Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Evaluation of the Least Significant Bit-Based Steganography's Performance Using Elliptic Curve Cryptography

Kafayat Odunayo Tajudeen[1], Akeem Femi Kadri[2], Oladele Taye Aro[3]

[1,3]Department of Computer Science, Al-Hikmah University, Ilorin, Nigeria.

[2]Department of Computer Science, Kwara State University, Malete, Nigeria.

kotajudeen@alhikmah.edu.ng[1], akeem.kadri@kwasu.edu.ng[2], taiwo774@gmail.com[3]

Abstract

The importance of data security is growing for companies trying to safeguard sensitive information. Cryptography and steganography are two widely used methods. Steganography enables the concealment of secret information within digital content to reduce suspicions. Even though the Least Significant Bit (LSB) technique is widely utilized, when applied alone, it is still vulnerable to security breaches. Traditional approaches that combine LSB steganography with RSA encryption enhance security but are computationally expensive, making them less useful for real-time applications like data storage, secure communication, and the Internet of Things. The need for a robust yet lightweight system that combines encryption and steganography without compromising performance is thus necessary. This study proposes a hybrid approach that blends LSB steganography with Elliptic Curve Cryptography (ECC). Because it uses less processing power and has lower key sizes than RSA, ECC was selected to maintain similar security standards. The method was implemented and evaluated in terms of encryption time, data-hiding capabilities, and efficiency. Experimental results show that the LSB–ECC framework outperforms the conventional LSB–RSA combination. Specifically, by significantly reducing encryption time and improving hiding efficiency, ECC offers a more scalable solution for safeguarding sensitive data. The findings confirm that ECC provides a lighter solution without sacrificing security. ECC's integration with LSB steganography exhibits exceptional efficiency and security performance, making it appropriate for contemporary data protection requirements. Adaptive steganographic approaches and testing on extensive multimedia datasets will be incorporated into future work to further validate robustness against sophisticated steganalysis and cryptographic attacks.

Keywords: Cryptography, Data Security, Elliptic Curve Cryptography (ECC), Information Hiding, Least Significant Bit (LSB) Steganography, Steganography

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

1. Introduction

The widespread and frequent use of the internet nowadays has resulted in a rise in data security risks because of the massive data transfers over the network (Basholli et al., 2024). The attackers are attempting to disrupt the transfer procedure to steal crucial data (Mallick & Nath, 2024). As a result, security has emerged as one of our top priorities (Movahed, 2024). Because their data is so valuable, banks and government buildings are the most crucial places to implement security (Wang et al., 2024). Steganography and cryptography are the most popular techniques for applying security and guaranteeing data transmission (Rashid et al., 2024). The process of concealing data is called steganography (Rakhra et al., 2021). There are several different kinds of steganography, such as text, audio, video, and image (Kunhoth, 2024). One of the most modern steganography techniques that hides data inside an image is called LSB image steganography (Alanzy et al., 2023). The primary disadvantage of this LSB approach is that it does not have the security required to protect data (Rinki et al., 2022). The cryptography method can be used to provide the highest level of data security (Mua'ad et al., 2022). One aspect of cryptography is the conversion of plaintext into an incomprehensible format (Biryukov, 2025). The two types of cryptography are symmetric and asymmetric; symmetric cryptography is less secure than asymmetric encryption (Kapoor & Thakur, 2022). Symmetric and asymmetric encryption techniques include AES, RSA, ECC, Blowfish, DES, and others (Abroshan, 2021). The computational cost of RSA asymmetric encryption is a challenge, despite its well-known strong security (Chen & Ye, 2022). Sharma and Thapa's (2019) multilayer security technique create a hybrid system combining steganography and cryptography. The technique uses RSA cryptography and LSB steganography to boost security. However, it is slower than other encryption techniques and ineffective for long-term data storage. Some scholars have created steganography to circumvent the problem with RSA and LSB algorithms for effective text data security using LSB pictures and other encryption approaches. Furthermore, Sani and Nujjaid (2025) recommend combining steganography and encryption using the Diffie-Hellman key exchange and the Least Significant Bit (LSB) technique. The Diffie-Hellman key exchange outperforms RSA in terms of data-hiding capabilities and operates more quickly. Wandera et al. (2024) write, techniques like square and multiply, divide and conquer, and repetition are devised to overcome the performance constraint of RSA. Iteration is most suitable for short keys, according to experiments, and square and multiply provide maximum efficiency for medium and long keys. The appropriate auxiliary algorithms aid in maximizing the performance of RSA without reducing security. Wahab et al. (2021) aim to reduce data transmission time and storage space while using the Internet without losing encryption and security. A hybrid compression method of integrating RSA cryptography and Huffman coding is proposed. The process compresses plaintext, wraps images, and embeds encrypted information and reduces dimensions through Discrete Wavelet Transform (DWT). The system works best

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

relative to other methods. ECC offers encryption mechanisms that are highly secure and efficient (Shaaban et al., 2024). The aim of this research is to develop a system to hide data and information in images in files with the aim of encrypting data so data and information offered through communication media are as effective and secure as possible.

To provide efficient and secure protection and security of internet text files, this study offers an efficient system that applies LSB steganography in pictures and ECC information encryption. Figure 1 illustrates the six main steganography groups, and Figure 2 shows encryption algorithms.

