International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V74I2P105

Volume 74 Issue 2, 75-87, February 2026

Original Article

Using Cryptography and Steganography for a Three-Layered Data Hiding System

Isaac Adjaye Aboagye[1], Bright Baba Danjuma[2], Stephen Ofori Yeboah[3], Divine Sackitey[4], Nii Longdon Sowah[5], Joseph Agyiri[6], Kuubore Marcellinus[7], Anthony Adu Gyamfi[8]

[1,2,3,4,5]*Computer Engineering Department, University of Ghana, Accra, Ghana.*
[6]*Department of Computer Science, Accra Technical University, Accra, Ghana.*
[7]*Department of Information Technology Studies, University of Professional Studies, Accra, Ghana.*
[8]*Marine and Freshwater Research Center, Atlantic Technological University, Galway, Ireland.*

[1]*Corresponding Author : iaaboagye@ug.edu.gh*

Received: 16 January 2025 Revised: 11 September 2025 Accepted: 25 November 2025 Published: 14 February 2026

Abstract *- Protecting sensitive information or data against cyber threats is critical for ensuring trust and integrity in many applications. Traditional cryptographic methods cannot withstand sophisticated attacks. To address these challenges, a three-layered security approach is proposed. In the proposed approach, the first layer uses a substitution encryption algorithm followed by AES-128 encryption. This enhanced ciphertext complexity. The encrypted data is then masked within cover media using steganography. This approach provided a secure defense mechanism, ensuring confidentiality, integrity, and availability of information. Through rigorous testing, the strategy proved effective against interception, decryption, and detection, meeting the specific requirements of banking systems and other institutions that handle sensitive data. This will enhance the proactive defense mechanism and ensure customer trust. Adopting this security strategy is vital for mitigating cyber risks and protecting banking and financial data integrity. It offers a robust defense mechanism crucial for safeguarding financial transactions and data security in an increasingly hostile digital landscape.*

Keywords *- Cryptography, Steganography, Cyphertext, ASCII, Cybersecurity.*

1. Introduction

Data security is a crucial aspect of information technology, and it is essential for safeguarding valuable information from cyberattacks and unauthorized users. Hiding data from attackers is the best way to ensure that unauthorized people do not have access to sensitive information [1-3]. Cybersecurity is now a top worry as many businesses and organizations operate using the internet and other digital platforms [4, 5]. According to statistics from the Ghana police service cybercrime unit, Ghana lost over US$200 million to cybercrime between 2016 and 2018 [6]. According to the 2021 Internet Crime Report by the FBI, more than $6.9 billion was lost to cybercrime worldwide in 2021. This amount surpassed losses reported in 2020 ($2 billion) [7]. Conventional security policies often depend on a single layer of defense, typically encryption. An encrypted file, though unreadable, is still identifiable as an encrypted file. This visibility can attract an attacker's attention, prompting attempts at brute-force attacks, cryptanalysis, or exploitation of implementation flaws [8]. On the other hand, steganography conceals data within benign cover files, but its effectiveness is threatened by steganalysis techniques that can detect statistical anomalies indicating the presence of a hidden message [9]. Cryptography and steganography are well-known and widely used techniques to encrypt information. Cryptography involves converting information into a secure and unintelligible format, known as ciphertext, which can only be converted back into its original form, known as plaintext, using a specific decryption key or process. Steganography is the practice of hiding secret information within a seemingly innocent carrier medium, such as an image, audio file, video, or text, in a way that the presence of the hidden information is not easily detectable [10-14]. This research aims to develop an application that will apply both cryptography and steganography to encrypt and decrypt information by implementing a custom substitution encryption algorithm coupled with the AES-128 algorithm. The objectives of the research are to create an application for encrypting customer data using a custom substitution encryption algorithm and also to develop an efficient method to encrypt the database rapidly without compromising file integrity within a short time [15-19]. The target users for the system are bank database administrators and other personnel authorized to handle customer data and sensitive information. Many studies have suggested hybrid systems, but they often fail to provide a comprehensive analysis [20, 21]. This research will evaluate a novel three-layered data hiding system

to address the identified gaps. In this research, an application for encrypting customer data is created using a custom substitution encryption algorithm. Also, an efficient method to encrypt a database without compromising file integrity within a short time is designed. A custom substitution cipher will serve as a pre-processing step to enhance data entropy. The output will be encrypted using the AES-128 algorithm, and the final ciphertext is concealed. The research is grouped into five sections. Section I introduces the concept of data security. In Section II, we will review related research works that are relevant to the study and further point out the strengths and limitations of the related works. Section III describes the system design and development, and the components used in the design implementation and design process of the system. It will provide a holistic view of the system, its core functionality, requirements, and specifications. Section IV focuses on the integration of the various components and subsystems, testing, analysis, and discussion of results. Section V is the conclusion and recommendations. It will highlight the key accomplishments of the work and possible recommendations for further research work.

2. Literature Review

In this section, a literature review of existing related work is discussed in detail. The concept of using custom or non-standard algorithms alongside established ones will be explored. Furthermore, we highlight the strengths and limitations of the available proposed approaches. Ahmed A. et al. [22] applied cryptography and steganography to hide secret information in an image, audio, or video. The message is encrypted by using the AES algorithm, and the key is hashed using SHA-2 to prevent attacks. They performed some modifications on the LSB algorithm by adding a key to make the hiding process non-sequential. While they reported strong security, their approach was limited by the low data-carrying capacity of the chosen cover media and did not explore text-based steganography. Swati S. P. et al. [23] presented an enhanced multi-level secret data hiding that integrates two different methods of encryption, namely visual cryptography and steganography. For the pre-processing step, they used halftoning to reduce the pixels and simplify the processing. After that, visual cryptography was performed, which produces the shares, which form the first level of security, and then steganography, in which they hide the shares in different media like images, audio, and video to obtain multi-level secret data hiding, which improves the security over the network. The limitation of this work is that the capacity of data that their proposed method can hide is too small, making their system less versatile. Ramadhan J. M. et al. [24] proposed a secure video steganography algorithm based on the principle of linear block code. In their project, they used nine uncompressed video sequences as cover data and a binary image logo as a secret message. The pixels' positions of both cover videos and a secret message are randomly reordered by using a private key to improve the system's security. The secret message is encoded by applying the Hamming code (7, 4) before the embedding process to make the message even more secure. The result of the encoded message will be added to randomly generated values by using the XOR function. Regarding the system's quality, the Peak Signal-to-Noise Ratio (PSNR) of stego videos is above 51 dB, which is close to the original video quality. The challenge of this work is that the multi-key method adds complexity to key management and calculations using the Hamming Code (7, 4). Also, the steganography scheme increases the capacity by up to 90 Kbits in each frame, which causes a slight degradation of the visual quality of the image.

Padmavathi B., and Ranjitha K.S., [25] presented a work that focused on the combination of cryptography and steganography to secure the data while transmitting in the network. They implemented three encryption techniques using DES, AES, and RSA algorithms along with steganographic algorithms like the LSB substitution technique and compared the performance of the encryption techniques based on the analysis of their execution time at the time of encryption and decryption process, and also their buffer size experimentally.

Based on the experimental result, it was concluded that the AES algorithm consumes the least encryption and decryption time and buffer usage compared to the DES and RSA algorithms. RSA consumes more encryption time, and its buffer usage is also very high, a conclusion that informs the choice of AES in the present study. Their analysis did not extend to multi-layered cryptographic approaches or alternative steganographic methods. Our proposed system leverages a custom substitution cipher not as a standalone solution, but as a primer to make the input to the AES algorithm more random, thereby strengthening the overall system. The system will present a reproducible framework with clear methods, empirical analysis, and measurable benchmarks.

