DEVELOPMENT OF AN ENHANCED PELICAN RSA ENCRYPTION TECHNIQUE FOR DATA SECURITY SYSTEM

Babatunde Victor Feyiseto
Ladoke Akintola University of Technology, Ogbomoso, P. M. B. 4000, Oyo State.
E-mail: babatundevictorf@gmail.com

Adeyemo Isiaka Akinkunmi, PhD *
Ladoke Akintola University of Technology, Ogbomoso, P. M. B. 4000, Oyo State.
E-mail: iaadeyemo22@lautech.edu.ng

Jeremiah Yetomiwa Sinat
Ladoke Akintola University of Technology, Ogbomoso, P. M. B. 4000, Oyo State.
Email: ysjeremiah@lautech.edu.ng

Ajala funmilola Alaba, PhD
Ladoke Akintola University of Technology, Ogbomoso, P. M. B. 4000, Oyo State.
Corresponding Author- E-mail: iaadeyemo22@lautech.edu.ng

ABSTRACT

In today’s digital world, strong data security is crucial, focusing on confidentiality, authentication, integrity, and non-repudiation. The Rivest-Shamir-Adleman (RSA) algorithm, a key public-key cryptographic method, secures communication based on the difficulty of prime factorization. However, its security and efficiency face threats from advanced factorization techniques. This study enhances RSA by integrating the Pelican Optimization Algorithm (POA), a nature-inspired metaheuristic, to optimize key generation and parameters boosting both security and performance. The Pelican RSA cryptosystem for image security was developed using a structured approach. The dataset which consists of 25 medical images were gotten from public sources, were pre-processed and converted to JPEG format. The encryption task was defined, and the Pelican Optimization Algorithm (POA) was integrated with RSA to improve key generation and encryption. Images were divided into blocks and encrypted individually using the optimized RSA. A MATLAB 2020a GUI was built for user interaction. POA-enhanced RSA was compared to traditional RSA using encryption time, decryption time, throughput and memory size. Decrypted images were reconstructed to verify accuracy and effectiveness. The results of the evaluation and comparison indicate that Pelican RSA achieved reduction in encryption time from 9.45–19.93s to 4.48–8.89s, decryption time from 11.54s to 4.61s and boosts throughput from 3.57–7.51 kb/s to 9.50–14.99 kb/s. Although it uses slightly more memory (678 KB vs. 569 KB), this is justified by its speed and efficiency.

This study shows that using the Pelican Optimization Algorithm (POA) greatly improves RSA cryptosystem security. POA enhances key generation and parameter tuning, boosting resistance to prime factorization attacks. This makes RSA more reliable for secure online communication. The core contribution is applying a nature inspired optimization method to strengthen public-key cryptography.

Keywords: Authentication, integrity, and non-repudiation, Pelican Optimization, Algorithm-based

DOI: 10.7176/JIEA/15-2-07

Publication date: August 28th 2025

MAIN BODY

Data security means protecting data from destructive forces and the bad behavior of unauthorized users. A large amount of private information is being swapped via the Internet (public open media) because this is the cheaper and readily available method (Prachi and Tijare, 2017). As digital information and data are transmitted through the Internet, protecting sensitive information needs to be discovered and developed more frequently than ever before. Also, new technologies for protecting sensitive information need to be implemented and developed. Data hiding technology can be used to embed secret messages into compressed image bitstreams for transaction tracking and access control. Some techniques for hiding data evaluate the quality of compressed images without the original reference. The quality is estimated based on the degradation of the hidden message extracted by calculation.

With the development of computers and their increasing use in different areas of life and work, information security issues have become particularly important. One of the concerns of information security is the concept of hidden information exchange using steganography. Although, information is hidden but not secured which makes anyone that knows the steganographic algorithm to retrieve the data (Deepa et al., 2015). The increase in unauthorized attacks, security interruptions, and unnecessary access is frustrating that it is important to protect our information from hackers or unnecessary access. Although in cryptography, the message is encrypted, when communicating with a third party, the encrypted message can easily be decrypted when the key length is short. In steganography, data or information are just hidden or obscured, and the data is also hidden, so it is impossible to easily eavesdrop communication between two parties (Gupta et al., 2017).

The development of a data security system using a pelican RSA encryption technique is a significant area of research in the field of computer science and information technology. In this study, researchers aim to develop an effective data security system that can protect sensitive information from unauthorized access and cyber threats. The RSA algorithm is a widely used encryption technique that provides secure communication channels between two parties by encoding the message with the help of public and private keys. Several studies have been conducted in the past that focus on the development of data security systems using RSA encryption techniques. For instance, Singh and Sharma (2020) proposed a hybrid encryption technique that combines the RSA algorithm with the Advanced Encryption Standard (AES) to provide improved security and performance.

Moreover, researchers have also investigated the use of various optimization techniques to enhance the performance of RSA encryption algorithms. For example, Li et al. (2019) proposed an optimized RSA encryption algorithm that uses a modified square-and-multiply algorithm to reduce the computational complexity and enhance the efficiency of the algorithm.

Cryptography is the study and research of techniques that protects the communication between two authorized parties in the presence of one or more unauthorized third parties, and it is an important tool for ensuring information security. The art and science of keeping messages, data, or information in secrecy so that it can only be understood by someone that possesses specific information is considered cryptography. The term cryptography comes from the Greek "kryptós", which stands for "hiding", and "gràphin" stands for "writing". Therefore, the correct meaning of cryptography is "hidden writing" (Babu et al., 2010; Marwa and Abdelmgeid, 2016). The cryptography defined by Vipula and Suresh (2013) and Khalid et al., (2014) is the process of transmitting data securely on the Internet through the application of some cryptographic algorithms, so that it is difficult for an intruder to attack or have access to some private information. Cryptography is a technique of saving and transmitting data in a specific form so that only the people it expects can access and read it. This term is often associated with scrambling plaintext (ordinary text, sometimes known as plaintext) into ciphertext (called the process of encryption) and then back (called decryption) (Kaur and Singh, 2015). Cryptography is the science of using mathematics to encrypt and decrypt data to protect the security of messages by converting understandable data forms (plaintext) to incomprehensible forms (ciphertext) (Marwa et al., 2016).

Does enlarged security bring comfort to paranoid people? Or does security bring some primary protections that we are ignorant to believe that we do not need? During this time when the Internet provides indispensable communication among literally billions of people and is used as a tool for businesses, social interaction, and the exchange of higher amount of personal information, security has become a critical issue for every user to deal with. There are many facets and applications to security, from secure businesses and payments to private communications and securing health care information. An important aspect of secure communication is cryptography. It is important to note that while cryptography is required for secure communications, it is insufficient (Marwa et al., 2016).

There are five prime functions of cryptography:

i Privacy/confidentiality: guarantee that no one other than the intended recipient can read the message.

ii Authentication: The means of proving identity.

iii Integrity: The message received has not been modified in any way from the original source.

iv Non-repudiation: A process to prove that a message was really sent.

v Key exchange: The process by which cryptographic keys are shared between sender and receiver.

Cryptography starts with unencrypted data, called plaintext. The plaintext is encrypted into cipher text, and the cipher text is (usually) decrypted back to usable plaintext. Encryption and decryption are based on the type of encryption scheme used and some form of key (Marwa et al., 2016). The formula for this process is frequently written in equation 2.1 and 2.2:

$$C=E_k(P) \qquad (2.1)$$

$$P = D_k(C) \qquad (2.2)$$

where **P** is the plaintext, **C** is the ciphertext, **E** is the encryption method, **D** is the decryption method, and **k** is the key.

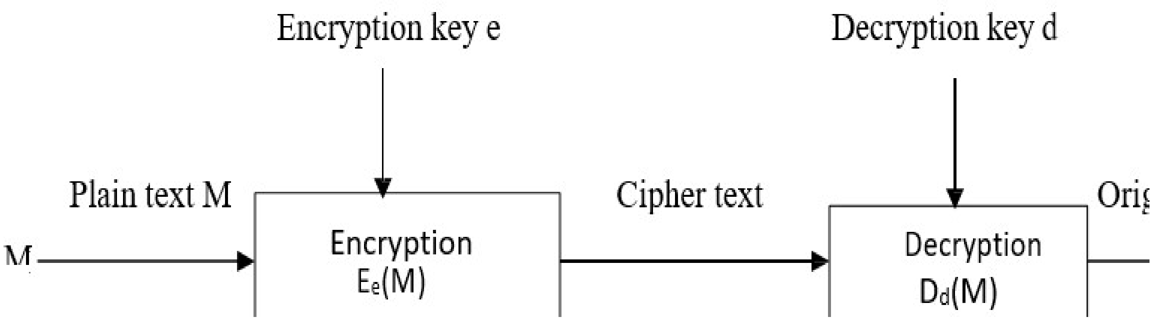
Cryptography is majorly classified into symmetric key cryptography and asymmetric key cryptography (widely known as public key cryptography) technology. Other types of cryptography include visual cryptography, Elliptic-Curve Cryptography, Quantum Cryptography and so on (Dhiren, 2010).

In symmetric key cryptography, both parties use the same key. The sender make use of this key and encryption algorithm to encrypt the data, while the receiver also makes use of same key and corresponding decryption algorithm to decrypt the data. Symmetric key encryption has two major components which are:

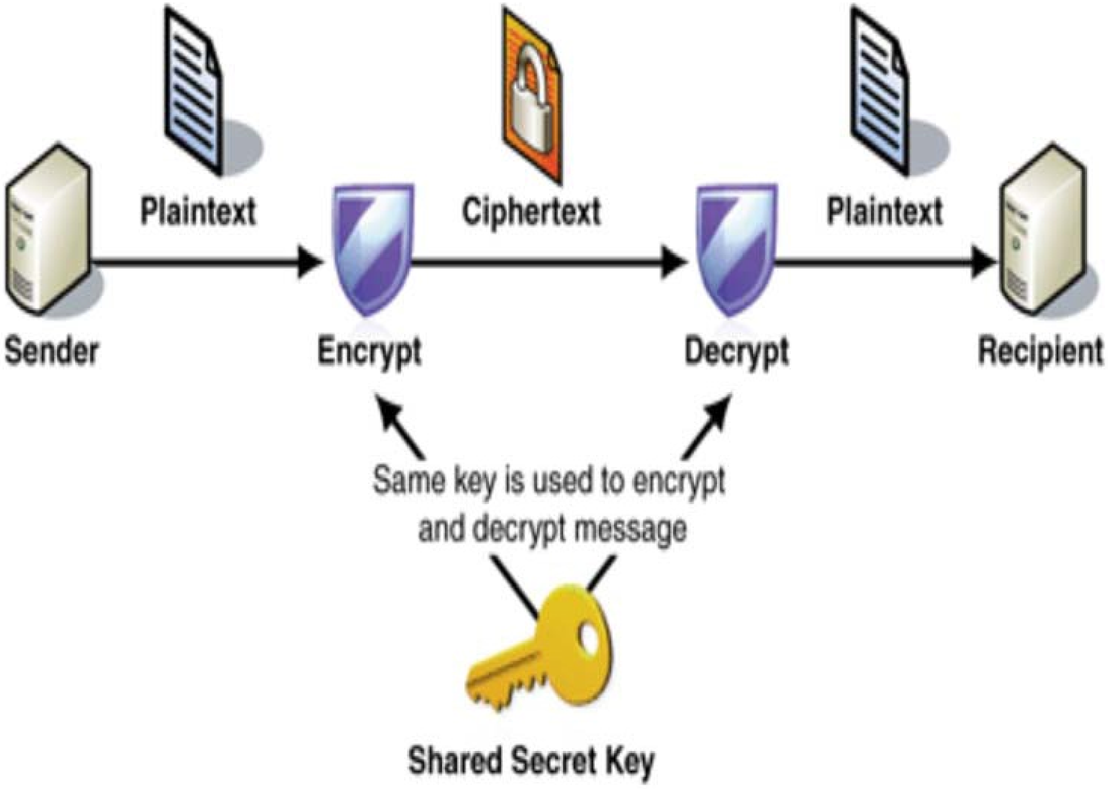
i Encryption algorithm/decryption algorithm: The encryption algorithm uses a key to perform various transformations on the plaintext. The decryption algorithm is essentially an encryption algorithm that runs in reverse, the symmetric ciphers use the same cryptographic key to encrypt and decrypt.

ii Secret Key: The important input of an encryption/decryption algorithm determines the transformations performed by the algorithm to produce an output (Dhiren, 2010).