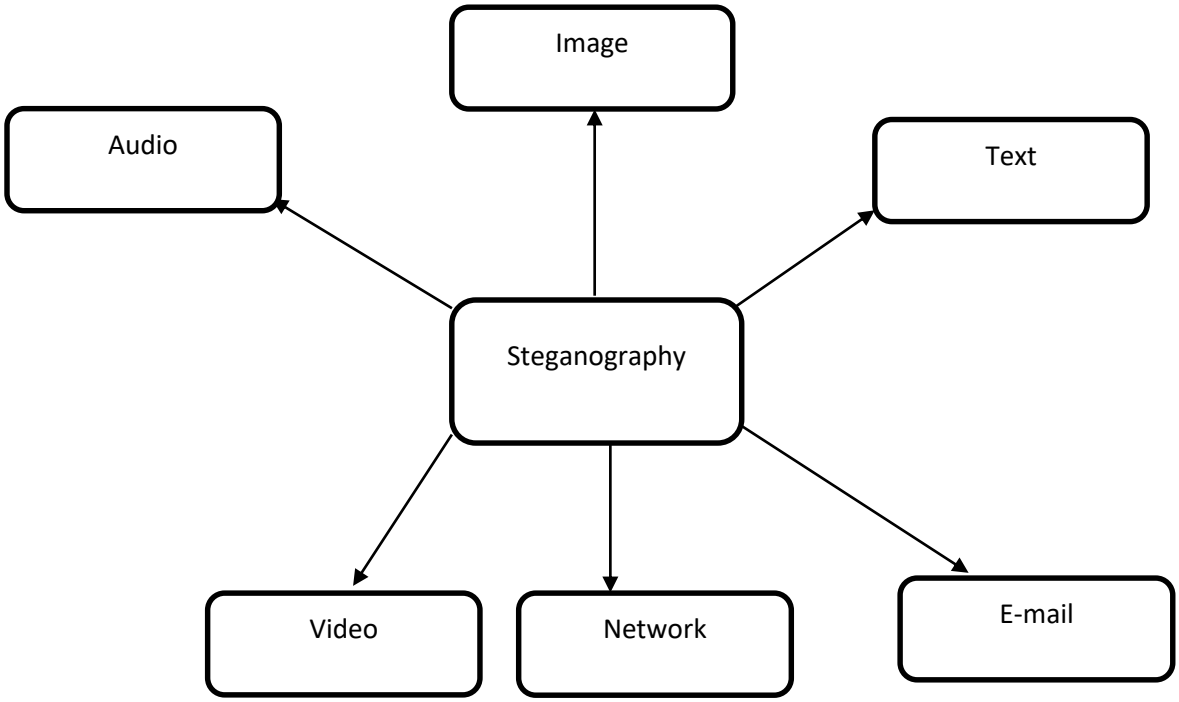

Figure 1. Classification of Steganography (Majeed et al., 2021)

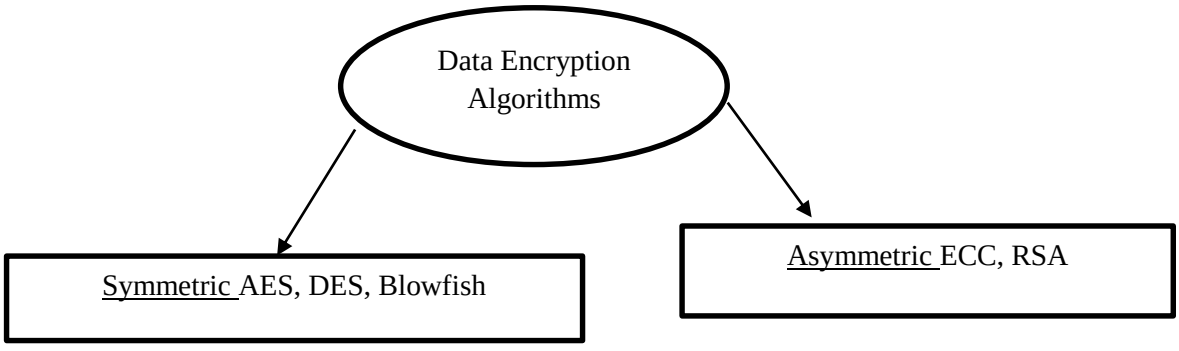

Figure 2. Types of Data Encryption Algorithms (Matta et al., 2021)

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

2. Related Works

Sani et al. (2025) combined steganography and encryption by applying the Diffie-Hellman key exchange and the Least Significant Bit (LSB) approach. The study demonstrates that the Diffie-Hellman key exchange is more efficient and faster than RSA in data concealment. Steganography, which is an art of covering secret information, is employed in online transaction systems, the Internet of Things, and warfare. New developments in steganalysis and image processing have made the application of techniques for revealing hidden information by adversaries more viable. Sharma and Thapa (2019) suggest a multilayer security approach that builds a hybrid framework with the help of the integration of steganography and cryptography. The approach integrates LSB steganography and RSA cryptography to achieve better security. However, it is slower compared to some other encryption techniques and not efficient for data storage in the long run. Data security is among the grave challenges of the era of electronic communication. While steganography can be broken if confidential data is exposed, traditional cryptography techniques are easily abused by adversaries. Kyei et al. (2019) integrate the most effective methods to build a strong system that is impervious to attacks. Cryptography transforms messages into unintelligible file formats, whereas steganography conceals messages within cover objects. Asymmetric cryptography uses RSA, while messages are inserted into cover objects using the least significant bit (LSB) technique. Data security and confidentiality on Android smart devices are guaranteed by this combination. Even though modern devices like Android and BlackBerry have better communication capabilities, security issues still exist because of insufficient safeguards. Many techniques have been tried, including as steganography and cryptography. Bhargava and Mukhija (2019) discuss how to secure photos using encryption techniques including LSB bits, DWT, and the RSA algorithm. It also touches upon new methods that combine cryptography and steganography with image processing to encrypt data and hide insights in other media. As the method is programmed in MATLAB and provisions for PSNR computation, MSE, and entropy of stego and cover images are included, the method is secure for the transmission of digital data. Apau and Adomako (2017) propose an image cover object as a technique of hiding RSA-encrypted messages that are encoded using the Least Significant Bit (LSB) method. Peak signal-to-noise ratio (PSNR) performance analysis shows that cryptography and steganography combined offer strong security and robustness in smartphones. Security concerns have been raised through increased usage of smartphones since they do not have security features. Steganography and cryptography are two novel ways of addressing the above issues. Ullah et al. (2024) propose a steganography approach of audio encryption using LSB-based computation in order to assure maximum cloud protection. The technique includes image steganography, Multi-Level Encryption Algorithm (MLEA), and Two-Level Encryption Algorithm (TLEA) in encrypting the information,

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

where the performance is 1.732125% better on average than any other alternative. Wandira et al. (2024) mention that the RSA technique, designed by MIT in 1977, is the most popular public-key encryption scheme. Due to computations involved, this block cipher operates slower for longer key sizes. Methods such as divide and conquer, square and multiply, and repetition have been developed to overcome RSA's performance issues. Experiments indicate that iteration is optimal for short keys, while square and multiply is most efficient for medium to long keys. The use of appropriate support algorithms can improve RSA's performance without compromising security. Tajudeen et al. (2024) this paper introduced encryption by using RNS to improve the security of transmitted data. Traditional moduli set set $\{m1 = 2\,n - 1,\, m2 = 2\,n\; and\; m3 = 2\,n + 1\}$ and Chinese Remainder Theorem (CRT) were used to encrypt and decrypt the data based on information exchange. Furthermore, this is done by representing each character in the plaintext data is assigned a unique number value and which is done by converting each character into its ASCII value and the ASCII value is then converted to Excess-3 value. Thus, anytime an attacker tries to access the data, the RNS algorithm will generate residues data, this helps to discourage an attacker from further threat. The result showed that the proposed system was able to resist brute-force attacks compared to others systems. Values of n for the moduli set were dynamic against encryption and decryption of data. Hamidu et al. (2025) recommend using Huffman encoding for modified RSA-AES encrypted token compression in secure banking transactions to increase security because of the SHA-256 token generation's historical strength and resistance to brute-force attacks. Wahab et al. (2021) assert that security and encryption should be preserved while reducing the amount of time and storage space required for data transmission across the Internet. For data compression, a hybrid strategy combining RSA cryptography and Huffman coding is recommended. Using the Discrete Wavelet Transform (DWT), the method reduces the size of regular text and graphics while adding encrypted data. Unlike previous methods, the system functions well. Data compression is crucial to information security since it keeps data safe and usable. Different types of data are compressed using lossy and lossless approaches. After several previous approaches using compression techniques are proposed, ECC encryption algorithms are eventually applied to speed up the encryption time.