3. System Design and Development

The system design employs both cryptography and steganography techniques to secure information more robustly, even though each technique can work strongly independently. The system to be developed is divided into three parts to provide an engineering solution to the problem. These are the cryptographic subsystem, the steganographic subsystem, and the GUI application that will host the cryptographic and steganographic systems as a single unit. The GUI application consists of a user interface for data entry, data encryption, and data embedding, as well as the respective reversal processes. Figures 1, 2, and 3 below depict the system architecture for cryptography, steganography, and the integrated system, respectively, that will be implemented in this research.

3.1. Cryptographic System

The cryptographic system consists of the custom encryption algorithm and the AES-128 algorithm. The custom

algorithm is a symmetric encryption technique that uses the substitution method in combination with a 256-bit key to transform plaintext to ciphertext and vice versa. It is based on the XOR logical operations [26].

Table 1. Truth table for XOR with sample plain text (0101)

P (Plain text)	K (Key)	C (Cipher) = P XOR K	K (Key)	P (Plain text) = C XOR K
0	0	0	0	0
1	0	1	0	1
0	1	1	1	0
1	1	0	1	1

3.1.1. Layer 1: Custom Substitution Cipher

This initial layer employs a lightweight XOR-based symmetric cipher inspired by the One-Time Pad (OTP) principle. It obfuscates the plaintext rapidly and increases its entropy before feeding it into the AES module [27].

Algorithm 1: Custom Substitution Encryption/ Decryption

```
Procedure Encrypt(Plaintext, Key):
    Plaintext_Binary = ConvertToBinary(Plaintext)
    Key_Bin = ConvertToBinary(Key)
    Extended_Key = ExtendKey (Key_Bin, length (Plaintext_Binary))
    Ciphertext_Binary = Plaintext_Binary XOR Extended_Key
    Ciphertext = ConvertFromBinary(Ciphertext_Binary)
    return Ciphertext
Procedure Decrypt(Ciphertext, Key):
    return Encrypt(Ciphertext, Key)
```

The key is a 256-bit string of random characters. The encryption process guarantees high variability and minimal redundancy in the resulting ciphertext. The custom algorithm can be applied to any digital or binary information, including text-based data encoded with 8-bit ASCII code. In this case, the encryption key, which is a One Time Pad (OTP), can be represented as a randomly generated string of characters for each new encryption. In the Encryption section, the message will be converted from ASCII characters to binary.

The OTP key is used for encryption. An XOR operation is performed between the key and the plaintext in binary to produce the ciphertext. To decrypt, the same key is used.

3.1.2. Layer 2: AES-128 Encryption

The second layer uses AES-128 in Cipher Block Chaining (CBC) mode [28]. CBC mode is preferred over the simpler Electronic Codebook (ECB) mode because it links consecutive blocks, ensuring that identical plaintext blocks result in different ciphertext blocks and providing better security against pattern analysis.

3.2. Steganographic System

The steganographic module complements encryption by concealing the encrypted content within a carrier medium. Two techniques were used based on the type of media. A Steganographic algorithm combines the cover message with the embedded message, which is something to be hidden in the cover. The algorithm may or may not use a Steganographic key (stego key), which is additional secret data that may be needed in the hidden process.

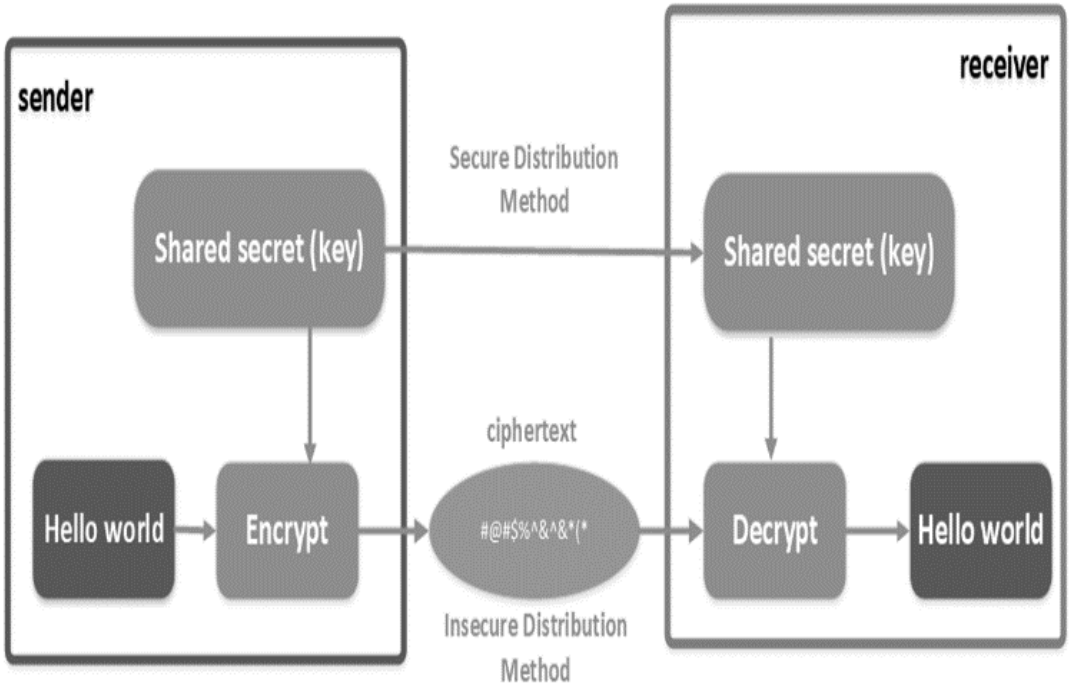

Fig. 1 System architecture for cryptography

3.2.1. Text Steganography: Whitespace Manipulation

White space steganography was implemented by using the SNOW library. The SNOW library provides features used to hide messages in text files by appending tabs and spaces to the end of lines, as well as to extract messages from files containing hidden messages. SNOW encrypts messages with a 64-bit block cipher. It compresses data using the basic Huffman encoding, with tables optimized for English text. The SNOW conceals content in ASCII text by adding spaces to line ends, hiding within text viewers. This encryption prevents casual detection and impedes readability.

A shared key shuffles the position of whitespace in each embedding and provides a different character-binary mapping, which makes it more difficult to guess the hidden data characters. SNOW employs ICE encryption in 1-bit Cipher Feedback (CFB) mode, supporting variable-length keys up to 1170 characters, and conceals messages if specified. The data is concealed in the text file by appending sequences of up to 7 spaces separated by tabs [29].

3.2.2. Image Steganography: Least Significant Bit (LSB)

A ciphertext intended for concealment is first converted into an array of its ASCII representations. The cover image provides 3 bits per pixel, corresponding to the Red, Green, and Blue channels. To embed the data, three consecutive pixels are grouped, providing a total of 9 bits for encoding.

The total number of pixels traversed is tracked during embedding. A key is generated to guide the hiding process and ensure controlled data embedding.

