



Enhancing Military Communication Security Using Hybrid Cryptography and Steganography

Wickramapala M.H.T.K^{a*}, Yakupitiya K.C^b, Pathirana M^c

^a*ESOFT Uni, No.03 ,De Fonseka place , Colombo and 00400, Sri Lanka*

^b*University of Sri Jayawardenapura, Gangodawila, Nugegoda and 10100, Sri Lanka*

^c*University of Ruhuna, Wellamada, Matara and 81000, Sri Lanka*

^a*Email: thayuri.kaushalya@esoft.lk*

^b*Email yakupitiya@sjp.ac.lk*

^c*Email: pathiranam@dcs.ruh.ac.lk*

Abstract

The increasing threat of military communication interception emphasises the critical need for secure data transmission methods to safeguard highly sensitive information. This research investigates ways to enhance communication security in Sri Lanka's military by employing a hybrid system that integrates Advanced Encryption Standard (AES), Rivest Shamir Adleman (RSA), and Least Significant Bit (LSB) steganography. The aim of this study is to devise, implement, and evaluate a hybrid cryptographic system that utilises AES-256 for encryption, RSA-2048 for key exchange, and LSB steganography, focusing on improving the security and operational efficiency of military communications. The system was developed with the NetBeans Integrated Development Environment (IDE) and Java Cryptography Architecture (JCA), using AES-256 for symmetric encryption, RSA-2048 for managing asymmetric keys, and LSB steganography to embed the encrypted data in digital images. A performance evaluation was conducted to measure the execution times of encryption, decryption, and steganography processes, as well as the precision of data extraction, and the system's resilience against potential security threats. The results indicated that AES encryption provided fast and efficient data

protection, RSA managed secure key exchanges efficiently, and LSB steganography successfully concealed the encrypted information in images without causing noticeable visual differences. These findings confirm that the system offers a robust and secure approach to military communications, although the encoding time for steganography increased with larger image sizes. A proof of concept was developed to demonstrate and assess the proposed solution, incorporating only the vital elements needed to effectively showcase each phase of the secure communication process. This study enhances the current understanding by illustrating how hybrid cryptography and steganography can work together to meet the specific security requirements of military communications. This paper presents a feasible solution to reduce interception risks and has important implications for secure communication systems in the military sectors.

Keywords: Hybrid Cryptographic System; Military Communication Security; AES-256 Encryption; RSA-2048 Key Exchange; LSB Steganography

thayuri.kaushalya@esoft.lk

* Corresponding author. Email address:

1. Introduction

Critical flaws in traditional encryption methods used to protect sensitive defence communications have been revealed by recent cyberattacks and data breaches involving Sri Lanka's military agencies. For example, Sri Lanka state security agency exposed significant weaknesses in the country's cyber defence systems by allowing bad actors to access and disclose vital data [1]. These agencies' sensitive data was purportedly acquired and sold by a threat actor identified as "Kelvin Security". The 10.2 GB of compromised data included more than 12,000 files with names, residences, service numbers, and national identity card numbers of police. [2]. Unauthorised access to a military portal allowed the breach to occur, exposing flaws in the current cybersecurity setup.

The urgent necessity to implement next-generation data protection policies and strengthen national resilience against cyber warfare has been emphasised by events such as the Defence Cyber Command's Aegis Lexicon 2024 – Defence Cyber Exposition in response to these problems [3]. Beyond traditional perimeter protection and encryption, government authorisation agencies like Sri Lanka Computer Emergency Response Team (CERT) and specialists in digital infrastructure are still advocating for novel approaches to cybersecurity[4]. Despite employing hybrid cryptographic techniques that merge symmetric methods like Advanced Encryption Standard (AES) with asymmetric methods such as Rivest Shamir Adleman (RSA) encryption to protect sensitive information, a big problem remains: the symmetric key is exposed during transmission, even when encrypted. If encryption keys are captured or private RSA keys are compromised, the entire system can be compromised.

To combat these growing threats, encryption methods have become an essential component of data security strategies. In this area, steganography and hybrid encryption techniques have emerged as powerful technologies that offer unique advantages in maintaining both data integrity and confidentiality [5]. As the dependence on modern cloud systems and remote deployment settings grows, sensitive information is frequently kept in places that are not directly under administrative oversight [6]. This change has raised serious concerns about data breaches and unauthorised access to systems, highlighting the need to implement advanced security protocols to ensure strong and thorough protection of personal information.

Steganography, the technique of concealing information in an otherwise harmless carrier document, has a long-standing history that spans centuries, with historical uses that include clandestine communication during conflicts and the secure sharing of sensitive data. In addition to protecting the privacy of the information, this insertion provides an additional layer of concealment, complicating the task of detecting or obtaining the information by unauthorised individuals.

Unlike traditional methods, hybrid encryption takes advantage of the advantages of both symmetric and asymmetric encryption techniques [7]. Symmetric encryption is well-known for its rapid data processing capabilities and efficient use of computational resources [8]. It is typified by the usage of a shared secret key. In contrast, asymmetric encryption enhances security by separating keys and uses a pair of encryption keys (public and private) to establish a secure communication channel [9, 10]. Hybrid encryption, which combines these two approaches, produces a strong cryptographic architecture that successfully strikes a balance between security and performance.

Although Steganography and hybrid cryptography both offer notable security enhancements, they also have drawbacks. Symmetric key exposure during transmission makes traditional hybrid encryption schemes, like AES + RSA, vulnerable [11]. Standard cryptography raises the risk of targeting by warning attackers when sensitive data is present. Similarly, if unencrypted, Least Significant Bit (LSB) steganography is vulnerable to attacks such as statistical analysis and is therefore ineffective when used alone [12]. In order to close these gaps and improve the secrecy and integrity of military communications, this study suggests a multi-layered security framework that combines RSA-2048, AES-256, and LSB steganography to safeguard the encryption key as well as the contents.