3. Methodology

One of the advantages of using the ECC encryption algorithm is that it is very secure and fast in the encryption process. The study also aims to simulate a robust and secure text file encryption system that combines LSB image steganography with ECC encryption algorithms. By fusing steganography with encryption, the system ensures enhanced security for sensitive information in a text file, making it

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

difficult for unauthorized parties to detect and access the StegoCryptText information. This developed method has two phases: the hiding and encryption and decryption and extraction phase. The hiding and encryption phase exists in the sender part, in which this phase includes both the steganography method and the encryption method. Figures 3 and 4 display the flowchart of the developed StegoCrypt model. The StegoCryptText is received at the receiver's side; this text is unhidden and decrypted back to the original text file at the receiver's end.

3.1 Dataset Acquisition

The dataset used in the study consisted of digital text files and cover photos. To guarantee testing's reproducibility and equity, we used open-source repositories for cover images and free image databases. To replicate confidential communications, the secret data (StegoText) was built as text strings of varying lengths. These datasets were chosen because they are widely used in steganographic research and because they offer enough variation in image resolution, size, and format.

3.2 Data Preprocessing

To avoid distortion during embedding and extraction, all images were transformed into lossless formats (PNG) and standardized to a fixed resolution (e.g., 256×256 pixels) prior to embedding. ASCII encoding was used to convert text data into a binary format. The Least Significant Bit (LSB) replacement technique was then used to get this binary stream ready for embedding into the cover images.

3.3 System Implementation

Using Python 3.11, OpenCV for image processing, and readily available ECC encryption tools, the suggested solution was put into practice as a desktop program. The workflow proceeded as follows:

1. Use ECC to encrypt confidential text.
2. Create a binary stream out of the ciphertext.
3. Use LSB substitution to embed the binary stream into the cover image.
4. To ensure accuracy, extract and decrypt the concealed message.

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

3.4 Evaluation Metrics:

The performance of the developed system was evaluated using execution time metrics.

3.5 Experimental Environment

A desktop computer with an Intel Core i7 processor, 16 GB of RAM, and a Python-based programming environment running Windows 11 was used for all experiments.

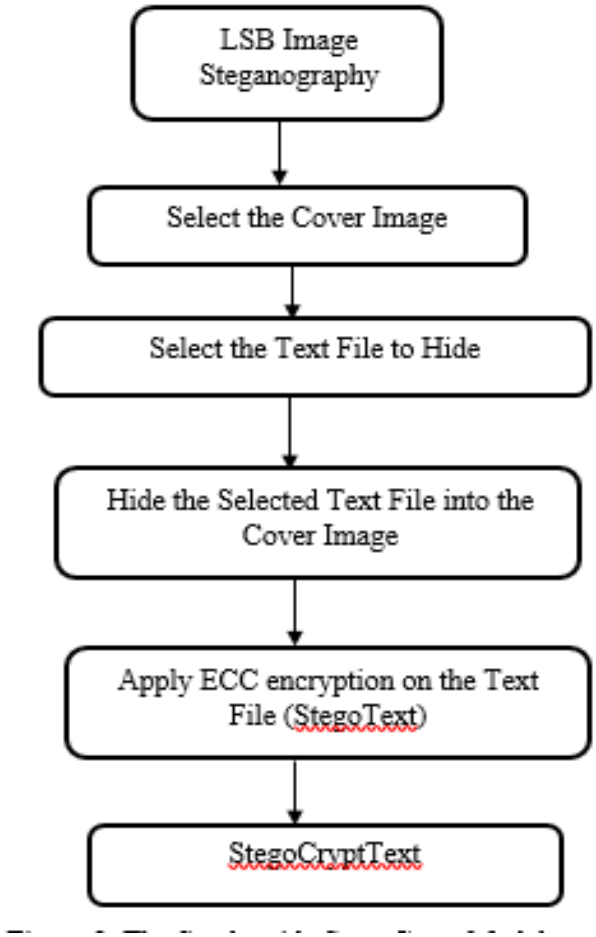

Figure 3. The Sender side StegoCrypt Model

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

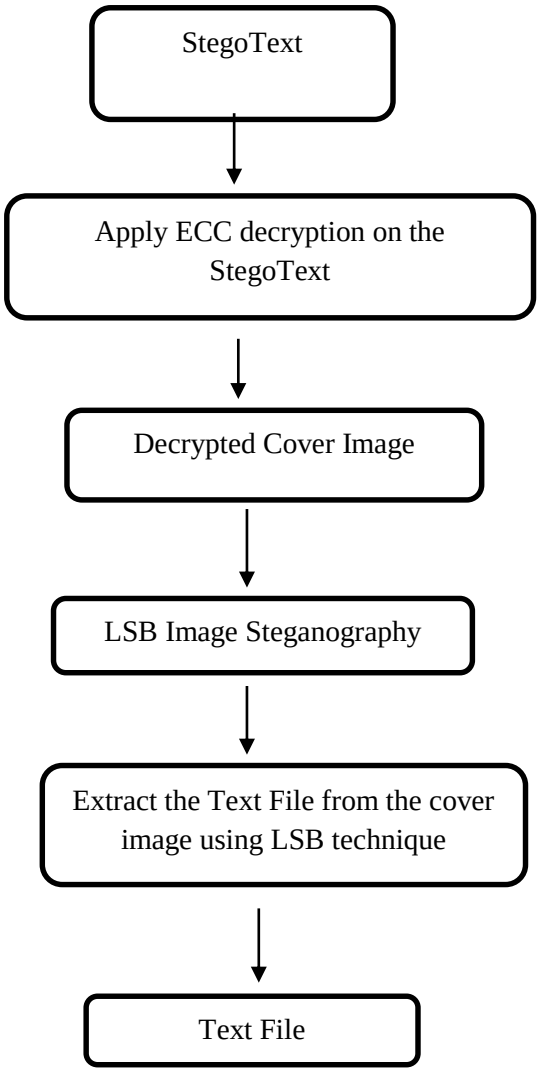

Figure 4. The Receiver Side StegoCrypt Model

Table1. Pseudocode for Hiding and Encrypting of Text File

Algorithm 1: Hiding and Encrypting of Text File

Sender Side (Embed → Encrypt)