Algorithm 2: LSB Embedding

```
Procedure Embed(Ciphertext, CoverImage):
  Ciphertext_Binary = ConvertToBinary(Ciphertext)
  DataIndex = 0
  For each Pixel in CoverImage:
    For each ColorChannel (R, G, B) in Pixel:
      If Data_Index < length(Ciphertext_Bin):
        ColorChannel = ColorChannel & 254
        ColorChannel = ColorChannel | Ciphertext_Bin
[DataIndex]
        DataIndex = DataIndex + 1
      Else:
        return StegoImage
  return StegoImage
```

The ciphertext can be decrypted by opening the stego-image and converting it into. The secret key can be used to obtain the hidden message in ASCII format from the stego-image. The ASCII bits are grouped into groups of 8, which are converted back into the ciphertext and further decrypted by the AES algorithm.

The dynamic combination of these security components is an innovative way of improving overall data security. By strategically placing spaces within ASCII text, text steganography conceals information and effectively clouds the message.

The Least Significant Bit (LSB) substitution is used in image steganography to embed encrypted content while maintaining image integrity.

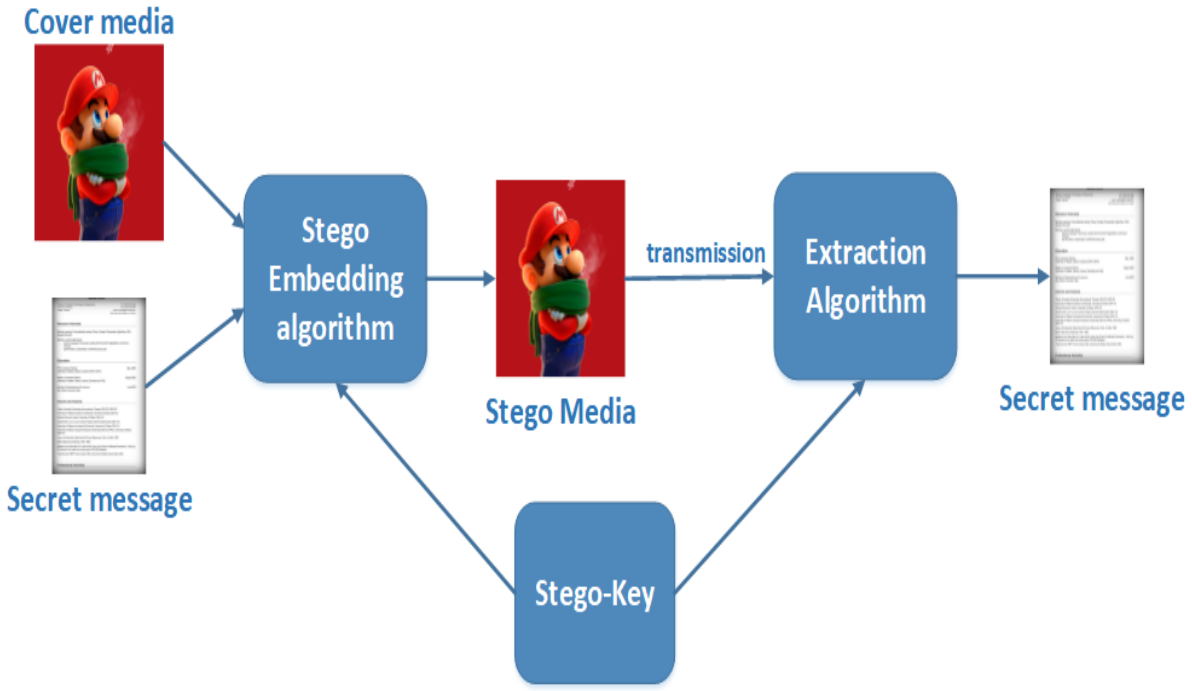

Fig. 2 System architecture for steganography

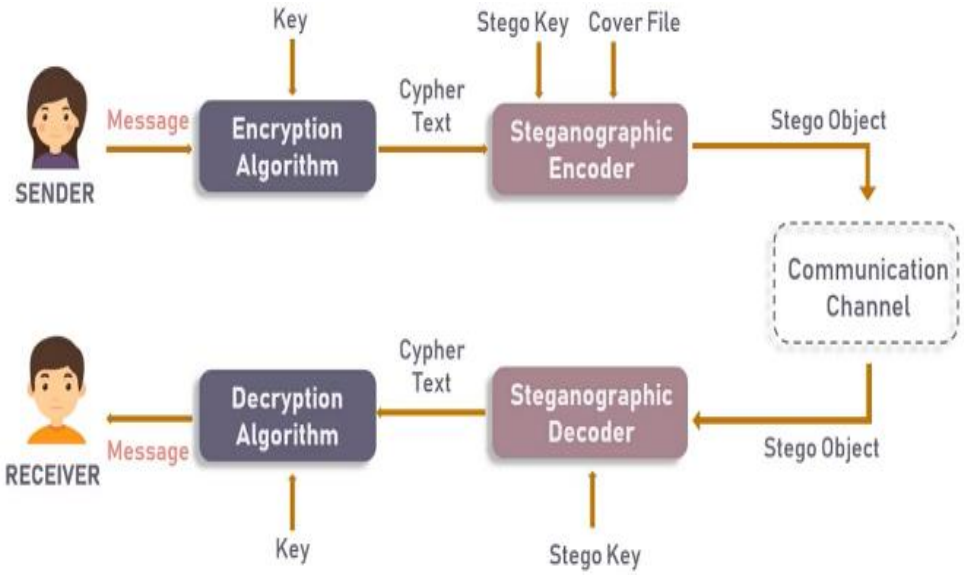

Fig. 3 System architecture for integrated system

The authorized user in this case acts as both the sender and receiver since the system is protecting data in storage.

3.3. Project Flow Diagram

Cryptography will be performed first on the file of interest. The ciphertext is what will be used for the steganography process. The ciphertext will be hidden in a cover medium. The media will now be sent to the receiver.

The receiver will retrieve the ciphertext from the image and then convert the ciphertext into plain text. Successful encryption and decryption of the database should be done in the shortest possible time.

3.4. Data Source, Type, and Format

All the data used in training and testing our system was generated by synthetic data. Many different datasets were generated using SQLite commands. These datasets were designed in such a way as to mimic real-life sensitive data to make the testing more accurate. In effect, synthetic data and original data should produce very similar results when subjected to the same statistical analysis.