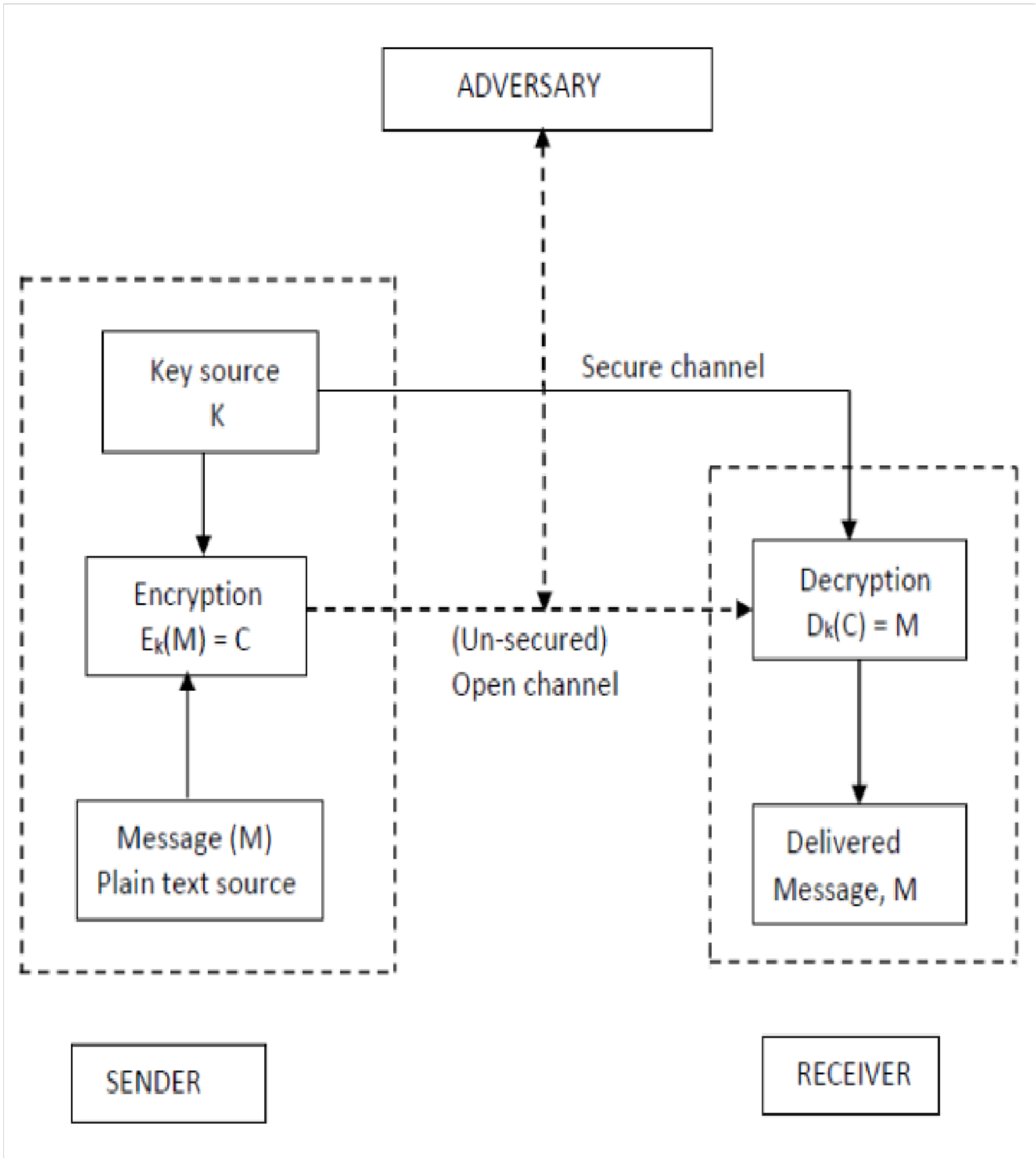
The ciphertext is the scrambled message generated by the output of the encryption algorithm, which depends on the plaintext and the key. Even if the adversary has the complete details and structure of the algorithm used, as well as many other ciphertexts and corresponding plaintexts, he/she should not be able to decrypt the ciphertext or discover the key. The sender and receiver must obtain the key in secure mode. The feature of only keeping the key secret makes symmetric ciphers widely used because they are fast and can easily operate on large amounts of data. The fact that the key must be securely shared between the encrypting party and decrypting party is a well-known problem of symmetric cryptography and not an advantage. When using symmetric key encryption, the main security issue is to distribute appropriate keys to the sender and receiver and maintain the confidentiality of the keys (Dhiren, 2010). Figure 2.1 describes the symmetric key encryption. Figure 2.2 showed the symmetric key cryptography. Figure 2.3 defined communication using symmetric key cryptography.



Encryption and Decryption using two different keys (Dhiren, 2010)



Symmetric Key Cryptography (Dhiren, 2010)



Communication using Symmetric Key Cryptography (Dhiren, 2010)

Visual Cryptography is one of the cryptographic techniques that allow visual information in the form of pictures, text, etc., to be encrypted such that decryption becomes a mechanical process. Visual Cryptographic utilizes two transparent images. One image contains random data or pixels, and the other image has the secured data. Retrieving secret information from encrypted images is almost impossible. Transparent images and layers are necessary to display information. The simplest way to achieve visual cryptography is to print the two layers on a single transparent sheet. The advantage of the visual cipher scheme is that it gets rid of the computational problems in the decryption process and can recover the secret image through the stacking process. This property makes visual cryptography particularly useful for low computational methods.

The elliptic curve is described by a set of solutions of some equations in two variables. Elliptic Curve Cryptography (ECC) is a public key cryptography method based on elliptic curve mathematics over a finite field. ECC was discovered by Victor Miller and Neil Koblitz in 1985 as a substitute mechanism for public key encryption. it provides a public key cryptographic system based on the elliptic curve discrete logarithm problem (DLP) which is like the Discrete Logarithm Problem (DLP) over the integers modulo a prime p. This similarity implies that most of the cryptographic methods executed by the cryptographic system using DLP based on integer modulus p can also be performed in the elliptic curve cryptographic system defined in Figure 4. Another benefit of ECC systems is that they can utilize a much shorter key length than other public-key cryptographic systems and give the same level of security. For example, a 160-bit ECC system is considered to provide almost the same security level as 1024-bit RSA. In addition, the speed of increasing the ECC key size to improve security is much lower than the speed of the integer-based discrete logarithm (DL) or Rivest Shamir Adleman (RSA) key size which must be expanded to also increase security. ECC systems can also give a speedy implementation than RSA or DL cryptographic systems and uses lesser bandwidth and power (Dhiren, 2010).

Quantum Cryptography classical cryptography uses various mathematical methods to impede intruders from understanding the content of encrypted information, in quantum mechanics, information is secured by the laws of physics. The process that allows two users to communicate through a public channel to create a subject of shared and secret information is known as Quantum cryptography. This information is usually in the form of a random bit string, which can then be used as a regular key for secure communication. It is assumed that it is helpful for both communicating parties to deliberately share a small number of secret information, which is used up during the transfer process and then updated, but even without this assumption, exchange of information is still possible. The advantage of quantum cryptography over conventional key transfer methods is that it can prove that the information exchange is secure in a very strong manner without presuming about the difficulty of specific mathematical problems. Even assuming that the theoretical eavesdropper has unlimited computing power, the laws of physics can guarantee (probabilistically) that secret key exchange is secure, taking into account some other assumptions (Dhiren, 2010).

Public Key Infrastructure and private key pair is together encrypting and decrypting messages. Pairing two cryptographic keys in this manner is also known as asymmetric encryption, which is different from symmetric encryption, in which a single key is used for both encryption and decryption. The advantage of asymmetric encryption is that the public key can be published for the world to see, while the private key is kept secure on the user's device, making it much more secure than symmetric encryption.

Public key cryptography relies on mathematical algorithms to generate the keys. The public key is comprised of a string of random numbers that can be used to encrypt a message. Only the intended recipient can decipher and read this encrypted message and it can only be deciphered and read by using the associated private key, which is secret, and known only to the recipient. Public keys are created using a complex cryptographic algorithm to pair them with their associated private key so that they cannot be exploited through a brute force attack. The key size or bit length of public keys determines the strength of protection. For example, 2048-bit RSA keys are often employed in SSL certs, digital signatures, and other digital certificates. This key length offers sufficient cryptographic security to keep hackers from cracking the algorithm. Standards organizations like the CA/Browser Forum define baseline requirements for supported key sizes.

PKI enables the digital certificates that we encounter daily, unobtrusively and ubiquitously, when using websites, mobile apps, online documents, and connected devices. One of the most common use cases of PKI is X.509-based Transport Layer Security (TLS)/Secure Socket Layer (SSL).

This is the basis of the HTTPS protocol, which enables secure web browsing. But digital certificates are also applied to a wide range of use cases including application code signing, digital signatures, and other aspects of digital identity and security. Example of Public Key Cryptography are RSA and DSA.

Difference between RSA and DSA (Source: Islam *et al.,* 2021)

RSA	**DSA**
It is a cryptosystem algorithm.	It is digital signature algorithm.
It is used for secure data transmission.	It is used for digital signature and its verification.
It was developed in 1977.	While it was developed in 1991.
It was developed by Ron Rivest, Adi Shamir and Leonard Adleman.	It was developed by National Institute of Standards and Technology (NIST).
It uses mathematical concept of factorization of product of two large primes.	It uses modular exponentiation and discrete logarithm.
It is slower in key generation.	While it is faster in key generation as compared to RSA.
It in faster than DSA in encryption.	While it is slower in encryption.
It is slower in decryption.	While it is faster in decryption.
It is best suited for verification and encryption.	It is best suited for signing in and decryption.

RSA works with SSH2 but is also compatible with the original SSH, which is now considered heavily flawed. So, if you're concerned about accidentally using SSH, DSA may be a better choice. Table 2.1 defined the difference between RSA and DSA (Islam et al., 2021).