The remainder of the document is structured as follows: Introduction was discussed in section I, materials and methods are presented in section II, section III presents the results followed by section IV discussing the results.

1.1. Literature Review

A few studies initiatives have been undertaken globally in security agencies in the domains of steganography, hybrid cryptography, and data security, aiming to enhance the confidentiality, integrity, and resilience of sensitive military communications [13, 14]. Several studies have highlighted the benefits of steganography in hiding communications, with some noting that it adds an extra layer of security, especially when used in conjunction with encryption [15]. While certain research has investigated the use of steganography in cybercrime and military domains [16, 17], these studies have limitations, especially regarding their payload capacity, which renders them inappropriate for extensive data transfers.

Some of studies have shown the effectiveness of hybrid cryptography, especially in military applications, where RSA-AES frameworks merge the speed of symmetric encryption (AES) for large-scale data encryption with the secure key distribution strengths of asymmetric encryption (RSA), offering an extra level of security [18, 19]. AES is recognised for its fast performance and ability to effectively encrypt large data sets, which makes it suitable for military applications due to its speed, low resource consumption, and strong security [20]. Several studies have shown that AES is more efficient than other symmetric encryption techniques in terms of speed and throughput [7, 18]. However, a significant drawback of AES is that while it protects against conventional attacks, it is vulnerable to emerging quantum threats. As well, as a symmetric encryption method, AES encounters considerable

difficulties with secure key distribution. These shortcomings emphasise the necessity of combining hybrid encryption models and steganography to improve the overall protection of sensitive information. Although some of studies have highlighted the advantages of hybrid cryptography ministry of defence services and cybercrimes [18, 21]., there are significant limitations in its application such as remain vulnerable to quantum attacks, Public Key Infrastructure (PKI) weaknesses, certificate authority compromises, and data interception. These limitations can undermine the security of encrypted communication and potentially violate confidentiality. Specifically, A few of studies in Sri Lanka focused on securing communication by combining steganography and visual cryptography to enhance data security by adding an extra layer of protection to concealed information, although they did not employ LSB steganography, which is typically used to hide data in the least significant bits of pixel values without perceptible changes to the image [22, 23]. While a large portion of these studies focus on individual techniques, only a limited number of studies attempt to merge cryptography and steganography to achieve a higher level of secure and secret communications.

However, these studies failed to address key challenges in hybrid cryptography management, including complexity, performance overhead, and interoperability among algorithms, which, combined with limitations in payload size and vulnerability to steganalysis tools in high-surveillance environments, can lead to inefficiencies in secure communication; although combining steganography with encryption strengthens covert communication by making encrypted data harder to detect, techniques like LSB steganography are particularly effective as they embed data into the least significant bits of images without perceptible quality loss.

Although there are studies on hybrid cryptography and steganography individually, there is a scarcity of research integrating hybrid cryptography with LSB steganography, particularly within military sectors in Sri Lanka. This study aims to mitigate these limitations by combining hybrid cryptography with LSB steganography to enhance military communication security. In this study, RSA is used to securely encrypt the AES key, ensuring that only the intended recipient can decrypt it using their private key. While the key remains encrypted, it is also hidden in the image through steganography to provide an additional layer of security, increasing its resistance to potential quantum attacks that could reveal RSA encrypted information. By embedding the AES-256 key within a cover image encrypted with RSA-2048, this approach addresses payload capacity, detectability, and steganalysis vulnerabilities. LSB steganography minimises file alterations, making the hidden data more resilient against surveillance and active attacks, while overcoming common hybrid cryptography limitations such as key management and processing overhead. Table 1 shows limitations of the existing studies.

Table 1: Limitations of the existing studies

Study	Focus Area	Key Findings	Limitations
[14] [13] [24]	Military data security; hybrid models	Emphasises the need for secure, resilient communications in military applications	Lack of joint implementation for military-specific use cases
[16] [15] [25] [17]	Steganography	Steganography adds a layer of security and effectively facilitates secret communications	Payload capacity limitations make it unsuitable for large data
[18] [21] [19]	Hybrid cryptography (AES + RSA) in military sectors	RSA +AES model ensures efficient encryption + secure key distribution	Vulnerable to quantum threats, PKI weaknesses, and interoperability issues
[7]	AES performance in secure systems	AES is fast, efficient, and suitable for large datasets	AES alone faces key distribution challenges and lacks quantum resistance
[22]	Steganography + visual cryptography in Sri Lanka	Explored combining techniques for stronger data concealment	Did not use LSB; lacked integration with hybrid encryption
[18] [21]	Hybrid cryptography in defence & cybercrime	Highlight use in government and sensitive sectors	Missed addressing processing overhead, quantum risk, and PKI flaws
General highlighted synthesis	gap in Combining LSB steganography with hybrid encryption techniques.	It is recommended to combine AES-256 and RSA-2048 using LSB for enhanced covert security.	Current research has not fully incorporated or assessed the Sri Lankan military environment.

1.2. Steganography concept

Steganography involves hiding secret information in seemingly harmless carrier files, such as images, audio, video, or text, to obscure the existence of concealed data. The primary goal of steganography is to keep the presence of confidential information unnoticed by unauthorised parties. In contrast to encryption, which modifies data to make it unreadable, steganography seeks to make the hidden data unseen or hard to detect. To comprehend steganography, it is important to grasp several key concepts. A carrier file, often called a cover file, serves as the medium that conceals the secret information. It can be an image, audio track, video, or any other kind of digital file.