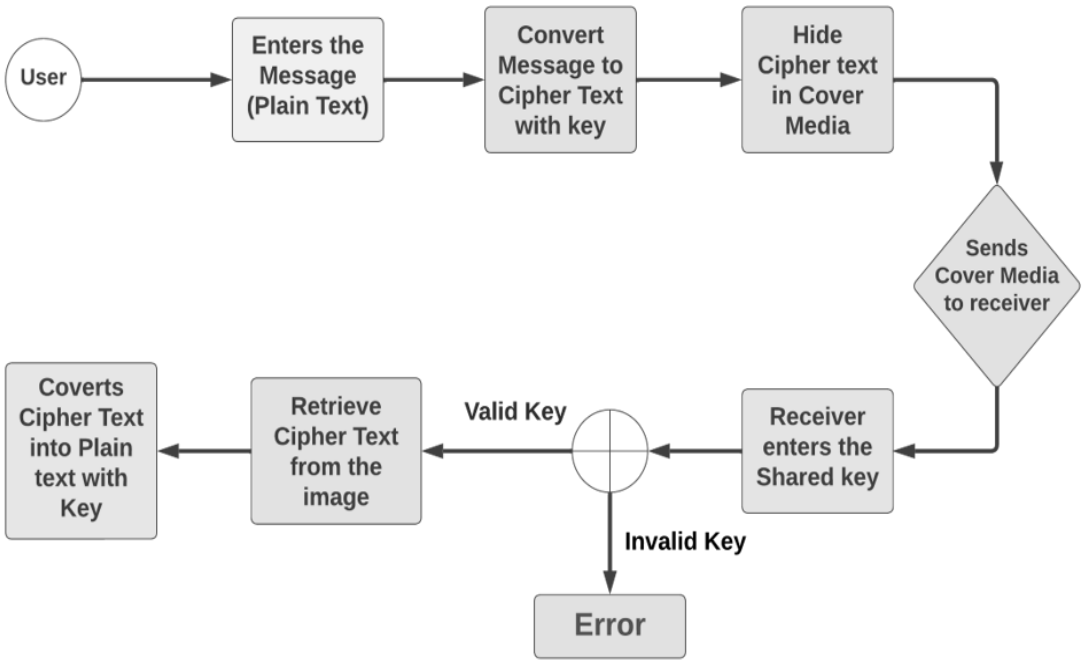

Fig. 4 Project flow diagram

	id	level	brand	card_number	expiration	cvv	card_type
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	5965147	PLATINUM	AMERICAN EXPRESS	1543194391893990	8/28	643	CREDIT
2	7727868	CLASSIC	MASTERCARD	4413739154705400	5/26	626	CREDIT
3	8803362	PLATINUM	MASTERCARD	8121502739271050	2/28	217	DEBIT
4	6665560	PLATINUM	VISA	9309004475796900	5/28	435	CREDIT
5	8520047	PLATINUM	VISA	1973849125191190	12/25	609	DEBIT
6	3881021	CLASSIC	MASTERCARD	9315938350492760	12/28	290	CREDIT
7	9573326	GOLD	MASTERCARD	3612659176397020	2/27	945	DEBIT
8	1955039	GOLD	MASTERCARD	8386612815988490	9/26	454	DEBIT
9	5032435	GOLD	MASTERCARD	8080128605509220	10/27	796	CREDIT
10	9627093	CLASSIC	AMERICAN EXPRESS	5243301130254900	9/25	557	CREDIT
11	4539897	GOLD	AMERICAN EXPRESS	1786330250254400	1/29	952	DEBIT
12	4879972	PLATINUM	AMERICAN EXPRESS	3921156268067450	2/29	169	CREDIT
13	1191175	CLASSIC	VISA	8476428868812130	1/27	910	CREDIT
14	4019190	CLASSIC	MASTERCARD	2852225458408340	8/26	141	CREDIT
15	6584093	PLATINUM	MASTERCARD	4925617653233330	11/26	541	CREDIT
16	1324909	PLATINUM	AMERICAN EXPRESS	5109610708799570	4/26	758	DEBIT
17	9845210	PLATINUM	VISA	2386985250143220	12/30	941	DEBIT
18	4270298	PLATINUM	MASTERCARD	6083176425023250	5/30	917	CREDIT
19	5731027	CLASSIC	AMERICAN EXPRESS	3663158419276930	10/28	382	CREDIT
20	5160471	PLATINUM	MASTERCARD	8792884197173480	12/27	510	CREDIT
21	5022418	PLATINUM	VISA	7170205379957930	11/25	451	CREDIT
22	1133320	GOLD	VISA	7246957004026340	4/28	554	CREDIT
23	8384984	PLATINUM	AMERICAN EXPRESS	1861407468151900	9/30	196	DEBIT

Fig. 5 Sample dataset 1

Table: customer_details — Filter in any column

	customer_id	first_name	last_name	email	city	state	zip	country
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
279	6873788	Anastasia	Yeldham	ayeldham7q@youtube.com	Phoenix	AZ	85010	US
280	5425749	Augustine	Shirrell	ashirrell7r@google.pl	Richmond	VA	23293	US
281	4741816	Son	Baumert	sbaumert7s@symantec.com	Los Angeles	CA	90087	US
282	8006115	Myrtle	Papierz	mpapierz7t@wufoo.com	Grand Rapids	MI	49505	US
283	2682840	Dorry	Crates	dcrates7u@nydailynews.com	Buffalo	NY	14215	US
284	6612821	Lura	McMoyer	lmcmoyer7v@usa.gov	Santa Monica	CA	90410	US
285	1880029	Hermon	Rait	hrait7w@unicef.org	Boston	MA	02203	US
286	9642108	Valera	Cuniffe	vcuniffe7x@people.com.cn	Gadsden	AL	35905	US
287	4442276	Bab	Rubertelli	brubertelli7y@zdnet.com	Amarillo	TX	79188	US
288	3769829	Bobinette	Atty	batty7z@time.com	Charlotte	NC	28272	US
289	6747106	Goldina	Torbard	gtorbard80@illinois.edu	Cincinnati	OH	45254	US
290	1415987	Sal	Dolley	sdolley81@house.gov	Staten Island	NY	10310	US
291	7863232	Any	Forber	aforber82@blogger.com	Savannah	GA	31422	US
292	5715964	Darcey	Duckit	dduckit83@slate.com	Fairbanks	AK	99790	US
293	1125596	Cristal	Daal	cdaal84@nih.gov	Dallas	TX	75379	US
294	9994559	Staffard	Winkworth	swinkworth85@zimbio.com	El Paso	TX	79923	US
295	1500691	Jaimie	Grange	jgrange86@state.gov	Jamaica	NY	11431	US
296	9586078	Kenyon	Melledy	kmelledy87@ox.ac.uk	San Jose	CA	95133	US
297	6342472	Arielle	Ivey	aivey88@independent.co.uk	Chicago	IL	60646	US
298	7896685	Arden	Pergens	apergens89@samsung.com	Akron	OH	44310	US
299	7188740	Rafa	Tyrie	rtyrie8a@odnoklassniki.ru	Suffolk	VA	23436	US
300	2699690	Christy	Carden	ccarden8b@sciencedirect.com	Ridgely	MD	21684	US
301	4901189	Darcy	Horsell	dhorsell8c@chronoengine.com	San Antonio	TX	78291	US

Fig. 6 Sample dataset 2

To simulate real-world applications, the system was tested using synthetic datasets representing sensitive data such as customer profiles, transaction logs, and credential files, as seen in Figures 5 and 6.

Performance was assessed using the following metrics. Encryption and decryption times were measured in seconds to evaluate efficiency across various file sizes.