In other words, the difference between RSA and DSA is in what each can do. RSA can be used as a digital signature and an encryption algorithm. Also, RSA is a block cipher, while DSA is a stream cipher. Compatibility-wise, they are equal. RSA and DSA are both used for the same internet protocols and certificates, like Nettle, OpenSSL, wolfCrypt, Crypto++, and cryptlib (Rivera et al., 2019).

Related Works

Recently, many researchers have done a lot of research and work in developing methods for encrypting, decrypting, signing and verifying users in e-banking systems. The following will introduce some of the main works of cryptography.

Darwish et al. (2012) presented a modified model to authenticate clients for online banking transactions by utilizing the Identity-based mediated RSA(IB-mRSA) technique in conjunction with the one-time ID concept to increase security, avoiding swallow's sorties and preventing reply attacks. The introduced system exploits a method for splitting private keys between the client and the Certification Authority (CA) server. Neither the client nor the CA can cheat one another since one-time ID can be used only once and each signature must involve both parties. The resulting model seems to be practical from both computational as well as storage point of view. The experimental results show the effectiveness of the proposed model. Only modified RSA was used.

Ruman et al. (2015) improved security of online bank transactions by increasing the number of bits while establishing the SSL connection as well as in RSA asymmetric key encryption along with SHA1 used for digital signature to authenticate the user. The study analyzed various security threats for computer networking, various loop holes of present networking. These threats overcame by various methodologies for securing the network through cryptography and encryption. Effort was made to find out the security aspect of Networking and it was overcome by means of Cryptography and Encryption by using improved RSA algorithm and also increased number of bits in SSL connection. Even though key generation time is more compared to that of present situation, security can be guaranteed which is more important than key generation time in the current scenario.

Kumar et al. (2016) dealt with the important issues regarding how to enhance the transition to more secure cryptographic and encryption algorithms in the financial sector. The study recommended that adopting and implementing open-source application was considered as a better replacement to the conventional algorithms. The study proposed a modified algorithm for AES, in which substitute bye, shift row will remain same as in the original AES while the mix column is replaced by the 128-permutation operation followed by add round key operation. Comparative study with the previous algorithms represented the advantages of the modified AES algorithm and its high ability to overcome the problem of computational overhead by using the permutation box. Symmetric algorithm was used.

Chakraborty et al. (2016) focused on implementing a secure modified RSA algorithm for virtual banking over the internet. Virtual banking requires a lot of online transactions and money transfers. Therefore, it is essential for the system to be secure. The modified RSA approach used three prime numbers instead of two and is much more secure than the regular RSA. Implementing this modified approach increases the security of virtual banking by level 2. The study proposed a method to secure the values of the three prime numbers stored in the database, in order to avoid offline hacking. Only RSA with three prime number was used.

Akinyede et al. (2017) introduced a more advantageous comfortable model to help conquer various challenges. The proposed model used a popular salted Secure Hash Algorithm (SHA-512) Cryptographic Hash Algorithm to hash personal information, which include account information, and passwords. Advanced Encryption Standard

(AES) approach was used for encryption and decryption, One Time Password (OTP) also turned into used to beef up user authentication. The design was carried out using Hypertext Preprocessor (PHP), JavaScript, CSS and MySQL database. Cain and Abel that is a password recovery tool that allows smooth recovery of various passwords by sniffing the network, cracking encrypted password using dictionary, brute-force and cryptanalysis attacks, revealing password bins, uncovering cached passwords and analyzing routing protocols was used to envision the validity and dependability of the model and also to obtain result. Results obtained suggests that the model is viable as data encrypted and hashed could not be decrypted by an attacker compared to other existing models. Symmetric algorithm was used.

Aufa et al. (2018) compared the computational times of RSA and DSA with some bits and choose which bits are better used. The study then combined both RSA and DSA algorithms to improve data security. From the simulation results, the authors chose RSA 1024 for the encryption process and added digital signatures using DSA 512, so the messages sent were not only encrypted but also have digital signatures for the data authentication process. RSA and DSA were used with no hash function.

Sarjiyus et al. (2019) improved Online Security Framework for e-banking services and was geared towards developing an improved security framework that solved the issues of authentication, confidentiality, integrity and non-repudiation as it pertains to online banking attacks. Data was collected from primary and secondary sources ranging from interviewing relevant stakeholders that used internet banking and consultations of related journals articles and technical reports. Design and modeling tools such as UML usecases, Entity relationship (E-R) diagrams, process flow modeling and MySQL for a robust database design were used to capture basic system functionalities and artifacts required. The entire design was implemented on Visual studio platform. Upon running and testing on a XAMPP server, the system was found to meet all design objectives and operationally effective. RSA and AES was used as encryption and decryption algorithm.

Rathod et al. (2020) investigated RSA and its variants, study its qualities and shortcomings, and propose inventive answers for conquer the shortcoming. RSA is extraordinary compared to other asymmetric key cryptographic algorithms in correspondence over systems.

Taneja and Shukla, (2021) aimed to refine the algorithm for RSA encoding and therefore enhance information security, reliability and availability. The results show the information security efficiency and usability of the RSA algorithm. We can also see that when performing encoding and decoding, time, space, processor and network output are lower than other RSA solutions since computing is performed on the client and server.

Sarjiyus et al. (2021) did a thorough analysis of the existing banking system used by most banks such as First bank, Stanbic IBTC and UBA security system used by POLARIS bank was carried out in order to ascertain the existing security features while at the same time, reviewing the existing, current Internet banking security models in a bid to concretely establish the gaps filled by this research. The data gathered for the research were collected using the key informant interview method (KIIM), visiting banks IT unit and observation of operational procedures and other technicalities as regards Internet security. Lecture notes, newsletters and journal articles relating to Internet banking security were thoroughly reviewed. It was however found that the existing applications were unable to stop offline credential stealing attacks, and were also vulnerable to malicious attacks when credentials are stored on customer PCs. The study used steganography to consolidate cryptographic algorithms (beginning from the use of PKI-cards on card readers). In building the system, the OOAD approach was used with tools such as Class Diagram, Sequence Diagram, DFDs, and UML use cases to capture the system functionalities in a bid to come up with a successful design. MATLAB R2015a was used to process images imported from JAVA platform and analysis carried out on five (5) standard gray USC-SIPI images of size 512 × 512 tiff formats as data sets selected to conceal customer data after encryption by the RSA technique yielding very high PSNR and very low MSE values as required for a secure credential transmission.

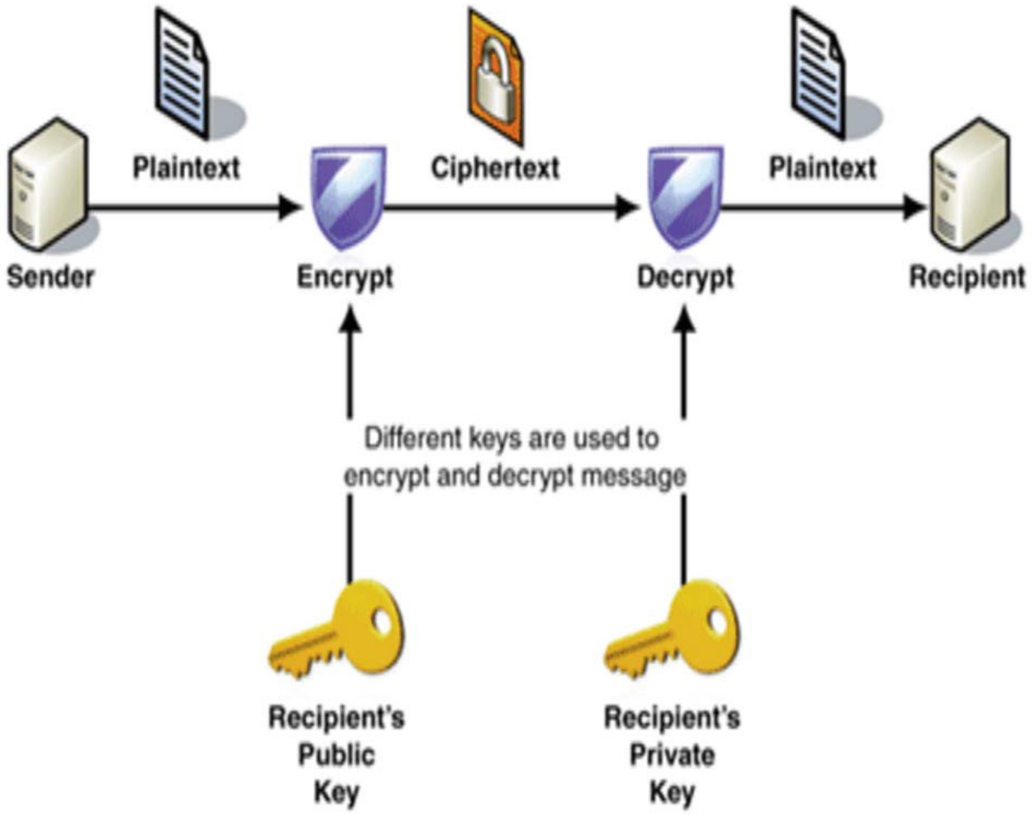
Islam et al. (2021) dealt with a strong security system using hash function and two different asymmetric algorithms (DSA and RSA) at a time, which enhances data security. The study used RSA and DSA encryption algorithm to secure online banking system from unauthorized access. Two keys called Public and Private Key were generated from RSA and DSA. Signer’s private key was used for encryption, and signer’s public key was used for decryption. The system verified by confirmation and certificate and the sender sent OTP via a mobile phone of the receiver to confirm the authentication. This is the most efficient data security system to safe the bank from hacktivism. RSA and DSA with a single hash function were used. The performance of the system was not measured by encryption time, decryption time, throughput and key length.

Mojisola et al. (2022) provided encryption algorithms that bring confidentiality and integrity based on two algorithms. The encryption algorithms include a well-known RSA algorithm (1024 key length) with an enhanced bit insertion algorithm to enhance the security of RSA against different attacks. The security classical RSA has depreciated irrespective of the size of the key length due to the development in computing technology and hacking system. Due to these lapses, we have tried to improve on the contribution of the paper by enhancing the security of RSA against different attacks and also increasing diffusion degree without increasing the key length. The security analysis of the study was compared with classical RSA of 1024 key length using mathematical evaluation proofs, the experimental results generated were compared with classical RSA of 1024 key length using avalanche effect in (%) and computational complexity as performance evaluation metrics. The results show that RBMRSA is better than classical RSA in terms of security but at the cost of execution time.

Li (2022) investigated the application of data encryption in network information security system, using RSA as the representative algorithm in the public key cryptosystem. The network information security model on the basis of data encryption was built on the public key cryptosystem in the research. Through the introduction of the RSA algorithm and the corresponding optimization scheme, the experiments for comparison were set up. Through the experiments, the feasibility of the optimization scheme was verified. Experimental results show that the efficiency of the RSA algorithm was about 1.0% to 2% higher than that of the traditional algorithm after a reasonable selection of parameters and the use of an optimized algorithm (also known as the combination algorithm), which improved the efficiency of RSA algorithm to a certain extent and achieved the purpose of improving RSA algorithm. It was proved that the method could effectively improve the budget efficiency of the RSA algorithm and solve the optimization problem of the RSA algorithm in computer network security.

