " and " $\mathbf{N}$ " are components of the private key. Figure below describes this.

Key Generation:

- Select two large prime numbers, p and q .
- Calculate $N = P * Q$. This is the modulus used for both the public and private keys.
- Compute $\phi(N) = (P-1) * (Q-1)$, where ϕ is Euler's totient function.
- Choose a public exponent, e , such that $1 < e < \phi(N)$ and $\gcd(e, \phi(N)) = 1$.
- Calculate the private exponent, d , using the extended Euclidean algorithm so that $d \cdot e \equiv 1 \pmod{\phi(N)}$. (e, n)
- the private key, while (d, N) is the public key.

Encryption:

- a. Convert the message to be sent into an integer representation, M , according to a predefined scheme.
- b. Compute the ciphertext, C , using the formula $C = M^e \bmod N$.
- c. Transmit the ciphertext to the receiving layer.

Decryption:

- a. Receive the ciphertext, C .
- b. Compute the plaintext, P , using the formula $P = C^d \bmod N$.
- c. Convert the plaintext back to its original form using the predefined scheme.

Optimizations for network inter-layer communication:

- a. Use a precomputed table of prime numbers to speed up the prime number generation process during key generation.
- b. Implement modular exponentiation efficiently using algorithms like exponentiation by squaring or Montgomery multiplication.
- c. Employ multi-threading or parallel processing techniques to distribute the computational load across multiple cores or machines, especially for large key sizes and frequent encryption/decryption operations.
- d. Implement a secure key exchange mechanism to securely distribute the public keys between network layers.
- e. Apply appropriate cryptographic protocols for secure communication, such as Transport Layer Security (TLS) or Secure Shell (SSH), to protect the RSA-encrypted data during transmission.

Definition of variables:

P, Q : The two large prime numbers used for key generation.

$\phi(N)$: Euler's totient function, computed as $\phi(N) = (p-1) * (q-1)$.

e : The public exponent.

d : The private exponent.

Use Case Diagram

In Figure 1, there are two actors—the sender and the receiver—as well as a number of procedures. The sender logs in, enters data to be encrypted or decrypted, and then logs out of the system. The identical set of processes can be carried out by the receiver on the other end to secure their communication as well.