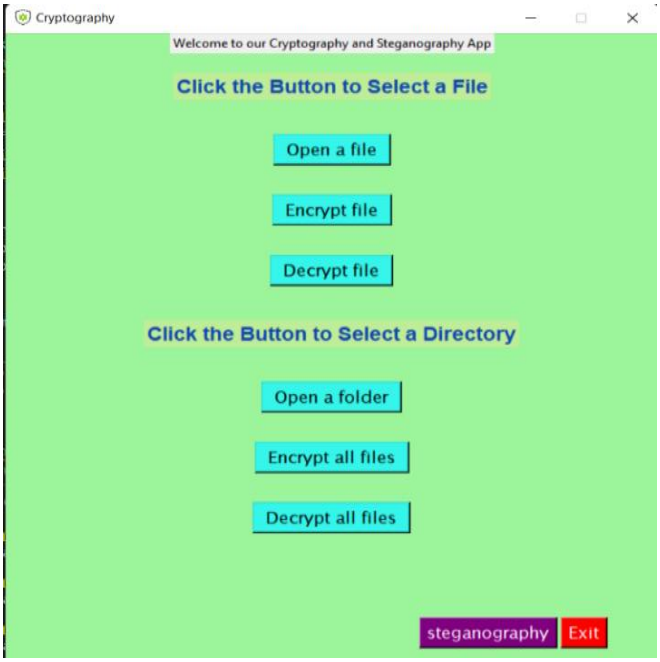

Fig. 7 Cryptography home page

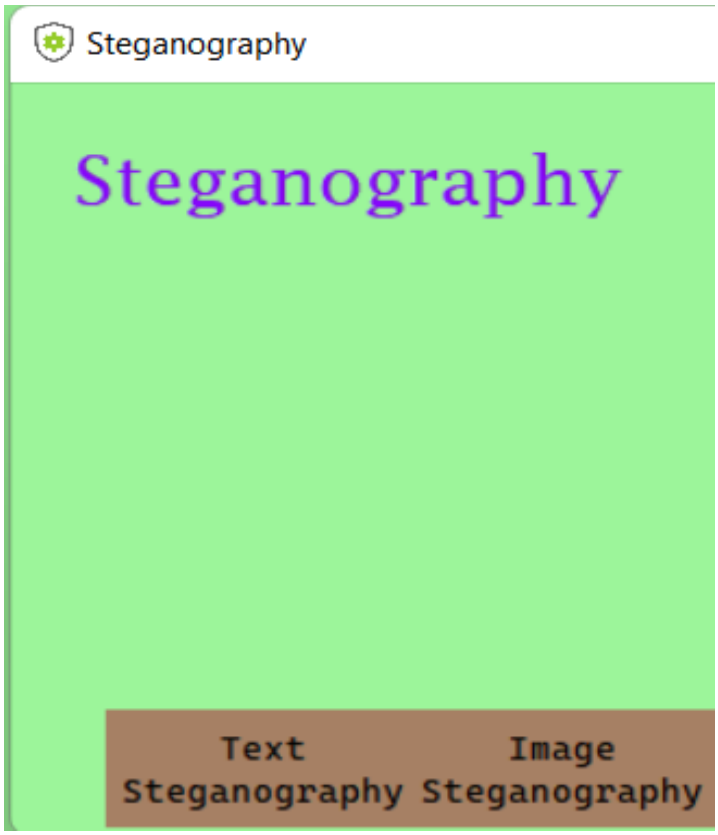

Fig. 8 Steganography home page

4. System Implementation and Testing

The three subsystems, cryptographic, steganographic, and GUI applications, were implemented and integrated to form a fully functional solution. The GUI application front end was built with Tkinter. The cryptographic system developed consists of two different algorithms integrated into one: a custom substitution encryption algorithm coupled with the standard AES-128 encryption. AES-128 cryptography was chosen for the project since it has proven to be strong if reasonable key sizes are used. It offers much faster processing speeds and lower resource requirements than the 192- and 256-bit variants, which will be key to the applications of this study.

The steganographic system employed two different techniques: text steganography using the whitespace technique, and image steganography using the LSB technique.A key retrieval system was developed and integrated into the system for the retrieval of secret keys in case the user forgets at any point in time. The user would, however, have to go through a strict verification procedure to ensure the key gets into safe hands. To ensure that the system is accessed only by the right people, the first layer of the application is authentication.

The user first enters his/her access credentials, which are already recognized by the institution, to verify his/her identity. Access is granted to the user if the login credentials are correct. The user is taken to the home page of the application, where the cryptography and steganography tools are displayed. Cryptography is performed first on the file of interest.

The user selects the file to be hidden, enters a secret key, and clicks the "Encode" button that finalizes the encryption process. This same key will be used for the decryption process. This ciphertext is what will be used for the steganography process. If an attacker can extract the hidden file from the cover object, it will be in an unreadable format, hence rendering it useless to the attacker.

4.1. Testing and Results

The dataset being considered for this test is named "customerdata.txt". This is one of the synthetic datasets generated earlier. This dataset was designed to mimic real-life sensitive customer data that a bank may store to make the testing more accurate. Attackers attempting to breach financial systems may target this confidential customer information. The contents of this dataset are shown below.

File Edit View

Account No	DATE	TRANSACTION DETAILS	CHQ.NO.	VALUE DATE	WITHDRAWAL AMT	DEPOSIT AMT
409000611074'	29-Jun-17	TRF FROM Indiaforensic SERVICES		29-Jun-17		1,000,
409000611074'	05-Jul-17	TRF FROM Indiaforensic SERVICES		05-Jul-17		1,000,
409000611074'	18-Jul-17	FDRL/INTERNAL FUND TRANSFE		18-Jul-17		500,000.00
409000611074'	01-Aug-17	TRF FRM Indiaforensic SERVICES		01-Aug-17		3,000,000.00
409000611074'	16-Aug-17	FDRL/INTERNAL FUND TRANSFE		16-Aug-17		500,000.00
409000611074'	16-Aug-17	FDRL/INTERNAL FUND TRANSFE		16-Aug-17		500,000.00
409000611074'	16-Aug-17	FDRL/INTERNAL FUND TRANSFE		16-Aug-17		500,000.00
409000611074'	16-Aug-17	FDRL/INTERNAL FUND TRANSFE		16-Aug-17		500,000.00
409000611074'	16-Aug-17	FDRL/INTERNAL FUND TRANSFE		16-Aug-17		500,000.00
409000611074'	16-Aug-17	FDRL/INTERNAL FUND TRANSFE		16-Aug-17		500,000.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL01071		16-Aug-17		133,900.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL02071		16-Aug-17		18,000.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL03071		16-Aug-17		5,000.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL04071		16-Aug-17		195,800.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL05071		16-Aug-17		81,600.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL06071		16-Aug-17		41,800.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL07071		16-Aug-17		98,500.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL10071		16-Aug-17		143,800.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL11071		16-Aug-17		331,650.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL12071		16-Aug-17		129,000.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL13071		16-Aug-17		230,013.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL14071		16-Aug-17		367,900.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL15071		16-Aug-17		108,000.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL16071		16-Aug-17		64,800.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL17071		16-Aug-17		141,000.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL18071		16-Aug-17		61,750.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL19071		16-Aug-17		67,920.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL20071		16-Aug-17		78,100.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL21071		16-Aug-17		35,650.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL22071		16-Aug-17		206,000.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL24071		16-Aug-17		35,300.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL25071		16-Aug-17		49,800.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL26071		16-Aug-17		53,000.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL27071		16-Aug-17		91,300.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL28071		16-Aug-17		57,499.00
409000611074'	16-Aug-17	INDO GIBL Indiaforensic STL30071		16-Aug-17		20,000.00

Fig. 9 Contents of customerdata.txt

When the application is launched in the cryptography section, "customerdata.txt" is selected, and a 16-bit long secret key is entered because that is the key length for the AES-128 algorithm. The user clicks "Encrypt File," and the file is encrypted. The file extension after the encryption process changes from ".txt" to ".txt.enc," indicating encryption has taken place. The contents of the encrypted file are shown below:

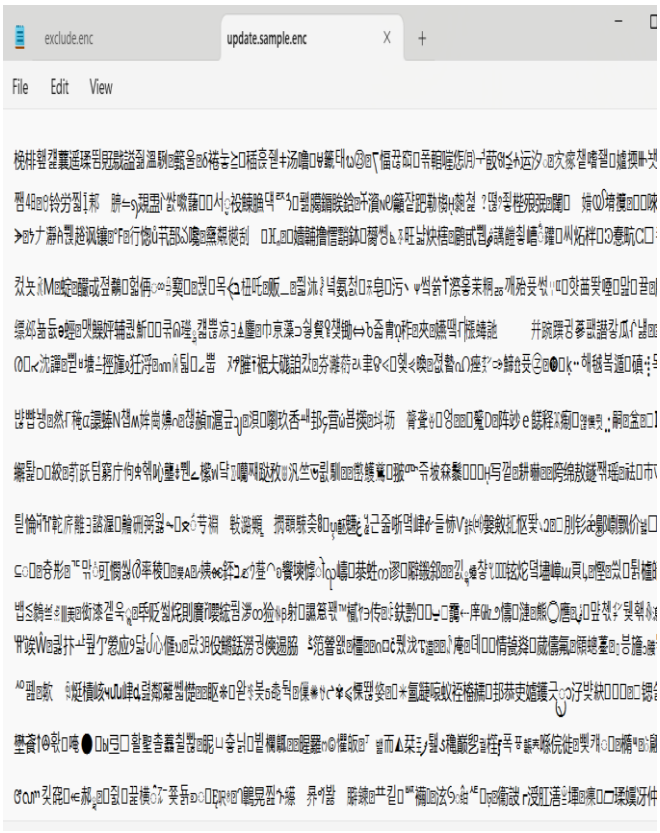

Fig. 10 Contents of the encrypted file

Text steganography will be used for the embedding process. An unsuspicious file named "loanterms.txt" will be used as the cover file for the embedding process. "loanterms.txt," which contains the terms and conditions for loan agreements, was used mainly to avoid suspicion, which is the key idea behind steganography in general. Its content does not look useful to an outsider, and hence may not trigger an attack on the file. Its contents are shown below:

In the text steganography section, "loanterms.txt" is selected as the cover file, "customerdata.txt.enc" is selected as the file to be embedded, and the user chooses a secret key for the embedding process.

The user clicks "Encode data," and the file is embedded in "loanterms.txt". The contents of "loanterms.txt" do not change after the embedding process, making the entire process unsuspecting.

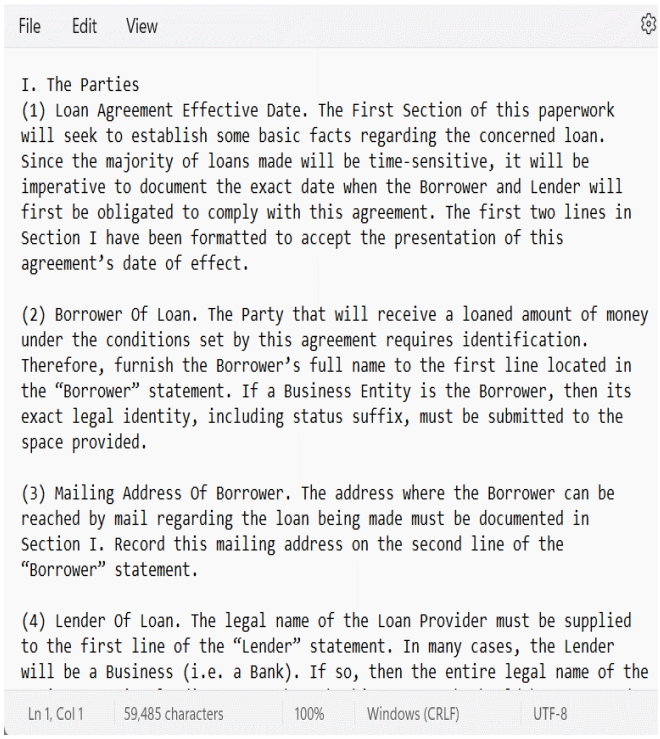

Fig. 11 Contents of "loanterms.txt"

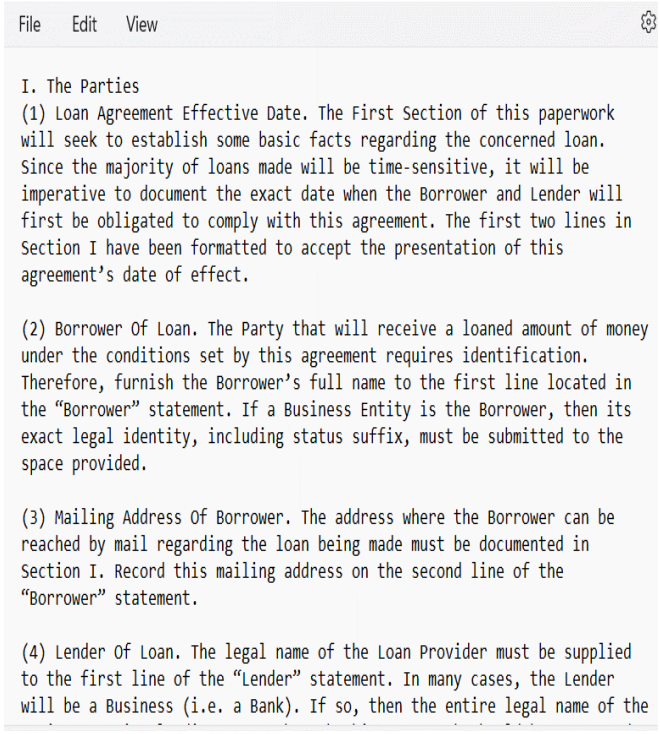

Fig. 12 Contents of "loanterms.txt" after the embedding process

This unsuspicious stego file "loanterms.txt" is uploaded to a file storage system for a bank. An attacker who manages to penetrate the file storage system will not be able to get the actual encrypted file in "loanterms.txt" since it looks very suspicious. Even if the attacker can get the encrypted file, he

would still have to convert it back into plaintext, which is nearly impossible. The integrated system was evaluated using synthetic datasets designed to mimic real-world data such as financial records, login credentials, and customer transactions. The focus of this section is to present empirical performance results and a direct comparison between the proposed three-layer architecture and a baseline two-layer system (AES-128 + LSB only) to highlight the enhancements. Tests were conducted across varying file types and sizes to assess scalability and efficiency.

Table 2. Custom encryption algorithm combined with AES module

File Name	Original Size (Kb)	Encryption Time (s)	Decryption Time (s)	File Size After Encryption (Kb)	File Size After Decryption (Kb)
Customer-data.txt	9.14	0.0457	0.0021	12.02	9.14
Employee-data.pdf	813	0.2886	0.2321	817	813
Login-credentials.db	57.22	0.0672	0.0611	61.2	57.22
Carddetails.txt	7505.43	1.5382	0.7655	7612.2	7505.43
Overdraftlist.db	4672.25	1.1322	0.6303	4714.7	4672.25
Account -numbers.pdf	1550.5	0.3886	0.2973	1566.1	1550.5
Creditlist.txt	789.3	0.2563	0.1922	796.9	789.3

From Table 2, the results indicate that the encryption process is very fast, as the maximum encryption time recorded is less than 2 seconds.

The encryption time for larger files was a bit longer than for the smaller files. Secondly, the decryption times were less than the respective encryption times. The file sizes increased after encryption and reduced to the original file sizes after decryption. Results from Table 3 indicate that the encryption process is very fast, as the maximum encryption time recorded is less than 2 seconds.

It is clear from the table that file size and encryption time are directly proportional. This means that the larger the file, the longer it takes to be encrypted. Also, it can be observed that the decryption times recorded are generally less than the respective encryption times. Additionally, the file sizes increase after encryption, and this is because of the secret files being embedded in them. However, the original file sizes are not attained after decryption, though the values are very close. This is because even after the extraction of the secret file, there is still some noise left in the file, which accounts for the extra size.