Ideally, the carrier file should possess enough size to accommodate the concealed data without arousing suspicion. The payload or hidden data is the confidential information that is meant to be hidden in the carrier file. This payload can include plaintext messages, images, documents, or any other data that the sender wishes to keep private. The steganographic algorithm specifies the technique used to embed the hidden information in the carrier file [26]. Mutual methods of steganography consist of LSB insertion, spread spectrum, and phase encoding [26, 27]. LSB Insertion is a simple steganography method. It works by substituting the least significant bits of pixel values in the carrier file with bits from the hidden data. Since the least significant bits have a negligible impact on the overall visual or auditory quality, these changes are typically not noticeable to human observers.

1.3. Hybrid Cryptography concept

Hybrid encryption is a method that combines the advantages of symmetric and asymmetric encryption methods to enhance data security and address the shortcomings of both technologies [7, 28]. This approach allows for precise detection of any unauthorised modifications to the data. Symmetric encryption faces challenges related to key distribution; if the key is compromised, the entire system is compromised. To address this issue, asymmetric encryption is used to protect the AES symmetric key, facilitating secure key exchange between the sender and receiver.

Symmetric encryption is primarily used to protect data and ensure its confidentiality while it is in transit or stored. However, encryption alone cannot guarantee the integrity or authenticity of data. A malicious actor could alter the ciphertext without any notice. To address this issue, we implemented a Hash-based Message Authentication Code (HMAC) [29, 30]. It provides a way to verify that data has not been altered in transit. HMAC uses a cryptographic hash function, such as Secure Hash Algorithm (SHA)-256 or SHA-3, combined with a secret key known only to the sender and receiver [31, 32]. This approach allows for precise detection of any unauthorised modifications to the data. Symmetric encryption faces challenges related to key distribution; if the key is compromised, the entire system is compromised. To address this issue, asymmetric encryption is used to protect the AES symmetric key, facilitating secure key exchange between the sender and receiver.

Asymmetric encryption, often referred to as public-key cryptography, utilises a pair of keys that are mathematically linked: one public key and one private key. The public key is shared publicly and is used to encrypt data, while the private key is kept secret and is used to decrypt data. This form of encryption provides a secure way to exchange keys because the private key cannot be revealed to others [33]. A major disadvantage of asymmetric encryption is that it is slower and computationally more expensive than symmetric encryption, especially when dealing with large amounts of data [34]. Due to the inefficiency of asymmetric encryption, it is not feasible to rely solely on it to encrypt large amounts of data.

In hybrid cryptography, the sender creates a random symmetric encryption key to encrypt the plaintext. The sender then secures this symmetric key using the recipient's public key through asymmetric encryption. Both the encrypted symmetric key and the encrypted message are transmitted to the recipient. Upon receiving the encrypted data, the recipient utilises their private key to decrypt the symmetric key. Once the symmetric key is successfully revealed, the recipient can then proceed to decrypt the data. By combining both symmetric and asymmetric

encryption, this method leverages the speed of symmetric encryption for processing large volumes of data while taking advantage of the security provided by asymmetric encryption for the safe exchange of keys. This approach guarantees that even if a potential attacker captures the encrypted information and has access to the public key, they will still be unable to decode the symmetric encryption key, thereby preserving the confidentiality of the data.

2. Materials and methods

Data gathered from the experiments, such as execution durations, accuracy of extracted data, and attempts at security breaches, was examined to evaluate the system's performance, strength against security threats, and overall efficiency.

2.1. Proof of concept

The proof of concept was developed to demonstrate and evaluate the functional aspects of the proposed solution. It validated key operations, including AES encryption and decryption, RSA-based key management, and methods for LSB embedding and extraction. The solution was implemented using the NetBeans Integrated Development Environment (IDE), which was selected for its strong compatibility with Java application development. The choice to use NetBeans stemmed from its comprehensive toolset tailored for applications focused on cryptography and Graphical User Interface (GUI) development. NetBeans IDE was favoured for its robust support of Java-based cryptography, offering essential tools such as a GUI builder, debugging functionalities, and alignment with the Java Cryptography Architecture (JCA), which was crucial for the execution of AES and RSA algorithms. Even more than that, its ability to manage external libraries and Java packages seamlessly aids in the integration of cryptographic elements. The modular programming capabilities and cross-platform compatibility ensured the establishment of a reliable and scalable application environment. Together, these features facilitated efficient development, testing, and enhancement of the proof-of-concept system. Figure 1 illustrates a proof of concept for the design, featuring a single window that highlights key functions such as encryption and decryption.