Input: Text file/message: M, Cover image: I, ECC public key: K_{pub}

Begin

FUNCTION SenderProcedure (secret.txt, cover.png, K_pub):

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

```
// Step 1: Read message
    M ← ReadFile(secret.txt)
    // Step 2: Convert message to binary bitstream
    B ← ConvertToBytes(M)
    S ← ConvertToBitstream(B) // Binary string of all bits
// Step 3: Load cover image
    I ← LoadImage(cover.png)
    pixel_list ← GetPixels(I)
// Step 4: Embed bits using LSB
    FOR i from 0 to length(S) - 1 DO
        pixel ← pixel_list[i / 3]
        channel ← i MOD 3 // 0 = R, 1 = G, 2 = B
        pixel[channel] ← ReplaceLSB(pixel[channel], S[i])
    END FOR
I' ← SaveImage(pixel_list)
// Step 5: Encrypt stego image using ECC
    image_bytes ← ReadFile(I') // Read binary data of modified image
    C ← ECC_Encrypt(image_bytes, K_pub)
 // Step 6: Save encrypted stego image
    WriteFile("encrypted_stego.dat", C)
 End
```

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Table 2. Pseudocode for Decrypting and Extraction of StegoCryptText File

Algorithm 2: Decrypting and Extraction of StegoCryptText File

```
Receiver Side (Decrypt → Extract)
Input: Encrypted stego image: C, ECC private key: K_priv
Output: M′      ← Decrypted message
Begin
// Step 1: Decrypt the stego image using ECC
  I′ ← ECC_Decrypt (C, Kpriv)

// Step 2: Extract hidden bits from the least significant bits (LSBs) of the image
  S′ ← Extract_LSBs(I′)

 // Step 3: Convert the extracted bit stream into bytes
  B′ ← Bits_To_Bytes(S′)
// Step 4: Decode the byte sequence into the original message
  M′ ← Decode(B′)
Return M′
End
```

4. Results and discussion

The simulation of the developed system and the result are presented in this section. In implementing the StegoCrypt model, all the algorithms presented in Section 3 are converted to Python codes, and the system is evaluated using time performance. The Interface as shown in Figure 5 provides a graphic user interface (GUI) for the program interface. The Select Image function gives users the ability to choose a local image file to use as the carrier for hiding their private text file as shown in Figure 5. The desktop application also provides a GUI for the selection and encryption of text files as displayed below.

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

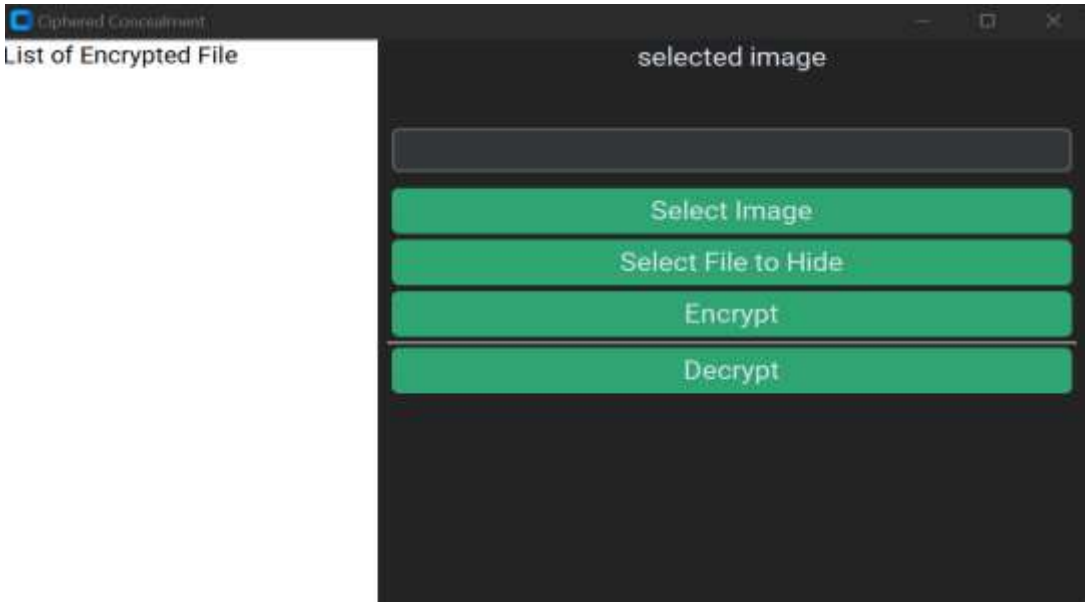

Figure 5. Program Interface of the Developed System.

Figure 6 allows users to initiate the steganographic process by selecting an image from the file directory, supported by various image formats like JPEG/JPG, PNG, BMP, and GIF.

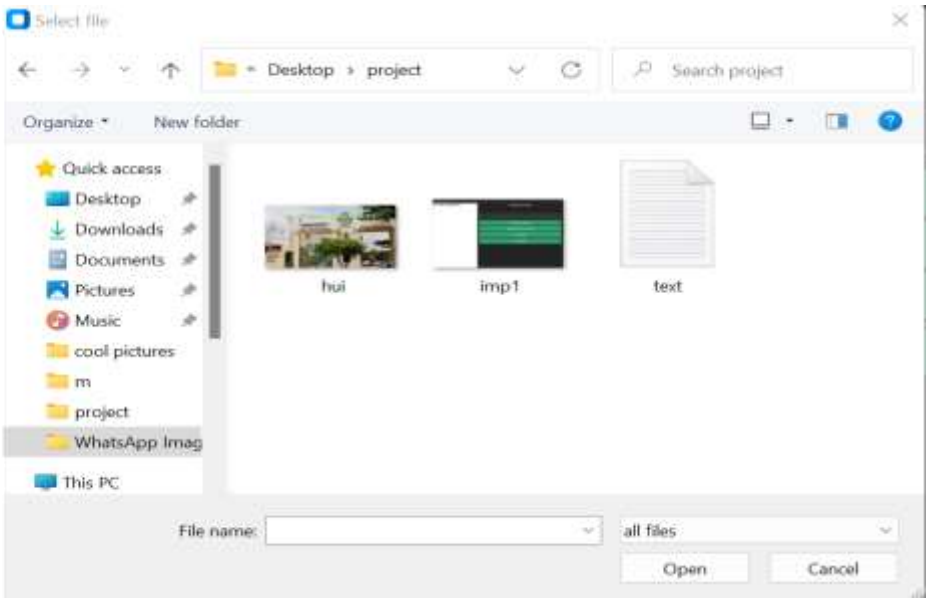

Figure 6. Interface for selecting Image

The Select File to Hide function in Figure 7 allows users to embed a text file within a selected carrier image, ensuring confidentiality and covert transmission of sensitive information.