Table 3. Text steganography module (whitespace)

File Name	Original Size (Kb)	Encryption Time (s)	Decryption Time (s)	File Size After Embedding (Kb)	File Size After Extraction (Kb)
SecurityPolicy.txt	1.79	0.0312	0.0065	48.4	1.91
LoanTerms.txt	12.9	0.0750	0.0073	59.3	13.4
CustomerData.db	308	0.3886	0.0613	386	312
Bank.txt	6500	1.7865	0.6862	6845	6598.2
Employeenames.pdf	81	0.2914	0.0953	964.4	86
Salarystructure.pdf	55	0.0925	0.0097	92.1	62.8
Creditlist.txt	124.7	0.3115	0.0598	162.7	131.3

Table 4. Image steganography module (LSB)

File Name	Original Size (Kb)	Encryption Time (s)	Decryption Time (s)	PSNR (dB)	PSNR after Encryption (dB)	PSNR after Decryption (dB)	Image Size After Embedding (Kb)	Image Size After Extraction (Kb)
Image1.png	104	0.2914	0.0953	80.5908	78.2345	79.8721	1910	108
Image2.png	3.45	0.0412	0.0321	45.2366	44.7893	45.1187	12.6	3.56
Image3.png	6.01	0.0455	0.0342	47.7828	46.9987	47.4521	27.2	6.15
Image4.png	12.6	0.0478	0.0351	48.2761	47.1246	47.8965	105	13.2
Image5.png	15.1	0.0511	0.0497	51.6453	50.7863	51.2418	110	16.2
Image6.png	18.57	0.0566	0.0518	53.8756	52.3765	53.0213	112.8	19.44
Image7.png	23.4	0.0763	0.0682	62.4929	61.0098	62.0123	139	24.3

From Table 4, the results recorded indicate that the embedding process is very fast, as the maximum encryption time recorded is less than 1 second for the files used. It is clear from the table that file size and PSNR value are directly proportional. This is logically correct, as larger files have more pixels for the LSB substitution process; hence, there is less distortion in image quality, leading to a higher PSNR value. Additionally, it can be observed that file sizes increase after encryption because of the secret files embedded in them. A linear correlation between file size and encryption time, confirming that the cryptographic module is scalable and consistent, is observed in Figure 13.

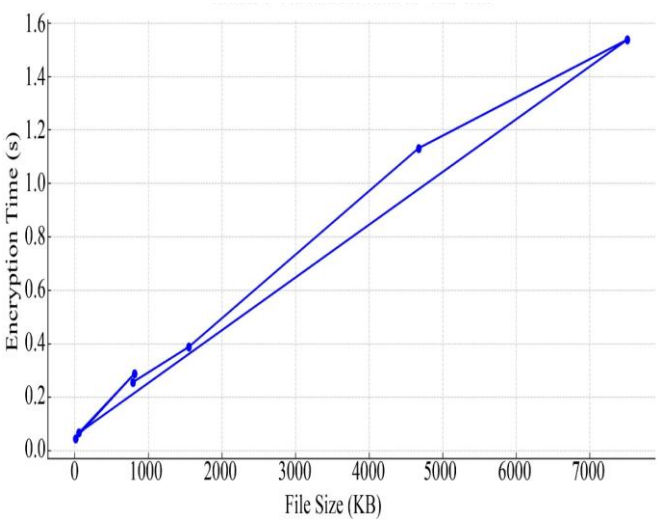

Fig. 13 Encryption time vs. File size

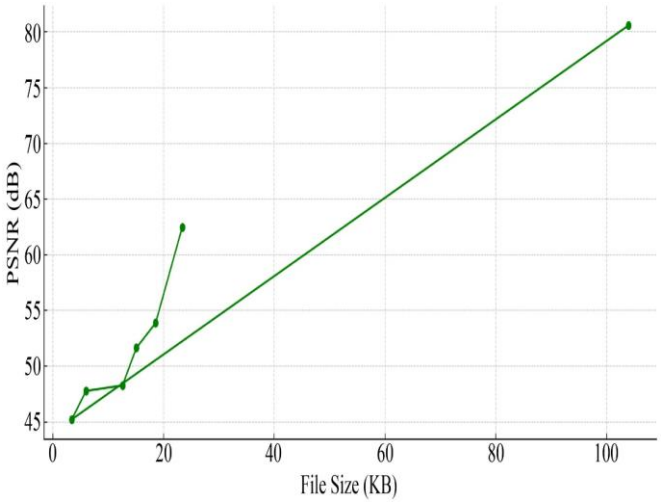

Fig. 14 PSNR vs. File size for image steganography

Figure 14 shows a graph of PSNR versus file size. It was observed that the PSNR had high values (above 40 dB). This confirmed the high quality of the images across different file sizes. The encryption was completed in under 2 seconds, even for 7.5 MB files. Also, decryption time was faster than encryption due to reduced processing overhead. The quality of the image was preserved after steganographic embedding.

4.2. Comparative Analysis

A comparative study was conducted between the proposed three-layer system and a standard AES+LSB implementation. The same dataset (customerdata.txt) and image (Image1.png from Table 5) were used for consistency.

Table 5. Comparative analysis of security systems

Metric	Two-Layer System (AES+LSB)	Proposed Three-Layer System	Enhancement/Trade-off
Total Encryption Time	0.28 seconds	0.33 seconds	+17.8% (Slightly slower due to extra layer)
Final Stego-Image Size	1908 KB	1910 KB	+0.1% (Negligible size increase)
PSNR (dB)	78.51	78.23	-0.35% (Negligible quality difference)
Avalanche Effect	~49.8%	~50.3%	Improved randomness due to custom cipher
Steganalysis Resilience	Moderate	High	Double encryption creates a ciphertext with high entropy, resisting detection of statistical patterns.

The three-layer approach introduced a slight increase in the processing time (a 17.8% increase in encryption time), but it was compensated for with a measurable improvement in the randomness of the ciphertext (avalanche effect). This enhanced resistance to statistical steganalysis. The proposed three-layer security system provided a substantial enhancement in data protection with only a minor impact on performance. The proposed system achieved stronger results because the custom cipher acted as a "scrambler," eliminating statistical patterns in the plaintext. Although the system was developed with the financial sector in mind, its versatility makes it suitable for other high-sensitivity domains.

5. Conclusion

In this research, a robust multi-layered security system that combined cryptographic and steganographic techniques to protect sensitive data is presented. The proposed system integrated a custom substitution cipher, AES-128 encryption, and dual-mode steganography (whitespace and LSB) to achieve defense-in-depth.

The encryption time for the custom encryption algorithm combined with the AES module was recorded in less than a second. Also, the encryption time for the text steganography module (Whitespace) was less than 2 seconds. Finally, the

maximum encryption time recorded for the image steganography module (LSB) was less than 1 second for the files used. High PSNR values confirmed that the quality of images was preserved.

This system will serve as a resource for training institutions and organizations that handle sensitive data. LSB and whitespace steganography techniques have limited payload capacity, making them unsuitable for embedding large files. Current implementation relies on pre-shared secret keys, which introduces challenges in key distribution and storage. Continued innovation in this domain is essential in developing more robust security frameworks and maintaining trust in an increasingly digital world.