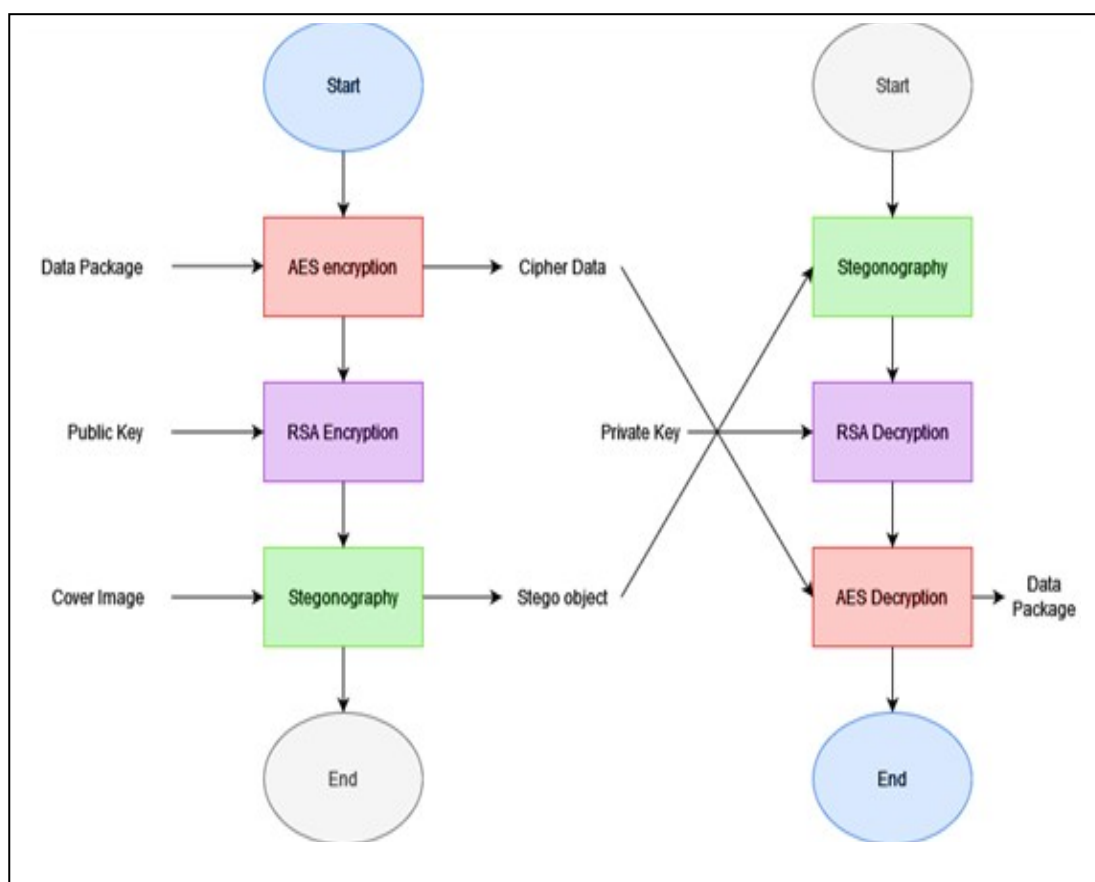


Figure 1: Proof of concept

Asymmetric key generation module creates a self-signed RSA key pair, consisting of both public and private keys, for the purposes of encryption and decryption. Although it's demonstrated in a single application, it mimics communication between systems that are located in different geographic areas. The public key we use to encrypt the symmetric key, while the private key is securely stored for decryption. In the AES/RSA encryption module, a random symmetric key (AES) is generated to effectively encrypt sensitive data. This newly created AES key is then protected by encrypting it using the recipient's public RSA key. The encrypted key is forwarded to the steganography encoding module, which provides an additional layer of security and guarantees confidentiality during the transfer of the key. When the stego file and encrypted data are received, the AES/RSA decryption module retrieves the concealed AES key with the private RSA key. It then decrypts the symmetric key and uses it to restore the original encrypted data, thus ensuring end-to-end confidentiality. The steganography encoding module embeds the encrypted AES key in a digital image using LSB steganography. It alters the least significant bits of specific image pixels to discreetly hide the key data in a way that is not noticeable to the human eye, creating a stego file for safe transmission. Finally, the steganography decoding module pulls the hidden bits from the stego image and reconstructs the encrypted AES key. This key is then sent to the decryption module for further steps, bridging the steganographic layer and the cryptographic decryption process.

The developed proof of concept has a single interface and demonstrates basic functionalities such as encryption and decryption. As well in addition, it allows users to select data files and cover images. User selections and commands are passed through the user interface to the application logic. The steganographic algorithm for embedding encrypted data into a carrier file is the LSB insertion method. The technique exploits the fact that changing the least significant bits of pixel values in a digital image has minimal perceptible impact while allowing secure data hiding. Figure 2 shows the primary interface of proof of concept.

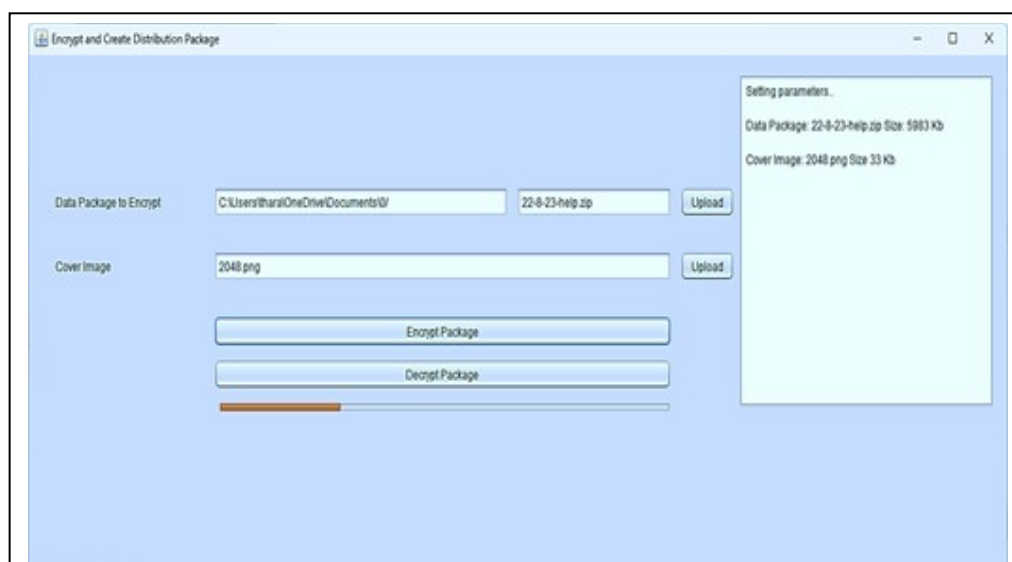


Figure 2: Primary interface of proof of concept

The main user interface of the proof of concept includes two upload buttons, one for the data pack and another for the cover image. By selecting the 'Encrypt Package' option, the encryption process will begin, leading to an encrypted data package with the image altered using steganography. On the other hand, clicking 'Decrypt Package' will start the decryption process, enabling access to the decrypted data package.