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

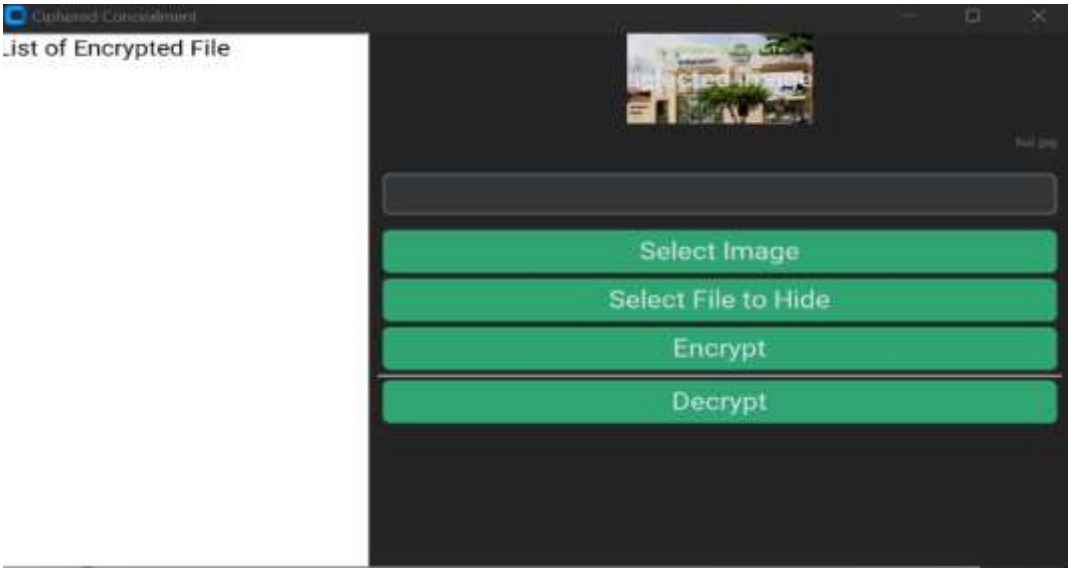

Figure 7. Interface to Select File to Hide

Users begin the process of safely hiding their data inside the selected image by scrolling through the location of the desired text file and verifying their selection. Only text files (.txt) are supported, keeping the tool's capabilities straightforward and focused as displayed in Figure 8.

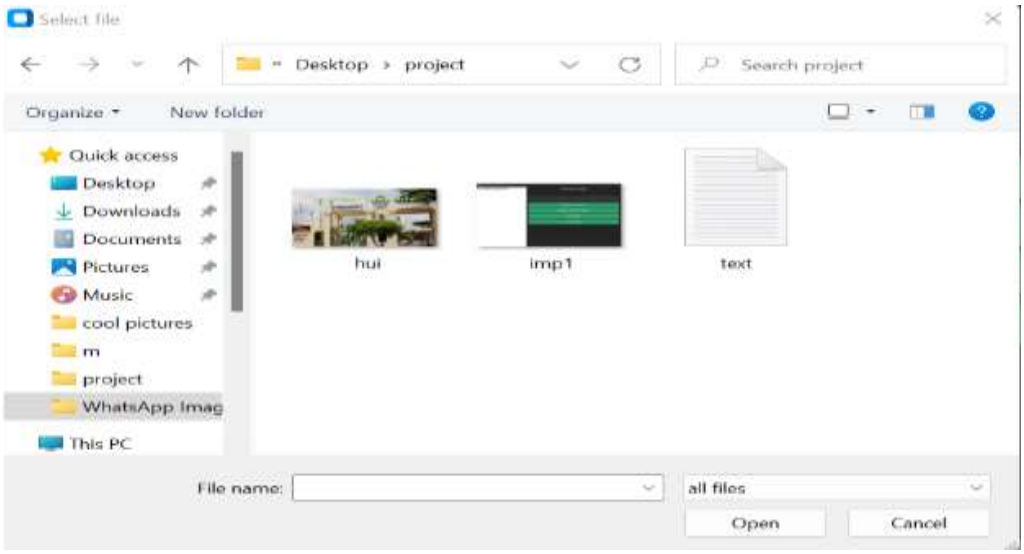

Figure 8. Interface to Select Text file

The.txt file can be hidden in the chosen image by clicking the Encrypt button as shown in Figure 9 and a successful encryption will be indicated by a message.

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

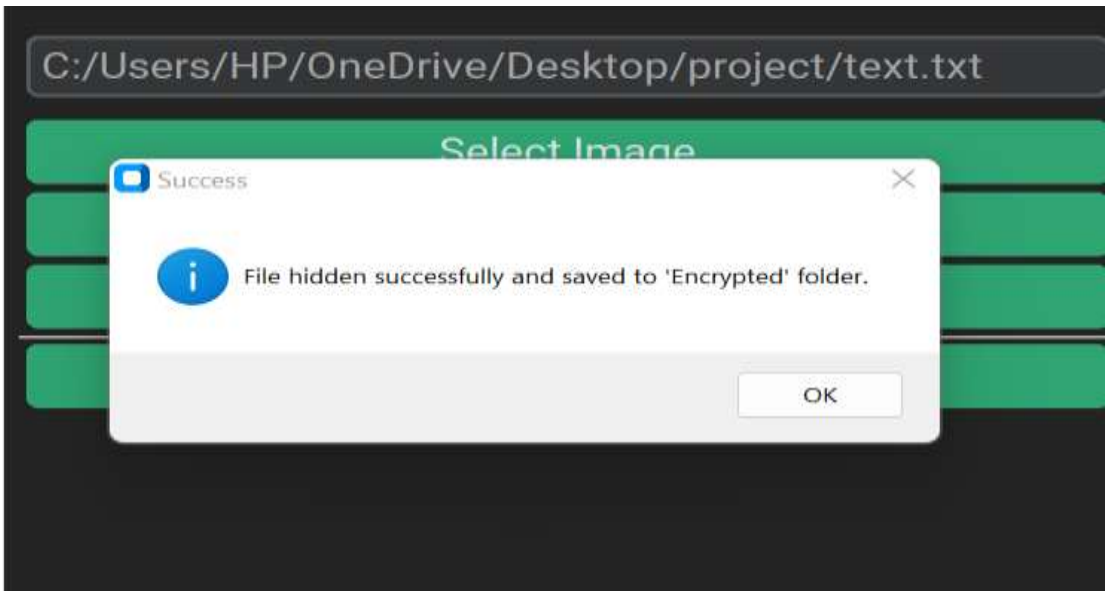

Figure 9. Message which Display Successfully Encrypted file.