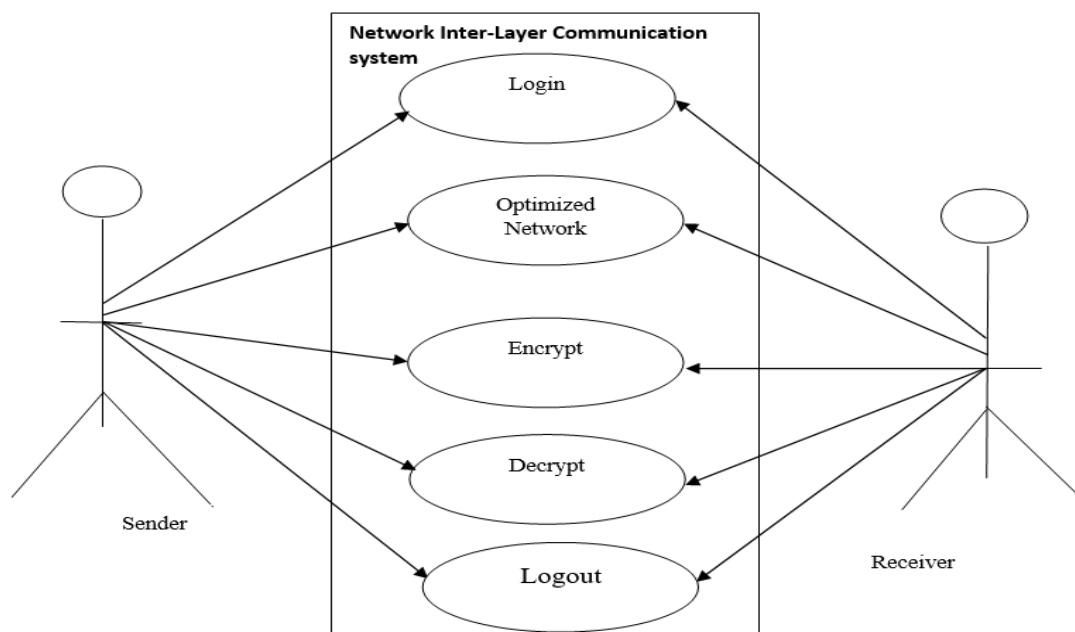


Figure 1: Use case diagram of network layer communication

Table 1: Network Layer communication fields

FIELD	DATATYPE	LENGTH/VALUES	NULL
ID	TEXT	30	
NAME	TEXT	15	
PLAINTEXT	TEXT	30	
NETWORK LAYER	TEXT	15	
ENCRYPT	TEXT	15	
OUTPUT	TEXT	15	

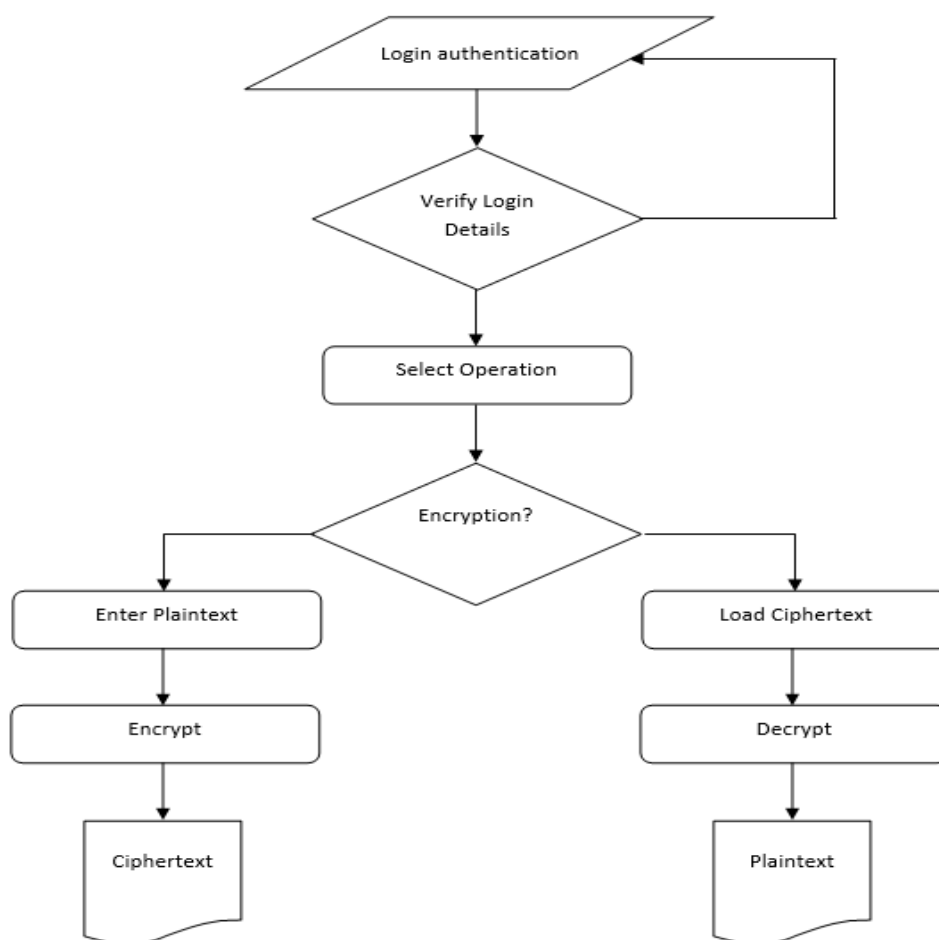


Figure 2: Program flowchart for the description and encryption process

The Throughput is based on the encryption and decryption times which indicate the communication speed of an encryption/decryption scheme. It is given by:

$$\text{Throughput} = \frac{T_p}{T_c} \quad (1)$$

Or

$$\text{Throughput} = \frac{E_t}{D_t} \quad (2)$$

Where T_p refers to the total plaintext bytes, T_c the total ciphertext bytes, E_t and D_t the encryption and decryption times (in seconds).