2.2. Proposed solution

A proof of concept is performed to demonstrate and evaluate the functional elements of the proposed solution. It confirms important operations such as AES encryption and decryption, RSA key management, and the process of embedding and extracting LSBs. The proposed solution was created using NetBeans IDE, which was chosen for its excellent support for Java application development. NetBeans provides key tools, including an easy-to-use GUI builder, integrated debugging options, and smooth compatibility with JCA, which is essential for implementing AES and RSA algorithms. Its modular coding features and support for multiple platforms facilitated the development of a reliable and adaptable application environment.

The encryption and decryption process uses a hybrid technology that combines AES-256 and RSA-2048. Data is encrypted using AES-256, which provides fast, secure encryption, ensures low latency, and efficient handling of large files. For the purpose of encrypting and transferring the AES key, RSA-2048 is implemented, facilitating a secure exchange of keys. In the decryption step, the AES key is extracted from the carrier image using the RSA private key, and the data is then decrypted using AES-256. The method is enhanced by parallel processing capabilities and can achieve fast decryption without the need for an initialisation vector. Figure 3 shows encryption and decryption process of the proposed solution.

```
private void embedFile(String inputFile, String outputFile)
throws InvalidKeyException, IllegalBlockSizeException, IOException, NoSuchAlgorithmExceptionExcept
NoSuchPaddingException, BadPaddingException, InvalidAlgorithmParameterException {

    System.out.println("Start Encryption: " + System.currentTimeMillis());

    KeyGenerator kgen = KeyGenerator.getInstance("AES");
    kgen.init(256);
    SecretKey key = kgen.generateKey();

    SecureRandom random = new SecureRandom();
    byte[] iv = new byte[16];
    random.nextBytes(iv);

    IvParameterSpec ivspec = new IvParameterSpec(iv);

    PublicKey pub = getPublicKey();

    FileInputStream fis = new FileInputStream(inputFile);
    Cipher cipher = Cipher.getInstance("RSA/ECB/PKCS1Padding");
    cipher.init(Cipher.ENCRYPT_MODE, pub);

    byte[] encKey = cipher.doFinal(key.getEncoded());
    byte[] encIV = cipher.doFinal(iv);

    File stegEncoded = new File("stage_key");
    Files.write(stegEncoded.toPath(), encKey);

    File stegEncodedIV = new File("stage_iv");
    Files.write(stegEncodedIV.toPath(), encIV);

    Steganography encoder = new Steganography();
    encoder.encode(inputFile, outputFile);
    encoder.encode("stage_key", "stage_key.png");
    encoder.encode("stage_iv", "stage_iv.png");

    System.out.println("Done Stage 1: " + System.currentTimeMillis());

    Cipher aesCipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
    aesCipher.init(Cipher.ENCRYPT_MODE, key, ivspec);

    try (FileOutputStream fos = new FileOutputStream(outputFile)) {
        byte[] buffer = new byte[1024];
        int numRead;
        while ((numRead = fis.read(buffer)) >= 0) {
            byte[] output = aesCipher.update(buffer, 0, numRead);
            if (output != null)
                fos.write(output);
        }
        byte[] finalBytes = aesCipher.doFinal();
        if (finalBytes != null)
            fos.write(finalBytes);
    }

    fis.close();
    System.out.println("End Stage 2: " + System.currentTimeMillis());
}

private static void decodeFile(String inputFile, String outputFile)
throws InvalidKeyException, IllegalBlockSizeException, IOException, NoSuchAlgorithmExceptionExcept
NoSuchPaddingException, BadPaddingException, InvalidAlgorithmParameterException {

    PrivateKey prv = getPrivateKey();

    File stegKey = new File("stage_key.png");
    Cipher cipher = Cipher.getInstance("RSA/ECB/PKCS1Padding");
    cipher.init(Cipher.DECRYPT_MODE, prv);

    // Decode stepo-encoded key and IV
    Steganography decoder = new Steganography();
    stegDecoder.decode("stage_key.png", "stage_key");
    stegDecoder.decode("stage_iv.png", "stage_iv");

    byte[] encKey = Files.readAllBytes(new File("stage_key").toPath());
    byte[] encIV = Files.readAllBytes(new File("stage_iv").toPath());

    SecretKeySpec key = new SecretKeySpec(cipher.doFinal(encKey), "AES");
    IvParameterSpec ivspec = new IvParameterSpec(cipher.doFinal(encIV));

    Cipher aesCipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
    aesCipher.init(Cipher.DECRYPT_MODE, key, ivspec);

    try (FileOutputStream fos = new FileOutputStream(outputFile);
        FileInputStream fis = new FileInputStream(inputFile)) {
        byte[] buffer = new byte[1024];
        int numRead;
        while ((numRead = fis.read(buffer)) >= 0) {
            byte[] output = aesCipher.update(buffer, 0, numRead);
            if (output != null)
                fos.write(output);
        }
        byte[] finalBytes = aesCipher.doFinal();
        if (finalBytes != null)
            fos.write(finalBytes);
    }

    System.out.println("Decryption Completed Successfully.");
}
```

Figure 3: Encryption and decryption process

In the process of steganography, the encrypted AES key is incorporated into a carrier image using the LSB steganography method. Images are selected as carrier files because they are capable of concealing large amounts of data while maintaining a natural visual appearance. The LSB technique alters the least significant bits of the pixel values in the image to embed the encrypted AES key, with the changes being imperceptible. The modified pixel matrix creates a stegosaurus that is visually similar to the original image but hides the encrypted data. In the decryption phase, the AES key is retrieved from the image and used to decrypt the encrypted information.