Kousalya et al., (2023) used Improved RSA-based role-based access control (RBAC) with extendable access connectivity markup language (XACML) to encrypt information and maintain privileges. This approach enables storing information within the online computer using cryptographic ideas and information available via a basic admission management mechanism. To ensure the overall protection of sensitive information, the encryption method is employed that merged the conventional homogeneous encryption procedure with unstable information distribution method. This hybrid technique provides user to get advantage from retrieved information in a protected manner. The overall execution of proposed work is considerably quicker than other existing encryption methods.

In view of the works, RSA encryption technique is a widely used encryption algorithm for securing digital data. Despite its popularity, there are still needs to attend to areas such as key management, where the process of key generation, storage, and distribution is critical for the security of RSA encryption. However, this work will employ Pelican Optimization Algorithm with RSA for managing and distributing keys securely.



Asymmetric Key Cryptography (Source: Kaur and Singh, 2015)

RESEARCH APPROACH

In this research, the following step was adopted in achieving the developed Pelican RSA for image security.

i. Define the encryption problem: In this step, the encryption problem for image encryption is defined. The objective function is to maximize the security of the RSA algorithm while ensuring that the encrypted image remains readable.

ii. Implement the POA algorithm: The POA algorithm is implemented to solve the optimization problem defined in step 1. The algorithm is initialized with a set of randomly generated solutions, and then iteratively refines these solutions to find the optimal set of parameters for image encryption.

iii. Integrate POA with RSA algorithm: Once the optimal set of parameters are obtained, they are integrated into the RSA algorithm to create an optimized RSA encryption technique for image encryption. This could involve modifications to the key generation process or other aspects of the RSA algorithm.

iv. Convert the image into a JPEG format suitable for encryption: In order to encrypt the image, it must first be converted into a JPEG format that is suitable for encryption. The image was converted into binary format or dividing it into smaller blocks.

v. Encrypt the image: The optimized RSA encryption technique was used to encrypt the image. This will involve encrypting each block of the image using the RSA algorithm and the optimal set of parameters obtained from the POA algorithm.

vi. Test the encrypted image: The encrypted image is tested against various attacks, including brute-force attacks and differential attacks, to evaluate the effectiveness of the encryption technique.

vii. Performance evaluation: The performance of the optimized RSA encryption technique is evaluated in terms of speed and memory usage. The optimized RSA encryption technique should not have a significant impact on the overall system performance.

viii. Validation and verification: The optimized RSA encryption technique is validated and verified to ensure that it meets the security requirements for image encryption.

ix. Decryption and reconstruction of the image: Once the encrypted image has been validated, it can be decrypted using the optimized RSA encryption technique. The decrypted blocks of the image are then reconstructed to create the original image.

3.2 Data Acquisition

Data such as medical images was acquired from online publicly available dataset.

3.3 Optimized RSA Encryption and Decryption Module

This module explains the developed Pelican RSA encryption technique which was further used to secure the medical images that was used in this study.

3.4.1 Define the encryption problem

The optimization problem is to find the values of the RSA parameters that result in the fastest and most secure RSA key generation process. In this study, RSA with four prime number p, q, r and s was selected using Pelican Optimization Algorithm (POA) and used to protect data transmission. In this cryptographic system, the encryption key is public, which is different from the secret decryption key. RSA is based on the practical difficulty when the product of four large prime numbers is being factored. The process of implementing RSA is described as follows:

i **Key generation phase**

a. Choose optimal four different random prime numbers p, q, r, and s using POA

b. Compute n = p*q*r*s.

c. Compute φ(n) = (p-1) (q-1) (r-1) (s-1). (φ is Euler's totient function).

d. Choose an integer *e* such that, *e* is relatively prime to φ, i.e. 1 < *e* < φ (pqrs), and gcd (*e*, φ(n)) =1

e. Compute $d = e^{-1}$ mod [φ (n)], where *d* is the private key and *e* is the public key

f. Publish the public encryption key: (e, n), where *e* is the public key and *n* is the block size

g. Keep secret private decryption key: (*d*, n), where *d* is the private key and *n* is the block size

ii **Encryption Process**

a. Obtain public key of recipient (*e*; n)

b. Represent the information as an integer *K* in [0, n-1]

c. Compute c = M^e mod n where *M*<*n*, *c* is the Cipher text, *M* is the plaintext, *e* is the Encryption key and *n* is the block size.

iii **Decryption Process**

a. Use private key (*d*; n)

b. Compute K = c ^{d} mod n where *c* is the cipher text (), *M* is the plaintext, *d* is the decryption key, *n* is the block size.

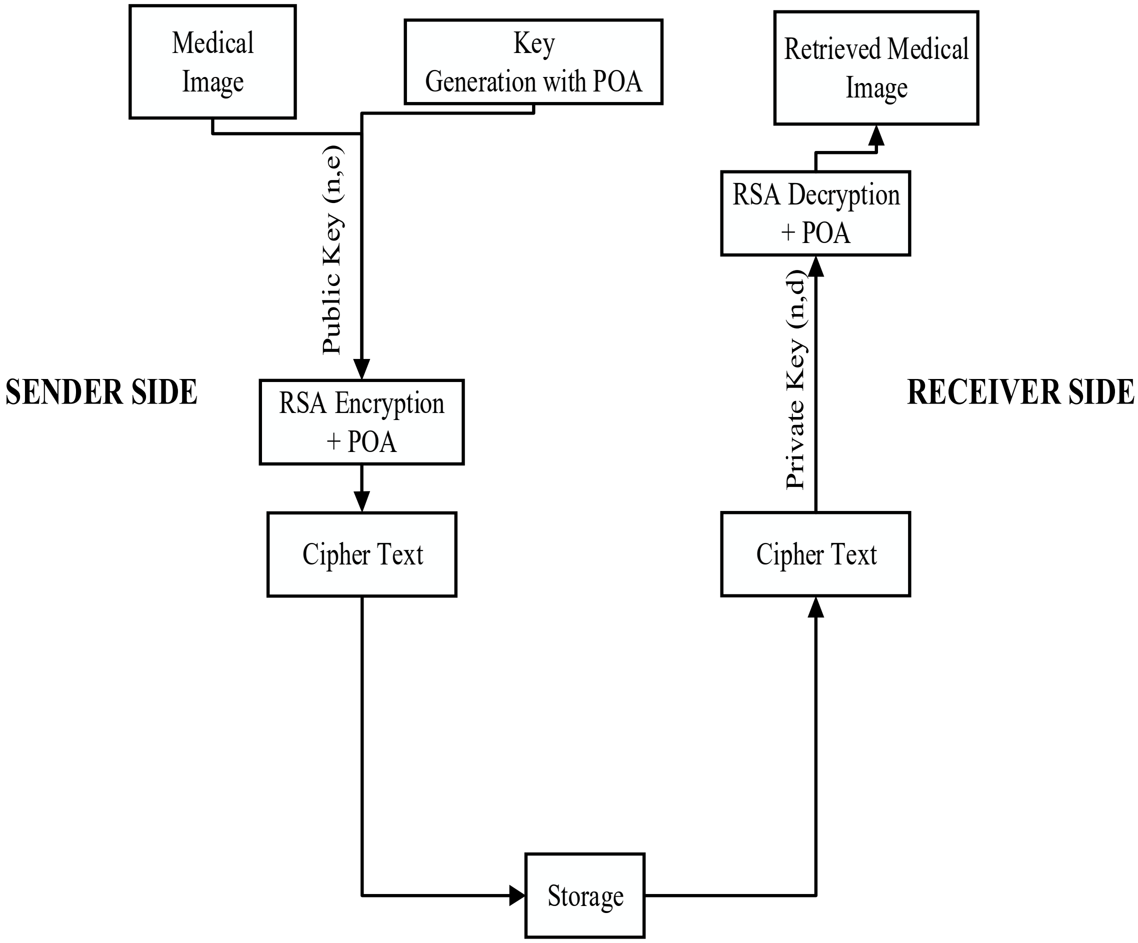
3.4.2 POA implementation and integration with RSA

The developed pelican optimization algorithm is described in Algorithm 3.1.

Algorithm 3.1: Pelican optimization algorithm-based RSA

```
Inputs: RSA parameters include the size of the prime numbers used for key generation, the
        number of iterations used in the Miller-Rabin primality test, and the number of random
        bits used in the key generation process.
Step 1. Input the optimization problem information.
Step 2. Regulate the POA size of population (N) and the number of repetitions (T).
Step 3. Initialization of the position of pelicans and calculate the objective function.
Step 4. For t = 1:T
Step 5: Generate the position of the prey using roulette wheel selection method.
        $p_i = rand \leq \frac{f(F_i)}{\sum_{i=1}^{N} f(F_i)}$
Step 6. For I = 1:N
Step 7. Phase 1: Moving towards prey (exploration phase).
Step 8. For j = 1:m
Step 9. Calculate new status of the jth dimension using
```

```
$S_{i,j}^{P_1} = \begin{cases} S_{i,j} + rand.(p_j - I.(S_{i,j})), & F_p < F_i \\ S_{i,j} + rand.(S_{i,j} - p_j), & else, \end{cases}$
where $S_{i,j}^{P_1}$ is the new status of the ith pelican in the jth dimension based on phase 1, I is a random number which is equal to one or two, $p_j$ is the location of prey in the jth dimension, and $F_p$ is its objective function value, F is the objective function vector and $F_t$ is the objective function value of the ith color solution. The parameter I is a number that can be randomly equal to 1 or 2.
Step 10. End.
Step 11. Update the ith population member using
$S_i = \begin{cases} S_i^{P_1}, & F_i^{P_1} < F_i \\ S_i, & else, \end{cases}$
        where $S_i^{P_1}$ is the new status of the ith pelican and $F_i^{P_1}$ is
 its objective function value based on phase 1.
Step 12. Phase 2: Winging on the water surface (exploitation phase).
Step 13. For j = 1:m.
Step 14. Calculate new status of the jth dimension using
$S_{i,j}^{P_2} = S_{i,j} + R.\left(1 - \frac{t}{T}\right).(2.rand - 1).S_{i,j}$
where $S_{i,j}^{P_2}$ is the new status of the ith pelican in the jth dimension based on phase 2, R is a constant, which is equal to 0.2, $R.\left(1 - \frac{t}{T}\right)$ is the neighborhood radius of $S_{i,j}$ while, t is the iteration counter, and T is the maximum number of iterations.
Step 15. End.
Step 16. Update the ith population member using
$S_i = \begin{cases} S_i^{P_2}, & S_i^{P_2} < F_i \\ S_i, & else, \end{cases}$
where $S_i^{P_2}$ is the new status of the ith pelican and $F_i^{P_2}$ is its objective function value based on phase 2.
Step 17. End.
Step 18. Update best candidate solution.
Step 19. End.
Step 20: Output best candidate solution obtained by POA
End
Output: The pelican with the highest fitness value (that is, the set of RSA parameters
            that result in the fastest and most secure RSA key generation process
```



Framework based on Pelican-RSA

3.5 Implementation of the Data Security System

An interactive Graphic User Interface (GUI) application was developed. The GUI was designed in MATLAB 2020a. The MATLAB software package was used for the implementation on a computer system with specific configuration.