Results and Discussions

Login module

This module allows the user of the system to enter his/her credentials, once it is validated, it grants the user access to the system or if not validated, it denies access.



Figure 3: Shows the login interface

Encryption module

This module was designed to encrypt a plaintext message before been transferred over the network. This is to ensure that no malicious user is able to decipher the content.



Figure 4: Showing the encryption module function

The textbox field labeled "plaintext" asks for the plaintext from Figure 4 to be sent across the network. The "AES Key" field is filled out with an AES key (of the user's choosing and length 32), and the "Plaintext Encryption" button is then pressed to create the ciphertext seen in the "Ciphertext" field. The user must first generate a public key and a private key by giving two prime numbers in order to encrypt the AES key. The AES key will then be

encrypted using the newly created public key and displayed in the area beneath the ciphertext. Following that, the user will need to either keep the private key safe or come up with a secure method of sending it to any other parties engaged in the communication. Since it will be used to decrypt the AES key and eventually the delivered message.

The "Save Ciphertext" button must be clicked in order to store the ciphertext as a file, which creates a save file dialog box where the user can select a location and enter a file name. The ciphertext is then saved as an encrypted file (with the .enc extension). The user only needs to click the "Save AES Key" button and follow the same steps as they did to save the ciphertext in order to save the encrypted AES key.

When the Plaintext Encryption button is pressed, a call is made to the function "AesEncryptStringToBase64" with the input variables "plaintext" and "key"; the function returns a ciphertext that is displayed in the field "Ciphertext". It takes both a private key and a public key to perform RSA encryption. The user enters two distinct prime numbers and clicks the produce key pair button to generate the keys. With the two prime numbers input, the button calls the "generate_key_pair" function, which returns two pairs that are shown as the public key and the private key. By pressing the Encrypt AES Key button, you can call the "rsaEncryption" function, pass an AES key, and get the key's ciphertext.

Decryption module

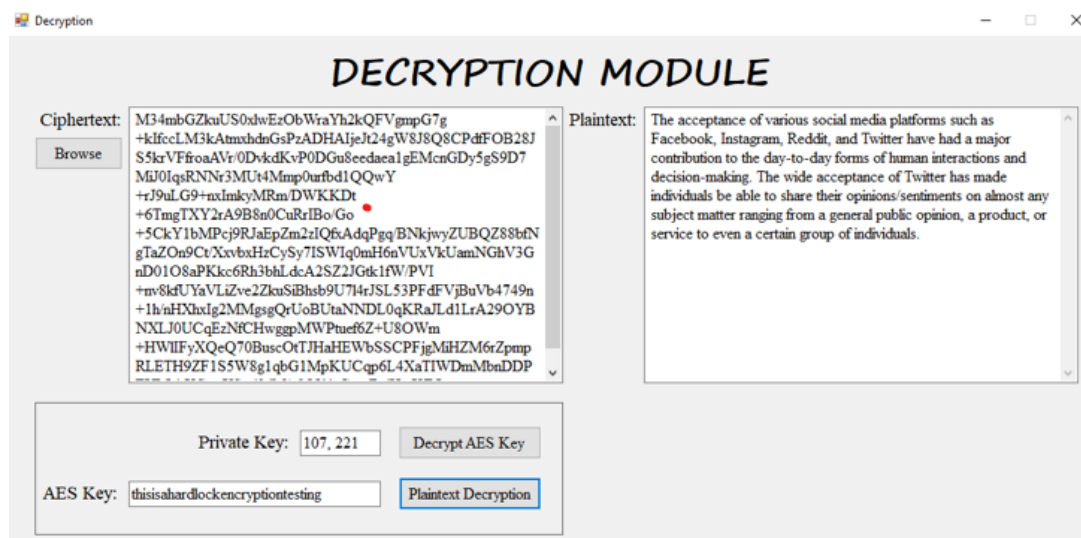


Figure 5: Showing the Decryption Module function

The recipient will then receive both the encrypted AES key file and the encrypted ciphertext file. The two encrypted files must be decrypted once they are sent to the recipient, as shown in Figure 5. The user only needs to click the "Browse" button and choose the location of the ciphertext to load it onto the module; it will then appear in the "Ciphertext" box. The encrypted key, which is saved in a variable, must be loaded by the user and is accessible under the "Browse" button as well. The encryption key must now be entered; after hitting the "Decrypt AES Key" button, the key will have been unlocked and displayed in the "AES

Key" field. This key was generated during the encryption procedure (and provided to the recipient in another secure means). The "Plaintext Decryption" button has now been clicked, and the ciphertext is now decrypted and displayed in the "Plaintext" box.