The process of encryption begins with the creation of a symmetric encryption key that is used to encrypt plaintext data utilising AES. This coded information is then securely transmitted through RSA, using the recipient's public key to encrypt the AES key. The encoded symmetric key is included in the stego image that conceals the information. Upon receiving the stego image, the recipient extracts their private key to decode the AES key, followed by employing this key to decrypt the data, thereby retrieving the original plaintext. Once the recipient

acquires the message that is both encoded and concealed via steganography, they use LSB decoding to uncover the hidden information, decrypt the AES key using their RSA private key, and then access the original message through the obtained AES key. This method guarantees that only the designated recipient can read the message, thereby ensuring complete confidentiality throughout the entire process. Furthermore, it ensures that the message stays concealed and secure during the entire transmission, fortified by hybrid encryption and embedded in the image using steganography. By blending AES and RSA encryption with LSB steganography, this method offers a dual-layer of protection, making the concealed message extremely hard to detect or compromise, making it particularly effective for situations facing advanced persistent threats and challenges related to quantum attacks.

The system underwent a range of testing methods to evaluate its functionality and performance. Functional testing confirmed the main operations of embedding and extracting encrypted data in the carrier image. Data volume analysis examines the quantity of encrypted information that can be concealed in a carrier image without causing perceptible alterations. Performance evaluation assessed the system's efficiency, concentrating on the speeds of encryption and decryption with different data loads. In a military context where immediate or nearly immediate responses are needed, the speed of the encryption and decryption process is essential. Consequently, it is vital to perform a thorough assessment and reference research-backed benchmarks to measure the time required for encryption to be completed. Security evaluations concentrated on the system's ability to thwart unauthorised access and steganalysis attempts, thereby preserving the confidentiality and integrity of the concealed data. The security testing tools utilised include the Virtual Steganalysis Library, StegExpose, and Aperisolve, which are employed to evaluate their effectiveness against Steganalysis detection.

3. Results

By implementing a proof of concept, the system was created and evaluated, and valuable insights were gained into its performance, security strength, and operational efficiency. The assessment of a hybrid encryption system that incorporates AES, RSA, and LSB steganography provides significant findings on its effectiveness. Through a proof of concept, the system was built and reviewed, gaining key insights into its performance, security robustness, and functional efficiency. The evaluation of a hybrid encryption system combining AES, RSA, and LSB steganography yielded significant results regarding its effectiveness. Time metrics were recorded for the stages of encryption, key exchange, and steganography encoding across different file and image sizes. Table 2 shows Performance Metrics for AES, RSA , and LSB Steganography Encoding Across Different File and Image Sizes.

Table 2: Performance Metrics for AES, RSA, and LSB Steganography Encoding Across Different File and Image Sizes

Data File Size	Cover Image Size	AES Encryption	RSA Encryption	STEGO Encoding
6,127,143 bytes	1,911 bytes (225×225, 8-bit)	72 ms	31 ms	3,199 ms
6,127,143 bytes	1,964 bytes (320×320, 32-bit)	79 ms	31 ms	8,742 ms
6,127,143 bytes	33,871 bytes (664×804, 24-bit)	2,824 ms	32 ms	51,557 ms
364,335,224 bytes	1,964 bytes (320×320, 32-bit)	72 ms	63 ms	8,707 ms

3.1. Observations

The results of the study showed that as file and image sizes increased, the time required for processing increased significantly. For example, encrypting a 6MB file using a 225×225 image takes 3,299 milliseconds, while using a 664×804 image the total time increases to over 51,000 milliseconds. During the steganalytic embedding process, since the cover image is larger, more input/output operations are required, which leads to a significant increase in processing time. AES encryption has proven to be fast and reliable even for larger files. RSA encryption times are relatively short and consistent because it only encrypts the AES key and not the data itself. results of the study showed that as file and image sizes increased, the time required for processing increased significantly. For example, encrypting a 6MB file using a 225×225 image takes 3,299 milliseconds, while using a 664×804 image the total time increases to over 51,000 milliseconds. During the steganalytic embedding process, since the cover image is larger, more input/output operations are required, which leads to a significant increase in processing time. AES encryption has proven to be fast and reliable even for larger files. RSA encryption times are relatively short and consistent because it only encrypts the AES key and not the data itself.

3.2. Performance Analysis

The results show that image size has a greater impact on performance than overall file size. The main drawback lies in LSB steganography, which is hampered by the slow process of modifying the least significant bit in each pixel. AES encryption demonstrated excellent scalability and very little delay, even when handling large data files. The impact of RSA on processing time became more apparent with extremely large datasets, though it remained within manageable limits. The decryption process was notably quicker, requiring less than a millisecond. This speed is due to the absence of random initialisation vector generation during decryption, along with the ability to parallelise AES decryption. Security assessments using steganalysis tools such as Virtual Steganalysis Library, StegExpose, and Aperisolve verified that the system successfully hides the encryption keys to prevent detection.