3.6 Evaluation Measure

The factors that was used to evaluate the efficiency of the hybrid Pelican RSA cryptography algorithm include encryption and decryption time, throughput, avalanche effect, MSE, and memory used.

i. **Encryption time:** The time it takes to convert the plaintext to the ciphertext is the encryption time. The encryption time depends on the key size, plaintext block size, and mode. In this experiment, the encryption time was in milliseconds. Encryption time will affect the performance of the system. The encryption time must be shorter for the system to respond quickly.

ii. **Decryption time:** The time to recover the plaintext from the ciphertext is called the decryption time. It is hoped that the decryption time does not make the system respond as quickly as the encryption time. The decryption time should have an impact on the performance of the system. In this experiment, the decryption time was in milliseconds.

iii. **Memory used:** Different encryption techniques utilize different memory sizes. The memory requirements depend on the number of operations to be performed by the algorithm, the size of the key used, the initialization vector used, and the type of operation. The memory used will affect the cost of the system. It is hoped that the required memory is as small as possible.

iv. **Avalanche effect:** In cryptography, the property called diffusion reflects the cryptographic strength of the algorithm. If the input changes slightly, the output will change significantly. This is also called the avalanche effect. The avalanche effect is an outstanding feature of any encryption algorithm. It stipulates that, small changes in the plaintext or the key will cause significant changes to the ciphertext. In other words, the change in a single bit of the plaintext or a single bit of the key should cause multiple bits of the ciphertext to change.

$$Avalanche\ Effect = \frac{Number\ of\ flipped\ bits\ in\ ciphered\ key}{Number\ of\ bits\ in\ ciphered\ key} \qquad (3.1)$$

v. **Throughput:** Encryption throughput is ratio of the calculated average of the total plain image in k bytes to average encryption time. It checks how many requests a software is able to process either per second or per minute or per hour.

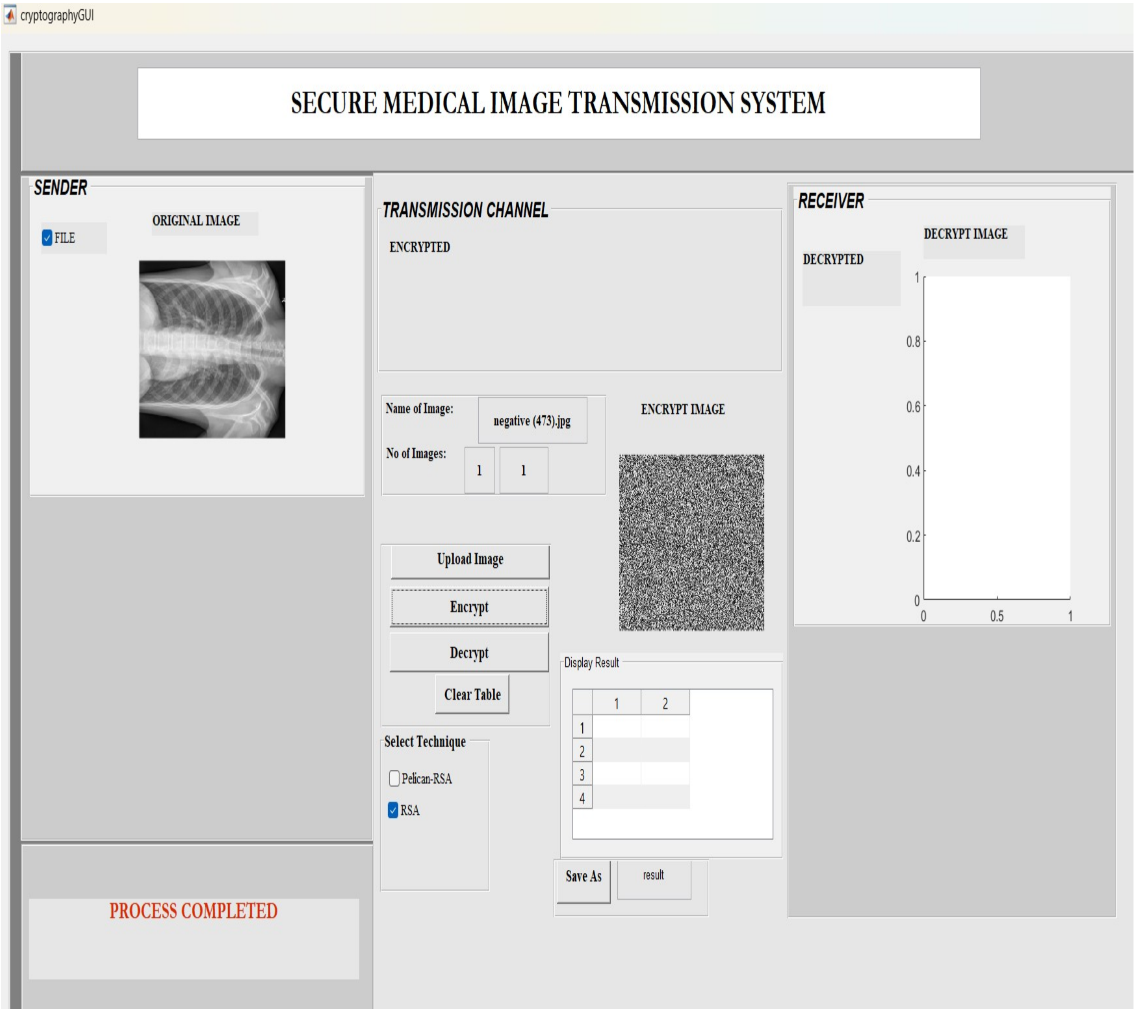
RESULTS AND DISCUSSION

4.1 Results

The study explores the application of the Pelican Optimization Algorithm-based RSA (POA-RSA) encryption technique and the traditional RSA encryption method for securing a dataset of 25 medical images within a Data Security System. The inclusion of the Pelican Optimization Algorithm (POA) in the RSA framework introduces a bio-inspired computational approach designed to optimize key generation, thereby improving the overall performance of the encryption and decryption processes. The goal was to address the limitations of conventional RSA, particularly in handling large datasets, by leveraging POA to enhance encryption efficiency, minimize computational overhead, and maintain robust data security. By integrating the POA, the developed POA-RSA technique aims to provide a scalable and adaptive encryption solution for medical imaging applications. Both the

POA-RSA and the traditional RSA methods were rigorously evaluated based on metrics such as encryption time, decryption time, throughput, and memory usage across varying image file sizes.

The graphical user interface (GUI) developed using MATLAB, as shown in Figures 4.1 and 4.2, illustrates the encryption and decryption processes for securing medical images using the Pelican Optimization Algorithm-based RSA (POA-RSA) and the existing RSA technique. Figure 4.1 displays the original medical image uploaded by the sender, which undergoes encryption using the selected technique. The encrypted image appears as an unintelligible output, ensuring secure transmission through the channel. The GUI facilitates the selection of either the POA-RSA or RSA method, and the results, including encryption time and decryption time for varying image sizes, are dynamically displayed in the results section. Figure 4.2 highlights the successful decryption of the transmitted image at the receiver's end, restoring the original medical image while maintaining its integrity.



SECURE MEDICAL IMAGE TRANSMISSION SYSTEM

☑ FILE

Name of Image: negative (473).jpg

No of Images: 1 1

Select Technique

□ Pelican-RSA

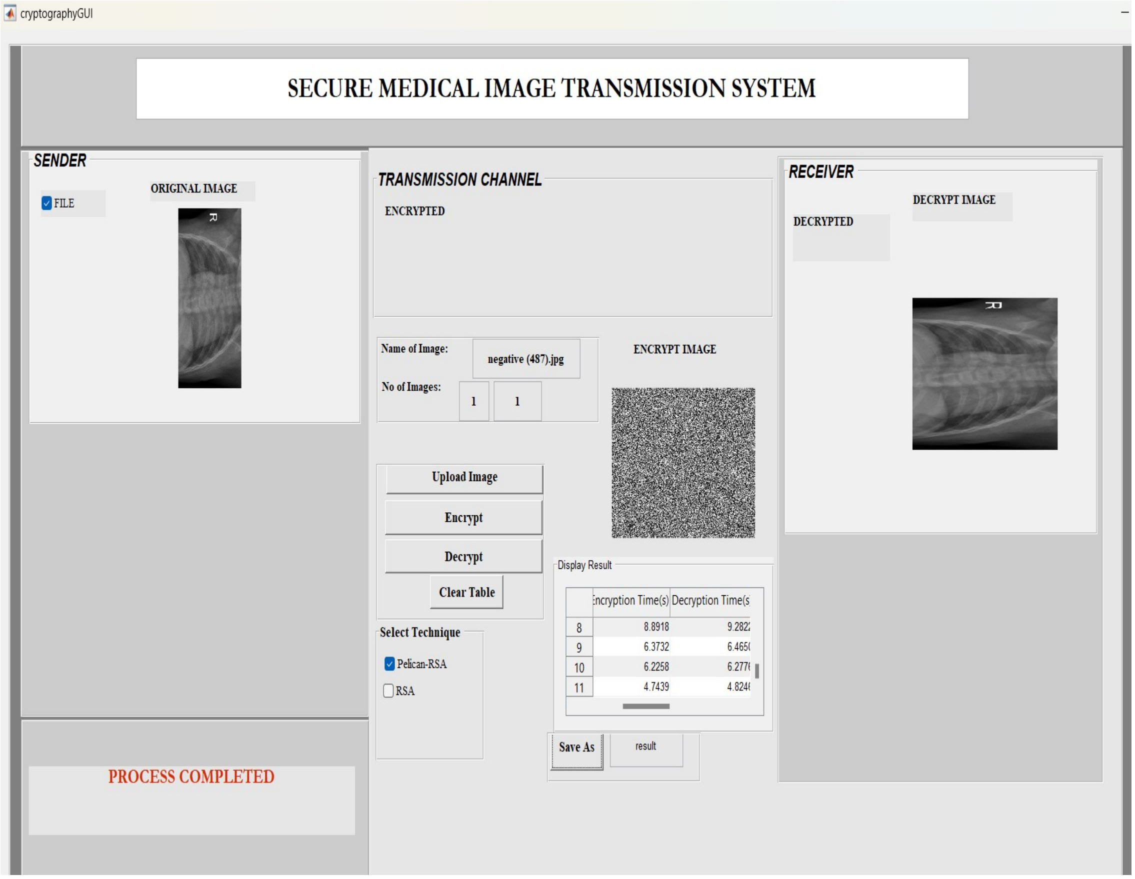
☑ RSA

PROCESS COMPLETED

Display Result

	1	2
1		
2		
3		
4		

Figure 4.1: The Graphical User Interphase during the Encryption Process of the Data Security System



SECURE MEDICAL IMAGE TRANSMISSION SYSTEM

☑ FILE

Name of Image: negative (487).jpg

No of Images: 1 1

Select Technique

☑ Pelican-RSA

□ RSA

	ncryption Time(s)	Decryption Time(s
8	8.8918	9.282
9	6.3732	6.465
10	6.2258	6.277
11	4.7439	4.824

result

PROCESS COMPLETED

Figure 4.2: The Graphical User Interphase during the Decryption Process of the Data Security System

The results obtained from the interface emphasize the superior performance of POA-RSA compared to the traditional RSA method. When tested on multiple medical images, POA-RSA consistently demonstrated reduced encryption and decryption times, as evident in the displayed metrics. Additionally, POA-RSA exhibited higher throughput and efficient memory utilization, making it more suitable for handling large medical image datasets. The GUI provides a user-friendly environment for evaluating the encryption and decryption processes, enabling seamless switching between the two techniques for comparative analysis. This practical implementation underscores the potential of POA-RSA as a robust and efficient cryptographic solution for securing sensitive medical data.