The decryption and extraction process interface are shown in Figure 10; the system provides a GUI decryption button to perform the decryption process.

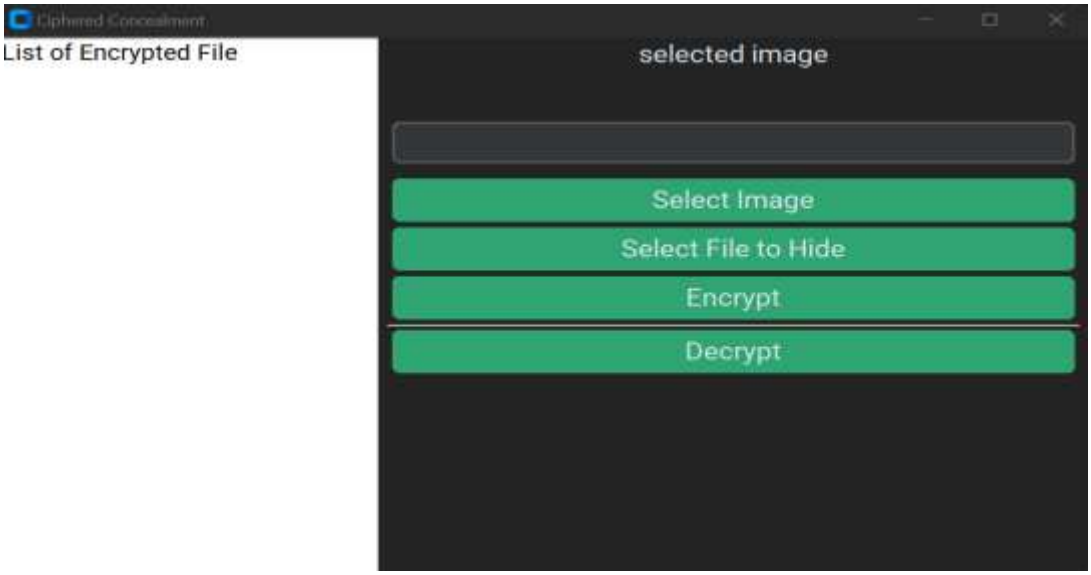

Figure 10. Interface for Decryption of StegoCryptText

To access the hidden.txt file stored at C:\Users\pc\Documents\cypher browse through the file directory and select the desired image. Then, click on the encrypted folder and choose the image used, and it will show a prompt to indicate a successful process as displayed in Figures 11 and 12.

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

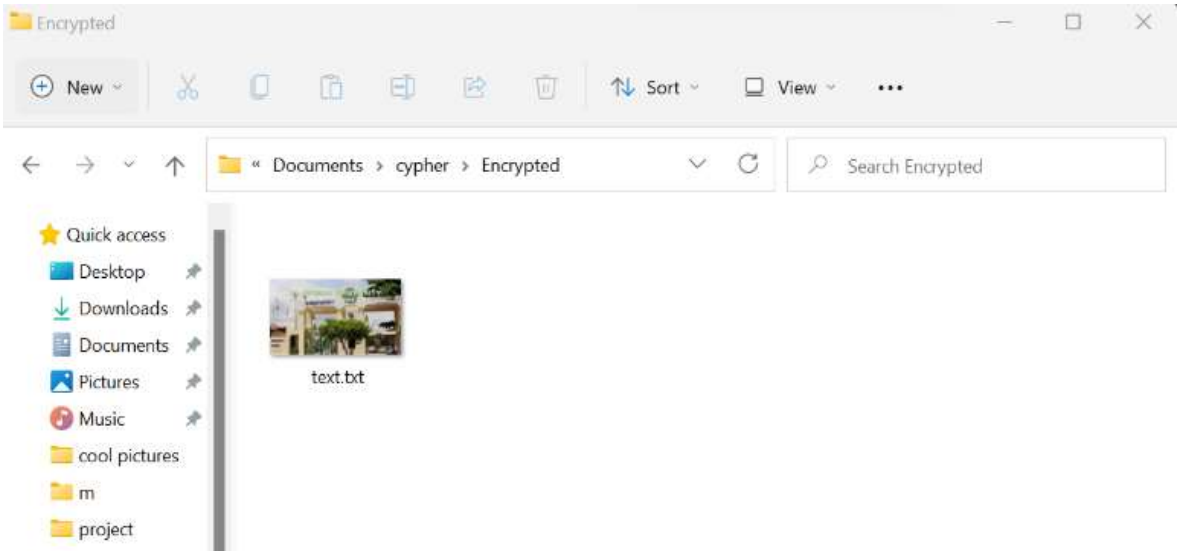

Figure 11. Interface for File Decryption.