Table 2: Encryption Times for AES and RSA

S/No	Test Data	AES(ms)	RSA(ns)	Total Time(ms)
1	Plaintext 1-354bytes	7	126	133
2	Plaintext 2-956bytes	20	139	159
3	Plaintext 3-1.7Kb	37	108	145
4	Plaintext 4-2.27Kb	58	138	196
5	Plaintext 5-3.91Kb	111	113	224

Table 3: Decryption Times for AES and RSA

S/No	Test Data	AES(ms)	RSA(ns)	Total Time(ms)
1	Ciphertext 1-495 bytes	7	131	138
2	Ciphertext 2-1.27Kb	7	132	139
3	Ciphertext 3-2.27Kb	9	132	141
4	Ciphertext 4-3.02Kb	10	132	142
5	Ciphertext 5-5.19Kb	13	132	145

The experiment consisted of five (5) rounds, and the results are shown in Table 3 as the length of time needed to encrypt the plaintext and Table 4 Encryption Throughput for various plaintext lengths using AES and RSA, respectively, as well as the variances in that time.

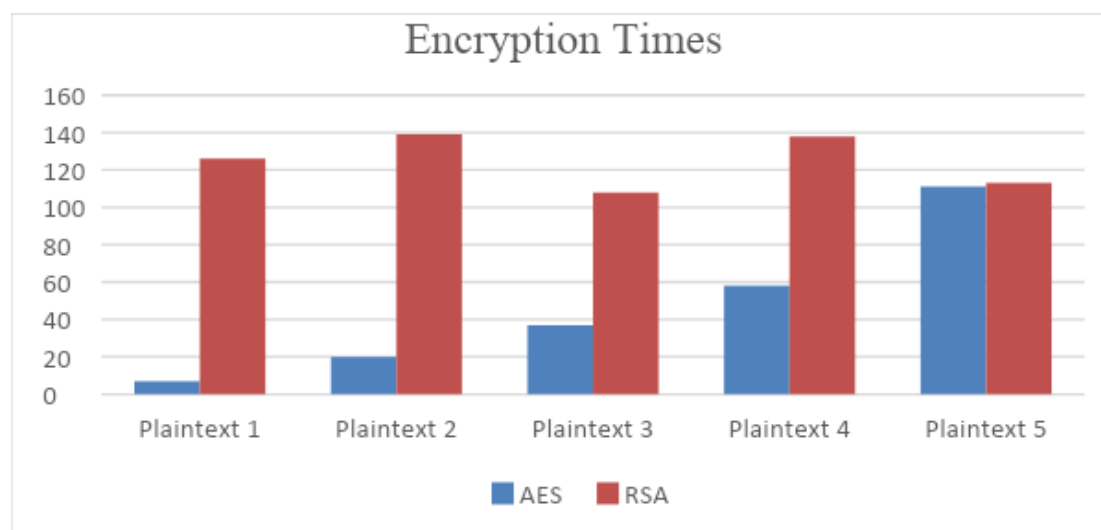


Figure 6: Visualize the comparison of AES and RSA Encryption Times

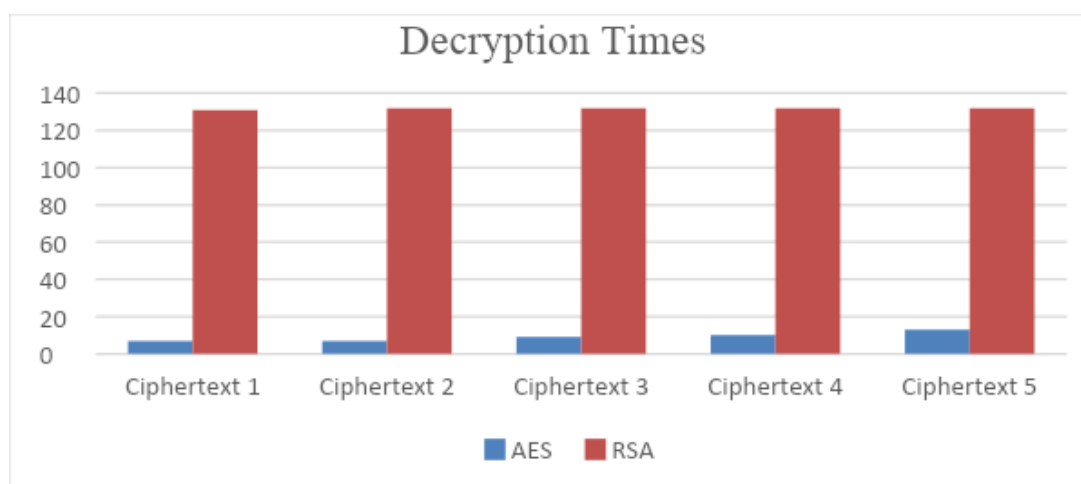


Figure 7: Shows the comparison of AES and RSA Decryption Times

From the results obtained in Table 2, it can be seen that for AES, the time taken to decrypt the ciphertexts is relatively lower compared to the encryption times even though the size of the ciphertext is larger as opposed to the plaintext. However, for RSA, the decryption time increased and remained almost linear for the different keys as the result shows in Figure 7.

Table 4: Encryption Throughput for various ciphertext lengths

S/No	Plaintext Size (Kb)	Encryption Time (s)	Throughput (Kb/s)	AES Key Size (Kb)	Encryption Time (s)	Throughput (Kb/s)
1.	0.39	0.007	55.71	0.03	0.126	0.24
2.	0.97	0.020	48.50	0.03	0.139	0.22
3.	1.74	0.037	47.03	0.03	0.108	0.28
4.	2.31	0.058	39.83	0.03	0.138	0.22
5.	3.95	0.111	35.59	0.03	0.113	0.27

Table 5: Decryption Throughput for various ciphertext lengths

S/No	Ciphertext Size (Kb)	Decryption Time (s)	Throughput (Kb/s)	AES Key Size (Kb)	Decryption Time (s)	Throughput (Kb/s)
1.	0.48	0.007	68.57	0.17	0.131	1.30
2.	1.27	0.007	181.43	0.17	0.132	1.29
3.	2.27	0.009	252.22	0.17	0.132	1.29
4.	3.02	0.010	302.00	0.17	0.132	1.29
5.	5.19	0.013	399.23	0.17	0.132	1.29