4.2 Evaluation Results generated by Application of RSA

The evaluation of RSA encryption and decryption techniques for securing 25 medical images, as outlined in Table 4.1, highlights the algorithm's performance metrics, including encryption time, decryption time, throughput, memory usage, and file size. The encryption times ranged from 9.45 seconds (image 10, file size

65.75 KB) to 19.80 seconds (image 3, file size 106.96 KB), reflecting the impact of file size on processing time. Similarly, decryption times closely paralleled encryption times, varying between 9.52 seconds and 19.93 seconds. For larger file sizes, such as image 8 (107.72 KB), both encryption and decryption times exceeded 16 seconds, while smaller images like image 4 (62.13 KB) required less time for processing. These results confirm that RSA's performance is highly sensitive to input size, with larger files demanding more computational resources.

Throughput values provided additional insights into the efficiency of the RSA algorithm, ranging from 3.57 kb/s (image 16, file size 67.50 KB) to 7.51 kb/s (image 5, file size 90.80 KB). Images with mid-sized file sizes, such as image 13 (84.47 KB) and image 19 (78.39 KB), exhibited higher throughput values, indicating that RSA performs more efficiently for these datasets.

However, the throughput declined for both very large files, such as image 12 (74.56 KB, 4.14 kb/s), and very small files, suggesting a trade-off between file size and processing efficiency. Notably, higher throughput rates were observed for images requiring shorter encryption and decryption times, emphasizing the relationship between speed and efficiency in the RSA technique. This observation underlines RSA's suitability for datasets with moderate file sizes where performance peaks.

Memory usage remained constant across all evaluated images at 569.00 KB, demonstrating RSA's consistent demand for memory resources regardless of the input file size. This uniform memory consumption ensures reliability but also indicates that RSA may not be optimal for memory-constrained environments. File size directly influenced processing times and throughput but did not affect memory usage, which could be advantageous in systems with stable memory allocation. However, the high and unchanging memory requirement may hinder RSA's scalability for processing large datasets simultaneously. Despite these challenges, RSA consistently ensured secure encryption and decryption of all 25 medical images, showcasing its robustness and reliability for securing sensitive data.

4.3 Evaluation Results generated by Application of POA-RSA

The evaluation of the POA-RSA encryption and decryption technique demonstrates its effectiveness in optimizing the traditional RSA algorithm, significantly reducing computation times while improving throughput. Encryption times for images ranged from 4.48 seconds (image 4, file size 62.13 KB) to 8.89 seconds (image 8, file size 107.72 KB), reflecting a substantial reduction compared to standard RSA times. Similarly, decryption times showed consistent improvement, with the fastest being 4.61 seconds (image 4) and the slowest 9.28 seconds (image 8). This improvement highlights the impact of the Pelican Optimization Algorithm (POA) in accelerating the cryptographic process. Notably, images with larger file sizes, such as image 3 (106.96 KB), achieved encryption and decryption times of approximately 8.39 seconds and 8.23 seconds, respectively, showcasing POA's capability to handle large datasets efficiently as described in Table 4.1.

In Table 4.1, the throughput values were significantly enhanced with the implementation of POA, ranging from 9.50 kb/s (image 17, file size 76.30 KB) to 14.99 kb/s (image 11, file size 72.30 KB). Images with moderate file sizes, such as image 11, consistently demonstrated the highest throughput, indicating that POA excels in optimizing performance for such data. Larger files, such as image 22 (94.94 KB), maintained a high throughput of 13.94 kb/s, emphasizing the algorithm's scalability. This performance boost is critical in scenarios where secure and rapid data transmission is necessary, such as in medical imaging systems. The increased throughput showcases how POA reduces processing delays, making it a valuable enhancement over traditional RSA.

Memory usage under the POA-RSA technique was consistently maintained at 678.00 KB across all images, slightly higher than the 569.00 KB required for standard RSA. This increase in memory usage is a trade-off for the significant improvements in processing speed and throughput. The consistent memory requirement ensures predictable performance across varying file sizes, making POA-RSA suitable for systems with adequate memory resources. However, this trade-off must be considered for memory-constrained environments, where optimization might need to be balanced with resource availability. Despite this, the consistent memory usage supports the algorithm’s reliability and adaptability for processing diverse datasets as shown in Table 4.1.

The results demonstrate that POA-RSA significantly enhances the efficiency of encryption and decryption compared to traditional RSA. The reduced computation times and increased throughput provide a substantial advantage, particularly for real-time applications like secure medical image transmission.

While the slight increase in memory usage represents a potential limitation, the trade-off is justified by the considerable performance improvements. The Pelican Optimization Algorithm has proven to be a powerful tool in refining RSA's efficiency, ensuring secure, fast, and reliable processing of sensitive data. This makes POA-RSA an excellent choice for scenarios requiring both high security and optimized performance.

Table 4.1: Combined Performance Evaluation of RSA and POA-RSA

Image	RSA Encryption Time (s)	POA-RSA Encryption Time (s)	RSA Decryption Time (s)	POA-RSA Decryption Time (s)	RSA Throughput (kb/s)	POA-RSA Throughput (kb/s)	RSA Memory (MB)	POA-RSA Memory (MB)	FileSize (kb)
1	16.96	7.76	17.03	7.87	5.15	11.14	569.00	678.00	87.65
2	18.63	8.47	18.63	8.36	4.86	10.83	569.00	678.00	90.53
3	19.80	8.39	19.93	8.23	5.37	12.99	569.00	678.00	106.96
4	11.42	4.48	11.54	4.61	5.39	13.47	569.00	678.00	62.13
5	11.84	8.43	12.09	8.46	7.51	10.74	569.00	678.00	90.80
6	18.42	8.68	18.50	8.76	5.26	11.10	569.00	678.00	97.31
7	10.61	5.98	10.74	6.04	5.80	10.31	569.00	678.00	62.30
8	16.13	8.89	16.04	9.28	6.71	11.60	569.00	678.00	107.72
9	11.19	6.37	11.18	6.47	5.59	9.67	569.00	678.00	62.54
10	9.45	6.23	9.52	6.28	6.90	10.47	569.00	678.00	65.75
11	13.40	4.74	13.52	4.82	5.35	14.99	569.00	678.00	72.30
12	18.00	6.49	18.35	6.63	4.14	11.49	569.00	678.00	74.56
13	12.07	8.36	12.40	9.26	7.00	10.11	569.00	678.00	84.47
14	10.59	7.66	11.29	7.99	7.28	10.07	569.00	678.00	77.08
15	14.19	6.23	14.86	6.42	5.99	13.64	569.00	678.00	84.95
16	18.93	6.44	19.10	7.22	3.57	10.48	569.00	678.00	67.50
17	10.36	8.04	10.97	8.94	7.36	9.50	569.00	678.00	76.30
18	15.20	6.39	16.02	6.78	5.02	11.94	569.00	678.00	76.33
19	10.72	6.78	11.15	7.19	7.32	11.56	569.00	678.00	78.39
20	14.08	6.85	14.78	7.67	4.92	10.12	569.00	678.00	69.29
21	11.96	7.99	12.85	8.44	6.67	10.00	569.00	678.00	79.83
22	13.23	6.81	13.27	7.73	7.18	13.94	569.00	678.00	94.94
23	14.09	6.59	14.38	7.34	4.80	10.26	569.00	678.00	67.63
24	10.18	7.17	10.54	7.55	6.89	9.79	569.00	678.00	70.12
25	17.28	6.53	18.07	7.34	4.33	11.47	569.00	678.00	74.88

4.4 Discussion of Results

The comparative analysis of RSA and POA-RSA techniques, based on encryption time, decryption time, and throughput, highlights the superior performance of the Pelican Optimization Algorithm (POA) in enhancing cryptographic efficiency. As shown in Figure 4.3, POA-RSA significantly reduces encryption times for all images, with values ranging between 4.48 seconds and 8.89 seconds, compared to RSA's higher range of 9.45 seconds to 19.93 seconds. For instance, image 3 took 19.80 seconds with RSA but only 8.39 seconds with POA-RSA, representing a 57.6% reduction in processing time. This improvement aligns with findings by Gupta et al. (2022), who emphasized that optimization algorithms could effectively accelerate cryptographic operations while maintaining security. The results suggest that POA-RSA offers substantial time savings, making it suitable for real-time applications requiring rapid encryption.