4. Discussion

The hybrid encryption method successfully encrypts the data and integrates it securely into the image. AES provides fast encryption, while RSA facilitates secure key exchange. However, the LSB steganography technique exhibits considerable delays when processing larger images, which suggests that optimisation is required when selecting the carrier image in practical applications. Choosing smaller but appropriately sized images can improve performance without compromising security. Although increases in file size did impact times, it was not as severely as the size of the images. The created model has enhancements in terms of safety strength and acceptable performance levels compared to current hybrid systems. The use of NetBeans IDE simplifies the development and debugging process and enables precise testing and modular programming. AES continues to be a highly effective encryption technique, while RSA offers strong safeguards for key exchanges. While LSB steganography may not be the fastest option, its immediacy and effectiveness in covert communications make it suitable for military use.

This study introduced, developed, and assessed a hybrid cryptographic system that combines AES-256, RSA-2048, and LSB steganography to enhance communication security for the military sector and the Ministry of Defence. The system's design was structured in a modular way, incorporating components for key generation, encryption, decryption, and steganography encoding/decoding to ensure layered protection. It was developed using the NetBeans IDE as well as JCA libraries, which guaranteed a dependable and secure execution of cryptographic functions. A modular approach was adopted to allow straightforward integration, debugging, and possible future upgrades of the system's capabilities. Experimental results demonstrated that AES provided swift and efficient encryption for large files, while RSA was efficacious in managing key exchanges with minimal performance impact. LSB steganography succeeded in embedding encrypted data into images without greatly compromising image quality, making covert communication feasible. However, the time taken for encoding increased significantly with larger images, making steganography the main performance bottleneck due to a rise in input/output operations. The analysis revealed that image size had a more pronounced effect on performance than the size of the file. AES displayed remarkable scalability and low latency, whereas RSA's impact remained managed even with large datasets. Decryption was notably quick, taking under one millisecond due to the absence of IV generation and the ability to parallelised AES. Security evaluations conducted with tools like the Virtual Steganographic Library, StegExpose, and Aperisolve confirmed that the system successfully concealed encrypted keys from detection. In conclusion, the suggested hybrid model improves security while sustaining acceptable performance, making it well-suited for secure and discreet communications in sensitive environments. When compared to existing solutions, the proposed system demonstrated advancements in both security robustness and operational efficiency. Even so, the LSB steganography technique introduced processing delays when dealing with larger images. Future research should examine steganography methods that employ Discrete Cosine Transform or Discrete Wavelet Transform to enhance the speed of data embedding while maintaining security integrity. Even more than that, the implementation of Elliptic Curve Cryptography could further reduce the key management burden. In short, the hybrid cryptographic system developed in this study presents an efficient, secure, and practical solution for safeguarding sensitive military communications. With minor improvements, it holds considerable promise for real-world implementation in national security frameworks.

Acknowledgements

The authors wish to convey their heartfelt appreciation to ESOF Uni, Sri Lanka, for offering the platform and resources essential for the successful execution of this research. Special thanks are also extended to Kingston University, United Kingdom, for their valuable academic partnership and ongoing support in facilitating this collaborative research effort.

Citations and References

- [1] The Economic Times, "Sri Lankan Police social media accounts, government website hacked," ed, 2025.
- [2] The Cyber Express, "Sri Lanka Ministry, Defence Face Data Breach.," in *The Cyber Express*, ed: Cyble Inc, 2022.
- [3] Defence.lk. "Aegis Lexicon 2024 – Defence Cyber Exposition." Defence.lk, 2024. https://www.defence.lk/Article/view_article/28104. (accessed.
- [4] CERT.lk. "Steps Taken to Establish the National Cyber Security Operations Center in Sri Lanka," Cyber Security Newsletter." <https://www.cert.gov.lk/> (accessed.
- [5] I. Haverkamp, "Analyzing Advantages and Limitations of Combining Cryptography and Steganography Across Applications: A Systematic Review," 2023, doi: <https://doi.org/10.21203/rs.3.rs-3167378/v1>.
- [6] N. H. Anh, "Hybrid Cloud Migration Strategies: Balancing Flexibility, Security, and Cost in a Multi-Cloud Environment," *Transactions on Machine Learning, Artificial Intelligence, and Advanced Intelligent Systems*, vol. 14, no. 10, pp. 14–26, 2024, doi: <https://doi.org/10.36948/ijfmr.2021.v03i02.39459>.
- [7] Q. Zhang, "An overview and analysis of hybrid encryption: the combination of symmetric encryption and asymmetric encryption," in *2021 2nd international conference on computing and data science (CDS)*, 2021: IEEE, pp. 616–622.
- [8] R. Bishukarma, "Privacy-preserving based encryption techniques for securing data in cloud computing environments," *Int. J. Sci. Res. Arch*, vol. 9, no. 2, pp. 1014–1025, 2023.
- [9] R. Banoth and R. Regar, "Asymmetric Key Cryptography," in *Classical and Modern Cryptography for Beginners*: Springer, 2023, pp. 109–165.
- [10] A. Parekh, M. Antani, K. Suvarna, R. Mangrulkar, and M. Narvekar, "Multilayer symmetric and asymmetric technique for audiovisual cryptography," *Multimedia Tools and Applications*, vol. 83, no. 11, pp. 31465–31503, 2024.
- [11] P. Kuppuswamy, S. Q. Y. A. K. Al, R. John, M. Haseebuddin, and A. A. S. Meeran, "A hybrid encryption system for communication and financial transactions using RSA and a novel symmetric key algorithm," *Bulletin of Electrical engineering and Informatics*, vol. 12, no. 2, pp. 1148–1158, 2023.
- [12] K. F. Rafat and S. M. Sajjad, "Advancing reversible LSB steganography: Addressing imperfections and embracing pioneering techniques for enhanced security," *IEEE Access*, 2024.
- [13] R. Montasari, *Countering cyberterrorism: The confluence of artificial intelligence, cyber forensics and digital policing in US and UK national cybersecurity*. Springer, 2023.
- [14] L. Chen, "Standardisation of and Migration to Post-Quantum Cryptography," in *International Conference on Research in Security Standardisation*, 2024: Springer, pp. 3–13.