Conclusion

In our review of related works, we stated that RSA is a strong cryptographic technique, but however suffers from long time complexity. In this work, we carried out several experiments to examine the impact of our proposed model, with our major focus on the encryption and decryption times of the techniques. To optimize the use of RSA for network inter-layer communication, we proposed a hybrid of the RSA with AES (which is another strong cryptographic technique) where the AES was used to encrypt the plaintext messages and then RSA was then used to encrypt the AES keys. From the results of the experiments, we saw that even though the size of the keys encrypted by RSA is not equal or more than the size of the plaintexts encrypted by AES, the time it took for encryption was more than that by AES. We can conclude that hybridization of these two algorithms have optimized RSA by reducing the computation time while still providing enhanced security. This work was able to achieve its aim and objectives and was by no means an exhaustive investigation into the impact of the choice of cryptographic techniques in attaining confidentiality in communication. However, some few recommendations are made. Future may consider exploring other block sizes of AES for encryption or splitting the plaintext messages into various halves and then use different techniques.

Recommendations

This study recommends that applications, services, or protocols that require encryption, digital signatures, or key exchange to ensure confidentiality, integrity, and authenticity of data:

- To provide secure web communication between web browsers and web servers.
- To secure email communication, including digital signatures and encryption to protect the content and verify the sender's identity.
- Secure Shell Communication for key-based authentication, allowing secure access to remote servers and network devices.
- To generate digital certificates used for website authentication and secure communication.
- Secure File Transfer ensuring data confidentiality.