5.1. Future Work

Future research can be extended to adaptive steganography, where an intelligent module could be developed to analyze both the payload and the available cover media to automatically choose the most suitable steganographic technique, such as using LSB when capacity is the priority, or other methods when higher resilience is needed. Another direction is the integration of machine learning algorithms for threat detection, allowing a model to be trained to recognize patterns that indicate a steganalysis attack in real time. Exploring alternative ciphers, such as ChaCha20, also presents an opportunity to compare various cryptographic algorithms in terms of performance and security.

References

[1] Wenli Duo, MengChu Zhou, and Abdullah Abusorrah, "A Survey of Cyber Attacks on Cyber-Physical Systems: Recent Advances and Challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, No. 5, pp. 784-800, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Chowdhury A., "Applications and Techniques in Information Security," *6th International Conference on Applications and Techniques in Information Security*, Cairns, pp. 54-65, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[3] Pan Yang, Naixue Xiong, and Jingli Ren, "Data Security and Privacy Protection for Cloud Storage: A Survey," *Special Section on Emerging Approaches to Cyber Security*, vol. 8, pp. 131723- 131740, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Danielle Kriz, "Cybersecurity Principles for Industry and Government: A Useful Framework for Efforts Globally to Improve Cybersecurity," *2011 Second Worldwide Cybersecurity Summit (WCS)*, London, UK, pp. 1-3, 2011. [Google Scholar] [Publisher Link]

[5] Richard Boateng et al., "Cyber Crime and Criminality in Ghana: Its Forms and Implications," *16th Americas Conference on Information Systems*, Lima, Peru, pp. 85-100, 2011. [Publisher Link]

[6] Ritz Carr, "Some Legal and Practical Challenges in the Investigation of Cybercrime," *Cybersecurity Undergraduate Research*, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Grace Odette Boussi, Himanshu Gupta, and Syed Akhter Hossain, "Financial Crime: A Conceptual Framework Implementation for Prevention of Malicious Request from a VPN or Proxy Server," *Scope Journal*, vol. 13, no. 1, pp 375-385, 2023. [Publisher Link]

[8] Md. Khalid Imam Rahmani, Mr. Amit Kumar Goyal, and Manisha Mudgal, "Study of Cryptography and Steganography System," *International Journal of Engineering and Computer Science*, vol. 4, no. 8, pp. 13685-13687, 2015. [Google Scholar] [Publisher Link]

[9] Kaustubh Dwivedi et al., "A Novel Classification of Attacks on Blockchain Layers: Vulnerabilities, Attacks, Mitigations, and Research Directions," *arXiv Preprint*, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[10] Richard Apau et al., "Image Steganography Techniques for Resisting Statistical Steganalysis Attacks: A Systematic Literature Review," *PloS one*, vol. 19, no. 9, pp. 1-47, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Hayfaa Abdulzahra Atee, Robiah Ahmad, and Norliza Mohd Noor, "Cryptography and Image Steganography using Dynamic Encryption on LSB and Color Image Based Data Hiding," *Middle-East Journal of Scientific Research*, vol. 23, no. 7, pp. 1450-1460, 2015. [Google Scholar]

[12] Mangesh Kulkarni, Prasad Jagtap, and Ketan Kulkarni, "An Efficient Data Hiding Scheme using Steganography and Cryptography Technique," *International Journal of Scientific and Research Publications*, vol. 5, no. 4, pp. 1-4, 2015. [Google Scholar] [Publisher Link]

[13] Abrar Alsaidi et al., "Compression Multi-Level Crypto Stego Security of Texts Utilizing Colored Email Forwarding," *Journal of Computer Science & Computational Mathematics (JCSCM)*, vol. 8, no. 3, pp. 33-42, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[14] Malak Gorm-Allah Alkhudaydi, and Adnan Abdul-Aziz Gutub, "Integrating Light-Weight Cryptography with Diacritics Arabic Text Steganography Improved for Practical Security Applications," *Journal of Information Security and Cybercrimes Research*, vol. 3, no. 1, pp.13-30, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Nouf A. Al-Juaid, Adnan A. Gutub, and Esam A. Khan, "Enhancing PC Data Security via Combining RSA Cryptography and Video-Based Steganography," *Journal of Information Security and Cybercrimes Research*, vol. 1, no. 1, pp. 5-13, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[16] G. Diwakara Reddy et al. "A Proficient and Secure Way of Transmission using Cryptography and Steganography," *2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, Tashkent, Uzbekistan, pp. 582-586, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] N.R.D.P. Astuti, E. Aribowo, and E. Saputra, "Data Security Improvements on Cloud Computing using Cryptography and Steganography," *IOP Conference Series: Materials Science and Engineering*, vol. 821, no. 1, pp. 1-8, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] Eshraq S. Bin Hureib, and Adnan A. Gutub, "Enhancing Medical Data Security via Combining Elliptic Curve Cryptography with 1-LSB and 2-LSB Image Steganography," *IJCSNS International Journal of Computer Science and Network Security*, vol. 20, no. 12, pp. 232-241, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Fredy Varghese, and P. Sasikala, "A Detailed Review based on Secure Data Transmission using Cryptography and Steganography," *Wireless Personal Communications*, vol. 129, no. 4, pp. 2291-2318, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Mustafa S. Abbas, Suadad S. Mahdi, and Shahad A. Hussien, "Security Improvement of Cloud Data using Hybrid Cryptography and Steganography," *2020 International Conference on Computer Science and Software Engineering (CSASE)*, Duhok, Iraq, pp. 123-127, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[21] Liliana Pasquale, "Automating Trade-Off Analysis of Security Requirements," *Requirements Engineering*, vol. 21, no. 4, pp. 481-504, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[22] Ahmed Ali AL-Shaaby, and Talal AlKharobi, "Cryptography and Steganography: New Approach," *Transactions on Networks and Communications*, vol. 5, no. 6, pp. 25-38, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[23] Swati S Patil, and Prof. Sangeetha Goud et al., "Enhanced Multi-Level Secret Data Hiding" *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 2, no. 2, pp. 846-850, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[24] Ramadhan J. Mstafa, and Khaled M. Elleithy, "A Highly Secure Video Steganography using Hamming Code (7, 4)," *IEEE Long Island Systems, Applications and Technology (LISAT) Conference 2014*, Farmingdale, NY, USA, pp. 1-7, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[25] B. Padmavathi, and S. K. Ranjitha, "A Survey on Performance Analysis of DES, AES and RSA Algorithm along with LSB Substitution Technique," *International Journal of Science and Research*, vol. 2, no. 4, pp. 170-174, 2013. [Google Scholar]

[26] P. Manikandaprabhu, and M. Samreetha, "A Review of Encryption and Decryption of Text using the AES Algorithm," *International Journal of Scientific Research & Engineering Trends*, vol. 10, no. 2, pp. 400-404, 2024. [Google Scholar] [Publisher Link]

[27] R. Lavanya, and M. Karpagam, "Enhancing the Security of AES through Small-Scale Confusion Operations for Data Communication," *Microprocessors and Microsystems*, vol. 75, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[28] Paweł Chodowiec, and Kris Gaj "Very Compact FPGA Implementation of the AES Algorithm," *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 319-333, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[29] Rajesh Kumar Tiwari, and G. Sahoo, "A Novel Methodology for Data Hiding in PDF Files," *Information Security Journal: A Global Perspective*, vol. 20, no. 1, pp. 45-57, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[30] Shabir A. Parah et al., "Hiding in Encrypted Images: A Three-Tier Security Data Hiding Technique," *Multidimensional Systems and Signal Processing*, vol. 28, no. 2, 549-572, 2017. [CrossRef] [Google Scholar] [Publisher Link]