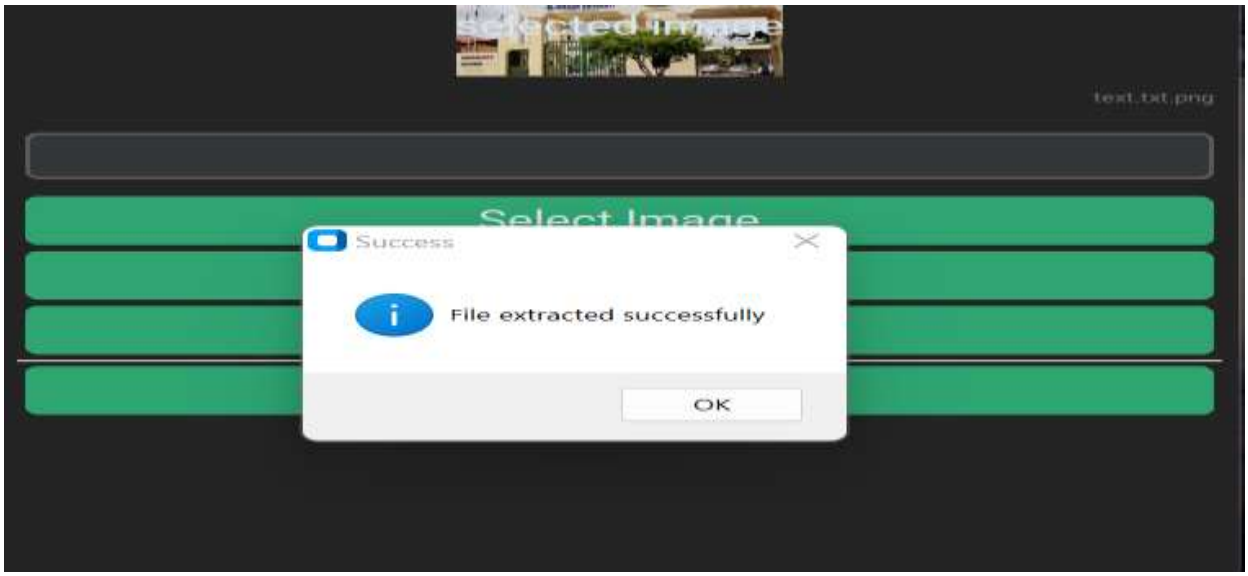

Figure 12. Interface for Successfully Decrypted StegoCryptText

When the user opens the encryption folder, they will see the used image, but they will also see the hidden message. This can be confirmed by comparing the hash functions of the original image with the encrypted output. Upon opening the decryption folder, the user will also be able to see the extracted text from the decryption process as seen in Figure 13.

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

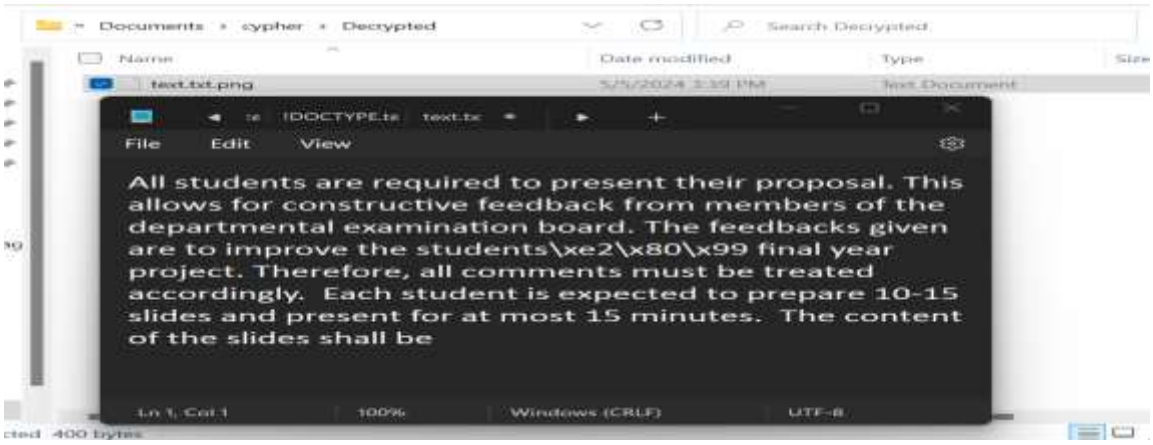

Figure 13. This Interface Displays the Text that Was Decrypted

4.1 The Execution Time Performance of the Developed System

Real-time steganography relies on the time metrics. A successful steganography algorithm must be able to process data quickly in addition to offering strong data confidentiality protection. An Intel Core i7 8th Gen processor with 8GB of RAM is used to run the Python code.

According to Table 3, the combination of LSB and ECC techniques takes 0.1440 seconds less time than the combination of AES, Diffie-Hellman key exchange algorithm, and LSB technique, as well as RSA and LSB technique. This demonstrates the speed efficiency of the former method. Figure 14 shows a more straightforward comparison of all the outcomes for existing and developed steganography techniques.

Table 3: Execution Time Performance of Various Steganography Techniques

Steganography Techniques	**Execution Time performance (sec)**
Proposed Technique (LSB+ECC)	0.1440
LSB steganography + RSA encryption algorithm	0.3335
AES encryption algorithm + Diffie-Hellman key exchange Algorithm + LSB steganography	0.2160

Kasu Journal of Computer Science Vol.2 No.3 [September, 2025], pp. 256-273
https://doi.org//10.47514/kjcs/2025.2.3.005
Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

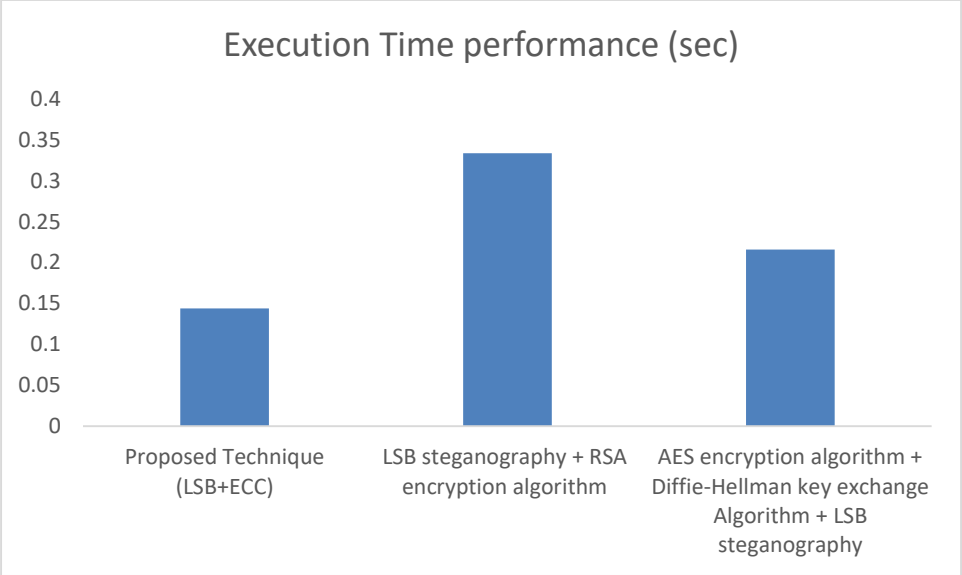

Figure 14. comparison of all the outcomes for existing and developed steganography techniques.