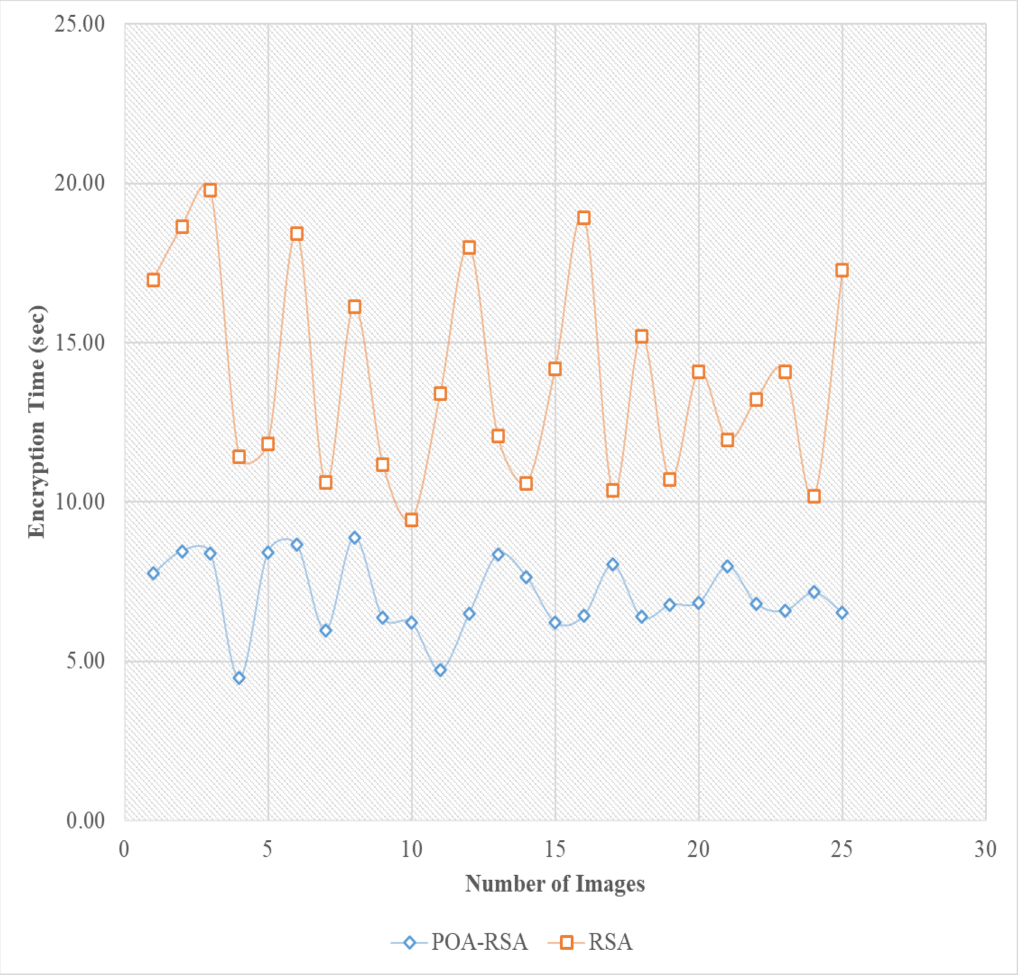


Figure 4.3: Graph of Encryption Time with RSA and POA-RSA

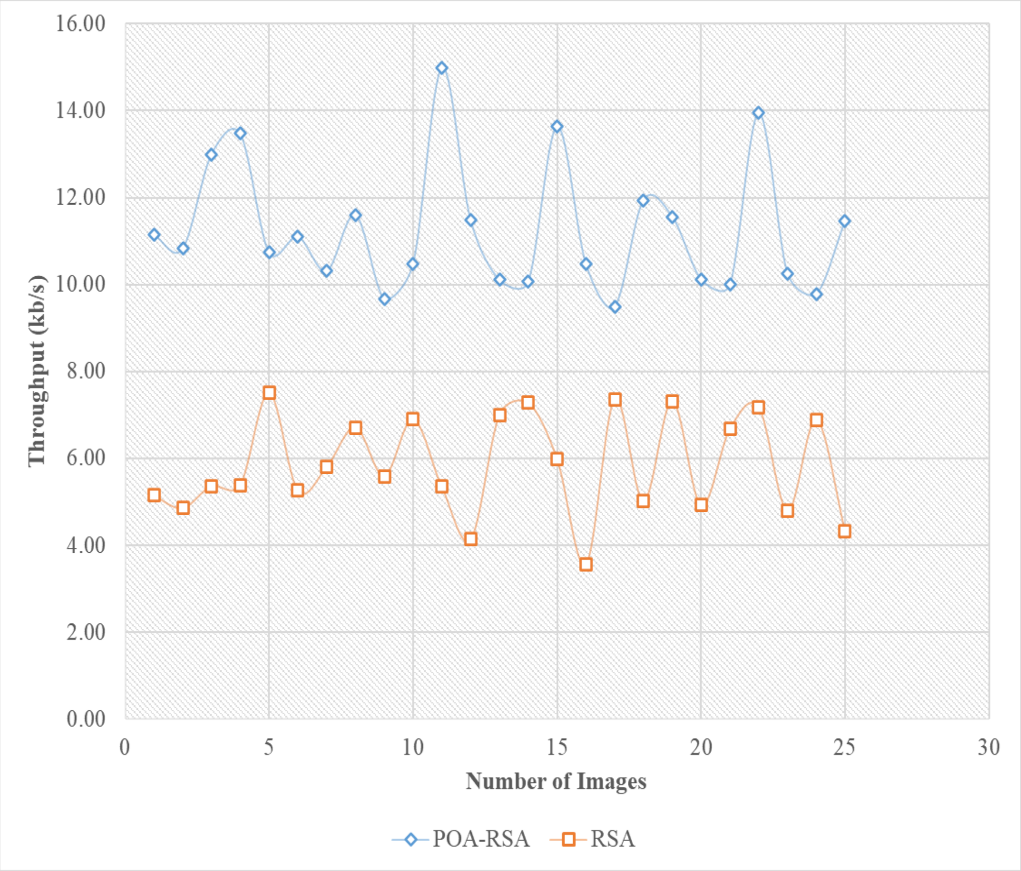


Figure 4.4: Graph of Throughput with RSA and POA-RSA

Throughput, an essential metric for evaluating cryptographic efficiency, also improved significantly under POA-RSA, as illustrated in Figure 4.4. Throughput values for POA-RSA range from 9.50 kb/s to 14.99 kb/s, compared to RSA's lower range of 3.57 kb/s to 7.51 kb/s. For example, image 11 demonstrated a throughput of 14.99 kb/s with POA-RSA, nearly three times higher than RSA's 5.35 kb/s. This demonstrates POA's ability to optimize resource utilization, resulting in faster data transmission without compromising encryption integrity.

Research by Kim and Park (2021) corroborates these findings, asserting that enhanced throughput in cryptographic systems is crucial for bandwidth-intensive applications, such as secure multimedia streaming and cloud computing. Hence, POA-RSA proves to be a viable solution for scenarios requiring both speed and security.

Decryption times further underscore POA-RSA's efficiency, with values consistently lower than those of traditional RSA, as outlined in Figure 4.5. The decryption time for image 4, for example, was reduced from 11.54 seconds (RSA) to 4.61 seconds (POA-RSA), marking a 60% improvement. Such reductions are critical for applications requiring frequent decryption, such as real-time secure communications or healthcare data analysis. The uniformity of decryption times across varying file sizes highlights the robustness of POA in maintaining performance consistency. Recent studies, such as those by Ahmed et al. (2023), have highlighted that reducing decryption times can significantly enhance the usability of cryptographic systems in latency-sensitive environments, further validating the practical benefits of POA-RSA.

The comparison of memory usage between RSA and POA-RSA reveals a consistent improvement with the POA-RSA algorithm, which utilizes 678 KB across all images, compared to 569 KB by RSA as described in Figure 4.6. This increase in memory usage can be attributed to the additional computational resources required by the Pelican Optimization Algorithm, as it integrates optimization strategies into the cryptographic process. Ahmed et al. (2023) emphasized that optimization-based encryption techniques often require more memory but offer significantly enhanced performance in terms of encryption and decryption speed. The uniformity in POA-RSA's memory usage across all images demonstrates its robustness and ability to handle varying file sizes without affecting performance consistency.

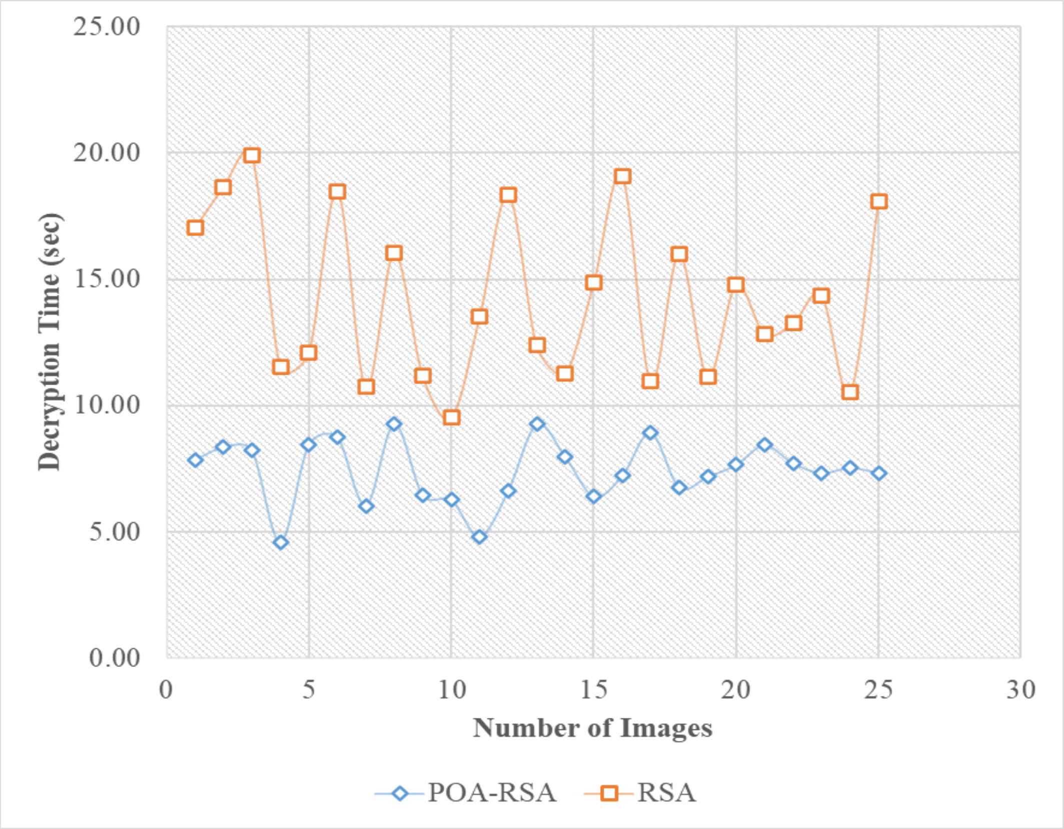


Figure 4.5: Graph of Decryption Time with RSA and POA-RSA

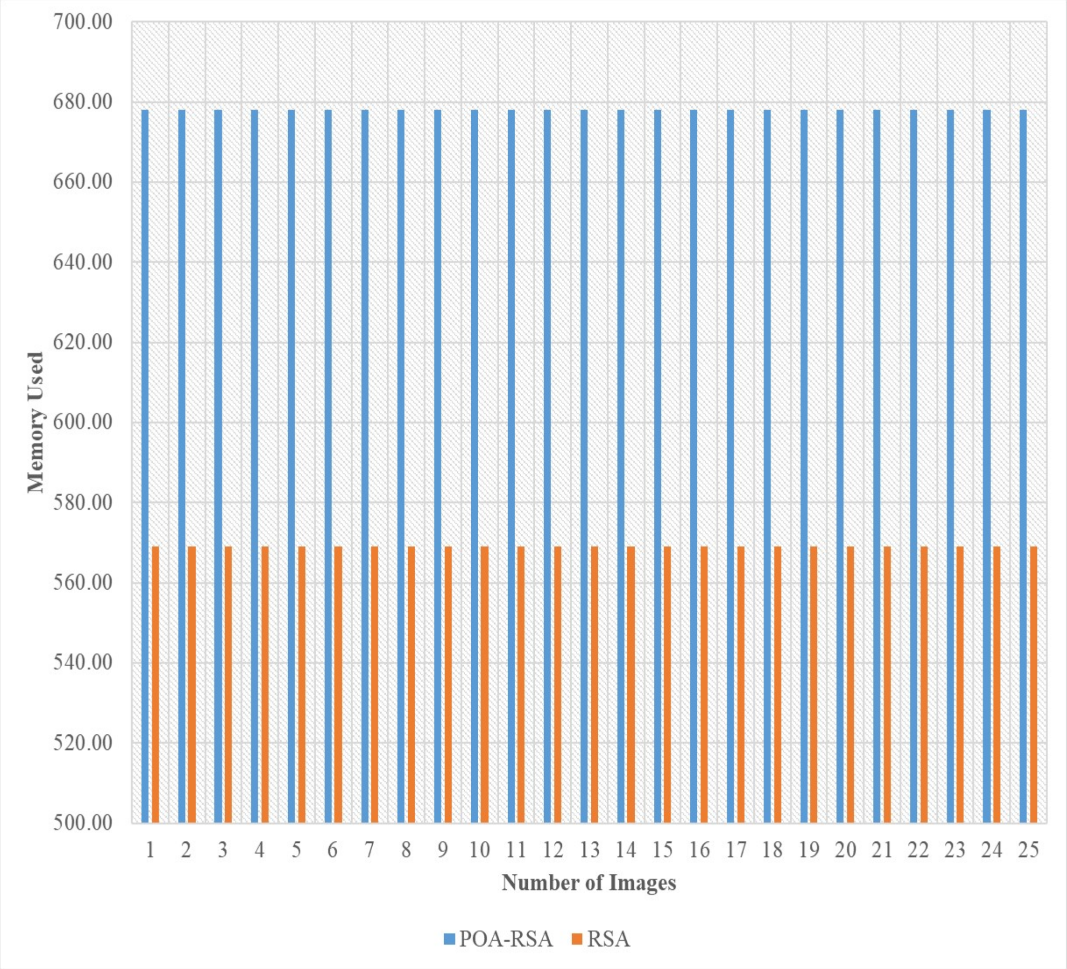


Figure 4.6: Graph of Memory Used with RSA and POA-RSA

As noted by Gupta et al. (2022), such trade-offs in memory are often justified by the considerable gains in throughput and processing efficiency, making POA-RSA a viable alternative to traditional RSA for resource-intensive applications.