References

- Abdeldaym, R. S., Abd Elkader, H. M., & Hussein, R. (2019). Modified RSA algorithm using two public key and Chinese remainder theorem. *IJ of Electronics and Information Engineering*, 10(1), 51-64.
- Al Busafi, S., & Kumar, B. (2020, December). Review and Analysis of Cryptography Techniques. In *2020 9th International Conference System Modeling and Advancement in Research Trends (SMART)* (pp. 323-327). IEEE.
- Jintcharadze, E., & Iavich, M. (2020, September). Hybrid implementation of Twofish, AES, ElGamal and RSA cryptosystems. In *2020 IEEE East-West Design & Test Symposium (EWDTS)* (pp. 1-5). IEEE.
- Jirwan, N., Singh, A., & Vijay, S. (2013). Review and analysis of cryptography techniques. *International Journal of Scientific & Engineering Research*, 4(3), 1-6.

- Lindquist, D. (2020). Does Internet Use Reduce Birth Rates?: A Cross-Country Study of 44 Lower-Middle-Income Countries Between 2000 and 2017.
- Mallouli, F., Hellal, A., Saeed, N. S., & Alzahrani, F. A. (2019, June). A survey on cryptography: comparative study between RSA vs ECC algorithms, and RSA vs El-Gamal algorithms. In *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)* (pp. 173-176). IEEE.
- Marcon, Y., Bishop, T., Avraam, D., Escriba-Montagut, X., Ryser-Welch, P., Wheeler, S., ... & González, J. R. (2021). Orchestrating privacy-protected big data analyses of data from different resources with R and DataSHIELD. *PLoS computational biology*, 17(3), e1008880.
- Nepal, S. (2010). *Enhanced Security Encryption for Data Storage Using Multiple Keys* (Doctoral dissertation, Department of Computer Science and Information Technology).
- Nisha, S., & Farik, M. (2017). RSA Public Key Cryptography Algorithm. A Review. *International Journal of Scientific and Technological Research*, 6, 187-191.
- Perwej, Y., Abbas, S. Q., Dixit, J. P., Akhtar, N., & Jaiswal, A. K. (2021). A systematic literature review on the cyber security. *International Journal of scientific research and management*, 9(12), 669-710.
- Rahim, R., & Ikhwan, A. (2016). Cryptography technique with modular multiplication block cipher and playfair cipher. *Int. J. Sci. Res. Sci. Technol*, 2(6), 71-78.
- Saho, N. J. G., & Ezin, E. C. (2020, October). Comparative study on the performance of elliptic curve cryptography algorithms with cryptography through RSA algorithm. In *CARI 2020-Colloque Africain sur la Recherche en Informatique et en Mathématiques Appliquées*.
- Sari, I. C., Zarlis, M., & Tulus, T. (2020, February). Optimization of data encryption modeling using RSA cryptography algorithm as security e-mail data. In *Journal of Physics: Conference Series* (Vol. 1471, No. 1, p. 012068). IOP Publishing.
- Sarkar, A., Chatterjee, S. R., & Chakraborty, M. (2021). Role of cryptography in network security. *The "Essence" of Network Security: An End-to-End Panorama*, 103-143.
- Sharma, M. (2018). Review on Cryptography in Network security. *International of Engineering Research and Technology*, 2(3).
- Sihotang, H. T., Efendi, S., Zamzami, E. M., & Mawengkang, H. (2020, November). Design and implementation of Rivest Shamir Adleman's (RSA) cryptography algorithm in text file data security. In *Journal of Physics: Conference Series* (Vol. 1641, No. 1, p. 012042). IOP Publishing.
- Subedar, Z., & Araballi, A. (2020). Hybrid cryptography: Performance analysis of various cryptographic combinations for secure communication. *International Journal of Mathematical Sciences and Computing (IJMSC)*, 6(4), 35-41.
- Zaw, T. M., Thant, M., & Bezzateev, S. V. (2019, June). Database security with AES encryption, elliptic curve encryption and signature. In *2019 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF)* (pp. 1-6). IEEE.