5. Conclusion

This study shows that the effectiveness and security of data protection technologies are greatly increased when steganography and encryption are combined. Performance was specifically enhanced by the use of the Elliptic Curve Cryptography (ECC) algorithm. According to experimental data, RSA with LSB steganography (0.3335 seconds) and AES with LSB steganography and Diffie-Hellman (0.2160 seconds) were not as fast as LSB steganography with ECC (0.1440 seconds). These results show that combining ECC encryption with LSB data-hiding provides faster processing and more robust cover image protection. Future research will use avalanche impact analysis to better assess the suggested method's security. Additionally, research initiatives will focus on integrating machine learning and artificial intelligence tools to improve resilience and adaptability. To boost usability and customization, user interface design enhancements will also be given top priority. Additionally, compression techniques will be investigated to handle bigger file sizes, increasing the applicability to secret data storage and secure communication.

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Reference

Abroshan, H. (2021). A hybrid encryption solution to improve cloud computing security using symmetric and asymmetric cryptography algorithms. *International Journal of Advanced Computer Science and Applications*, *12*(6).

Alanzy, M., Alomrani, R., Alqarni, B., & Almutairi, S. (2023). Image steganography using LSB and hybrid encryption algorithms. *Applied Sciences*, *13*(21), 11771.

Apau, R., & Adomako, C. (2017). Design of image steganography based on RSA algorithm and LSB insertion for android smartphones. *International Journal of Computer Applications*, *164*(1), 0975-8887.

Basholli, F., Juraev, D. A., & Egamberdiev, K. (2024). Framework, tools and challenges in cyber security. *Karshi Multidisciplinary International Scientific Journal*, *1*(1), 94-104.

Bhargava, S., & Mukhija, M. (2019). Hide image and text using LSB, DWT and RSA based on image steganography. *ICTACT Journal on Image & Video Processing*, *9*(3).

Biryukov, A. (2025). Chosen plaintext attack. In *Encyclopedia of Cryptography, Security and Privacy* (pp. 360-360). Cham: Springer Nature Switzerland.

Chen, Z., & Ye, G. (2022). An asymmetric image encryption scheme based on hash SHA-3, RSA and compressive sensing. *Optik*, *267*, 169676.

Hamidu, M., Sarjiyus, O., & Manga, I. (2025). Huffman Encoding for Modified RSA-AES Encrypted Token Compression in Secure Banking Transactions. *Journal of Science Innovation and Technology Research*.

Kapoor, J., & Thakur, D. (2022). Analysis of symmetric and asymmetric key algorithms. In *ICT analysis and applications* (pp. 133-143). Springer Singapore.

Kunhoth, J., Subramanian, N., Al-Maadeed, S., & Bouridane, A. (2023). Video steganography: recent advances and challenges. *Multimedia Tools and Applications*, *82*(27), 41943-41985.

Kyei, R. K., Panford, J. K., Hayfron-Acquah, J. B., Student, P., & Lecturers, S. (2019). Enhancing data security in android smartphones using image steganography, RSA encryption with LSB insertion. *Int. J. Comput. Sci. Info. Secur*, *17*(2).

Majeed, M. A., Sulaiman, R., Shukur, Z., & Hasan, M. K. (2021). A review on text steganography techniques. *Mathematics*, *9*(21), 2829.

Mallick, M. A. I., & Nath, R. (2024). Navigating the cyber security landscape: A comprehensive review of cyber-attacks, emerging trends, and recent developments. *World Scientific News*, *190*(1), 1-69.

Matta, P., Arora, M., & Sharma, D. (2021). A comparative survey on data encryption Techniques: Big data perspective. *Materials today: proceedings*, *46*, 11035-11039.

Movahed, A. B., Movahed, A. B., Nozari, H., & Rahmaty, M. (2024). Security criteria in financial systems in Industry 6.0. In *Advanced Businesses in Industry 6.0* (pp. 62-74). IGI Global.

Online ISSN: 1597-2216
Print ISSN: 1597-2178
www.kjcs.edu.ng

Mua'ad, M., Aldebei, K., & Alqadi, Z. A. (2022). Simple, efficient, highly secure, and multiple purposed method on data cryptography. *Traitement du Signal*, *39*(1), 173.

Ogundokun, R. O., & Abikoye, O. C. (2021). A safe and secured medical textual information using an improved LSB image steganography. *International Journal of Digital Multimedia Broadcasting*, *2021*(1), 8827055.

Ozaif, M., Mustajab, S., & Alam, M. (2024). Exploration of secured data transmission in internet of things: A survey. In *2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT)* (Vol. 5, pp. 106-112). IEEE.

Rakhra, M., Kumar, R., & Walia, H. (2021). A Review on Data hiding using Steganography and Cryptography. In *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 1-4). IEEE.

Rashid, O. F., Subhi, M. A., Hussein, M. K., & Mahdi, M. N. (2024). Enhancing Internet Data Security: A Fusion of Cryptography, Steganography, and Machine Learning Techniques. *Baghdad Science Journal*.

Rinki, K., Verma, P., & Singh, R. K. (2022). A novel matrix multiplication based LSB substitution mechanism for data security and authentication. *Journal of King Saud University-Computer and Information Sciences*, *34*(8), 5510-5524.

Sani, N. F. M., & Nujjaid, M. A. (2025). Efficient Text Data Hiding Technique Using Cryptography and Steganography.

Shaaban, M. A., Alsharkawy, A. S., AbouKreisha, M. T., & Razek, M. A. (2024). Efficient ECC-based authentication scheme for fog-based IoT environment. *arXiv preprint arXiv:2408.02826*.

Sharma, J., & Thapa, R. (2019). Hybrid approach for data security using RSA and LSB Algorithm. In *Proceedings of IOE Graduate Conference* (Vol. 7).

Tajudeen, K. O., Kadri, A. F., & Asaju-Gbolagade, A. W. (2024). An Enhanced Message Encryption Approach Using Residue Number System.

Ullah, A., Haque, M. I., Hossain, M. M., Ahammad, M. S., & Aktar, M. N. (2024). A Novel LSB Steganography Technique for Enhancing Cloud Security. *Journal of Information Security*, *15*(3), 355-377.

Wahab, O. F. A., Khalaf, A. A., Hussein, A. I., & Hamed, H. F. (2021). Hiding data using efficient combination of RSA cryptography, and compression steganography techniques. *IEEE access*, *9*, 31805-31815.

Wandira, A., Nasution, B. B., & Zarlis, M. (2024). Performance Improvement Technique of RSA Algorithm in Document Data Security. In *2024 12th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1-4). IEEE.

Wang, S., Asif, M., Shahzad, M. F., & Ashfaq, M. (2024). Data privacy and cybersecurity challenges in the digital transformation of the banking sector. *Computers & security*, *147*, 104051.