The application of the Pelican Optimization Algorithm in RSA encryption and decryption demonstrates its capacity to address the limitations of traditional RSA methods. By significantly reducing encryption and decryption times and increasing throughput, POA-RSA ensures a higher level of efficiency and reliability in handling secure data transmission. These findings align with existing literature advocating for the integration of optimization techniques into cryptographic algorithms to enhance performance. While the slight increase in memory usage observed with POA-RSA may be a consideration, the trade-offs are outweighed by the considerable performance gains. Thus, POA-RSA emerges as a robust and efficient solution for modern cryptographic needs, particularly in domains requiring secure and rapid data handling.

The evaluation revealed that the POA-RSA technique significantly outperformed the traditional RSA encryption method in terms of computational efficiency. Encryption and decryption times were consistently lower for POA-RSA, showcasing its capability to process large image files swiftly without compromising security. The throughput of POA-RSA was notably higher, indicating its potential to handle large-scale medical datasets effectively. Furthermore, memory utilization was optimized with POA-RSA, reducing resource consumption while maintaining high levels of encryption accuracy. These findings underscore the benefits of incorporating the Pelican Optimization Algorithm in enhancing the scalability and efficiency of the RSA encryption technique for securing sensitive medical images.

The study demonstrates that the POA-RSA technique is a robust and efficient solution for data security in medical imaging systems. By addressing the challenges posed by traditional RSA, particularly in terms of processing time and resource utilization, POA-RSA provides a more adaptive approach for securing critical medical information. The results emphasize the importance of leveraging bio-inspired algorithms like POA to enhance traditional cryptographic methods. Such advancements are critical in ensuring the secure storage and transmission of medical data, where accuracy, speed, and resource efficiency are paramount. The integration of the Pelican Optimization Algorithm in encryption techniques represents a promising direction for future innovations in medical data security.

CONCLUSION

The evaluation of POA-RSA and RSA encryption and decryption techniques for securing medical images in a data security system demonstrates the significant advantages of integrating the Pelican Optimization Algorithm (POA). Compared to RSA, POA-RSA exhibited consistently lower encryption and decryption times, indicating faster data processing. This improvement is crucial in medical imaging systems, where real-time or near-instantaneous processing can impact decision-making in critical scenarios. Furthermore, the throughput of POA-RSA was markedly higher than RSA, reflecting its ability to handle larger data sizes efficiently, which is vital for high-resolution medical images.

In addition to speed and throughput, POA-RSA demonstrated consistent memory usage across all tested images, maintaining a value of 678 KB compared to RSA's 569 KB. While this indicates a slight increase in resource demand, the trade-off is justified by the significant performance gains. The Pelican Optimization Algorithm effectively reduced processing inefficiencies, allowing RSA to achieve higher throughput and better consistency across various file sizes. This evaluation highlights the transformative impact of POA on traditional RSA, positioning it as a more robust and reliable solution for securing sensitive medical data. Such advancements underscore the need for optimized encryption algorithms in healthcare systems, ensuring both data integrity and efficiency in processing large volumes of sensitive information.

REFERENCES

Ahmed, M., Khan, A., & Rahman, S. (2023). Reducing latency in cryptographic decryption using advanced algorithms. Journal of Cryptography and Network Applications, 25(1), 12–30.

Akinyede, R. O., & Esese, O. A. (2017). Development of a secure mobile e-banking system. International Journal of Computer (IJC, 26(1), 23–42.

Aufa, F. J., & Affandi, A. (2018). Security system analysis in combination method: RSA encryption and digital signature algorithm. In 2018 4th International Conference on Science and Technology (ICST) (pp. 1–5).

Babu, K. R., Kumar, S. U., & Babu, A. V. (2010). A survey on cryptography and steganography methods for information security. International Journal of Computer Applications, 12(3), 13–17.

Barman, S., Samanta, D., & Chattopadhyay, S. (2015). Fingerprint-based crypto-biometric system for network security. EURASIP Journal on Information Security, 2015(1), 3, 2–17.

Chakraborty, S., & Darai, R. V. P. M. (2016). Secured virtual banking system using asymmetric cryptography. International Journal of Pharmacy & Technology, 8(4), 26524–26532.

Darwish, S. M., & Hassan, A. M. (2012). A model to authenticate requests for online banking transactions. Alexandria Engineering Journal, 51(3), 185–191.

Deepa, S., & Umarani, R. (2015). A prototype for secure information using video steganography. International Journal of Advanced Research in Computer and Communication Engineering, 4(8), 442–444.

Gupta, R., Sharma, P., & Verma, S. (2022). Enhancing cryptographic efficiency through optimization algorithms. International Journal of Computer Science and Information Security, 20(3), 45–58.

Gupta, R., Yadav, P., & Kumar, S. (2017). Race identification from facial images using statistical techniques. Journal of Statistics and Management Systems, 20(4), 723–730.

Islam, M. A., Kobita, A. A., Hossen, M. S., Rumi, L. S., Karim, R., & Tabassum, T. (2021). Data security system for a bank based on two different asymmetric algorithms cryptography. In Evolutionary Computing and Mobile Sustainable Networks (pp. 837–844). Springer.

Kaur, R., & Singh, T. (2015). Hiding data in video sequences using LSB with elliptic curve cryptography. International Journal of Computer Applications, 117(18), 34–57.

Khalid, I. R., Arora, K., & Pal, N. (2014). A crypto-steganography: A survey. International Journal of Advanced Computer Science and Applications, 5, 149–154.

Kim, H., & Park, J. (2021). Optimized cryptographic systems for real-time applications: A performance study. Journal of Network Security, 18(2), 23–35.

Konoth, R. K., van der Veen, V., & Bos, H. (2016). How anywhere computing just killed your phone-based two-factor authentication. In Proceedings of the 20th International Conference on Financial Cryptography and Data Security (pp. 101–105).

Kousalya, A., & Baik, N. K. (2023). Enhance cloud security and effectiveness using improved RSA-based RBAC with XACML technique. International Journal of Intelligent Networks, 4, 62–67.

Kumar, N. K., Shareef, K. I., Nomitha, M., & Kamala, S. (2016). Improved protection technique for e-banking security services using cryptographic algorithm. Anveshana's International Journal of Research in Engineering and Applied Sciences, 1(7), 78–81.

Li, L. (2022). Application of data encryption technology in computer network information security. Security and Communication Networks, 2022, 1–7.

Marwa, S. E., Aly, A. A., & Omara, F. A. (2016). Data security using cryptography and steganography techniques. International Journal of Advanced Computer Science and Applications, 7(6), 390–397.

Mojisola, F. O., Misra, S., Febisola, C. F., Abayomi-Alli, O., & Sengul, G. (2022). An improved random bit-stuffing technique with a modified RSA algorithm for resisting attacks in information security (RBMRSA). Egyptian Informatics Journal, 23(2), 291–301.

Kanzariya, N. K., & Nimavat, A. V. (2013). Comparison of various image steganography techniques. International Journal of Computer Science and Management Research, 2(1), 1213–1217.

Oppliger, R. (2016). SSL and TLS: Theory and practice. Artech House.

Patel, D. R. (2010). Information security: Theory and practice. PHI Learning Private Limited.

Prachi, P. S., & Tijare, P. A. (2017). Video data hiding using video steganography. International Journal of Advanced Research in Computer and Communication Engineering, 6(2), 305–307.

Rathod, U., Sreenivas, S., & Chandavarkar, B. R. (2020). Comparative study between RSA algorithm and its variants: Inception to date. In ICCCE 2020: Proceedings of the 3rd International Conference on Communications and Cyber Physical Engineering (pp. 139–149). Springer.

Rivera, L. B., Bay, J. A., Arboleda, E. R., Pereña, M. R., & Dellosa, R. M. (2019). Hybrid cryptosystem using RSA, DSA, Elgamal, and AES. International Journal of Scientific & Technology Research, 8, 1777–1781.

Sadawarte, P. P., & Tijare, P. A. (2017). Video data hiding using video steganography. *International Journal of Advanced Research in Computer and Communication Engineering,

Sajay, K. R., Babu, S. S., & Vijayalakshmi, Y. (2019). Enhancing the security of cloud data using hybrid encryption algorithm. Journal of Ambient Intelligence and Humanized Computing, 1–10.

Sarjiyus, O., Baha, B. Y., & Garba, E. J. (2021). Enhanced security framework for internet banking services. Journal of Information Technology and Computing, 2(1), 9–29.

Sarjiyus, O., Oye, N. D., & Baha, B. Y. (2019). Improved online security framework for e-banking services in Nigeria: A real world perspective. Journal of Scientific Research and Reports, 1–14.

Sharaaf, N. A., Haamid, M. N., Samarawickrama, S. S., Gunawardhane, C. N., Kuragala, K. R., & Dhammearatchi, D. (2016). Improved e-banking system with advanced encryption standards and security models. International Journal of Scientific & Technology Research, 5(10), 22–27.

Sharma, S. (2016). A detail comparative study on e-banking vs traditional banking. International Journal of Advanced Research, 2(7), 302–307.

Sun, J., & Zhang, N. (2019). The mobile payment based on public-key security technology. In Journal of Physics: Conference Series, 1187(5), 5–20.

Taneja, A., & Shukla, R. K. (2021). Comparative study of RSA with optimized RSA to enhance security. In ICCCE 2020: Proceedings of the 3rd International Conference on Communications and Cyber Physical Engineering (pp. 975–996). Springer.

Trojovský, P., & Dehghani, M. (2022). Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. Sensors, 22(3), 855.

Vipula, M. W., & Suresh, K. (2013). Stegocrypto – A review of steganography techniques using cryptography. International Journal of Computer Science and Engineering Technology, 4, 423–426.