- [15] M. Dalal and M. Juneja, "Steganography and Steganalysis (in digital forensics): a Cybersecurity guide," *Multimedia Tools and Applications*, vol. 80, no. 4, pp. 5723–5771, 2021.
- [16] A. Kumar, S. Tandon, S. Deorari, and R. Kumar, "Steganography: Unveiling Techniques and Research Agenda," 2024.
- [17] A. Nicolás-Sánchez and F. J. Castro-Toledo, "Uncovering the social impact of digital steganalysis tools applied to cybercrime investigations: a European Union perspective," *Crime Science*, vol. 13, no. 1, p. 11, 2024.
- [18] I. K. Azzani, S. A. Purwantoro, and H. Z. Almubarak, "Enhancing awareness of cyber crime: A crucial element in confronting the challenges of hybrid warfare in Indonesia," *Defense and Security Studies*, vol. 5, no. 1, pp. 1–9, 2024.
- [19] B. Haldar, P. K. Mukherjee, and H. N. Saha, "A Robust Security Strategy Using Hybrid Cryptography Approach to Protect Data in the Financial Sector," *International Journal of Computer Networks and Applications (IJCNA)*. <https://www.ijcna.org/abstract.php>, 2024.
- [20] Z. A. Mohammed, H. Q. Ghani, Z. J. Hussein, and A. K. M. Al-Qurabat, "Advancing cloud image security via AES algorithm enhancement techniques," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12694–12701, 2024.
- [21] W. Steingartner, D. Galinec, and A. Kozina, "Threat defense: Cyber deception approach and education for resilience in hybrid threats model," *Symmetry*, vol. 13, no. 4, p. 597, 2021.
- [22] S. Ediriweera, B. Dilhara, and C. Disanayake, "Web-Based Data Hiding: A Hybrid Approach Using Steganography and Visual Cryptography," in *2023 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, 2023, vol. 6: IEEE, pp. 1–7.
- [23] S. Pemasinghe, "WaveCryptor: A Secure Hybrid Framework Combining Encryption and Steganography via Frequency Mapping," in *2025 5th International Conference on Advanced Research in Computing (ICARC)*, 19–20 Feb. 2025 2025, pp. 1–6, doi: 10.1109/ICARC64760.2025.10962964.
- [24] C. Ruangsuwan and Y. Surapanpairoge, "The Cybersecurity, Cyber Threats, Cyberterrorism, and Cybersecurity Framework for Critical Information Infrastructure for the Ministry of Defence of Thailand," *National Defence Studies Institute Journal*, vol. 14, no. 2, pp. 54–64, 2023.
- [25] K. Michaylov, "Exploring the Use of Steganography and Steganalysis in Forensic Investigations for Analysing Digital Evidence," University of Twente, 2023.
- [26] M. Shyla, K. S. Kumar, and R. K. Das, "Image steganography using genetic algorithm for cover image selection and embedding," *Soft computing letters*, vol. 3, p. 100021, 2021.
- [27] S. Rahman, J. Uddin, H. U. Khan, H. Hussain, A. A. Khan, and M. Zakarya, "A novel steganography technique for digital images using the least significant bit substitution method," *IEEE Access*, vol. 10, pp. 124053–124075, 2022.
- [28] H. Abroshan, "A hybrid encryption solution to improve cloud computing security using symmetric and asymmetric cryptography algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, 2021.
- [29] T. Lawrence, F. Li, I. Ali, M. Y. Kpiebaareh, C. R. Haruna, and T. Christopher, "An HMAC-based authentication scheme for network coding with support for error correction and rogue node identification," *Journal of Systems Architecture*, vol. 116, p. 102051, 2021.

- [30] T. Lawrence, F. Li, I. Ali, C. R. Haruna, M. Y. Kpiebaareh, and T. Christopher, "A computationally efficient HMAC-based authentication scheme for network coding," *Telecommunication Systems*, pp. 1–23, 2022.
- [31] C. S. Ranganathan and C. Srinivasan, "Enhanced Cloud Data Security by Employing HMAC for Advanced Cryptographic Protection," *Innovations in Intelligent Systems and Advanced Engineering*, vol. 1, no. 1, pp. 1–10, 2025.
- [32] D. Upadhyay, N. Gaikwad, M. Zaman, and S. Sampalli, "Investigating the avalanche effect of various cryptographically secure hash functions and hash-based applications," *IEEE Access*, vol. 10, pp. 112472–112486, 2022.
- [33] D. R. Natarajan, S. Peddi, D. T. Valivarthi, S. Narla, S. S. Kethu, and D. Kurniadi, "Secure user authentication and data sharing for mobile cloud computing using BLAKE2 and Diffie-Hellman key exchange," in *2024 International Conference on Emerging Research in Computational Science (ICERCS)*, 2024: IEEE, pp. 1–8.
- [34] N. Kshetri, M. M. Rahman, M. M. Rana, O. F. Osama, and J. Hutson, "algoTRIC: Symmetric and asymmetric encryption algorithms for Cryptography--A comparative analysis in AI era," *arXiv preprint arXiv:2412.15237*, 2024.