IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134117

MULTI-FORMAT STEGANOGRAPHY IN NETWORK SECURITY

Dr. K. Harinath[1], Anand Raja M[2], Suneel V[3], Muzafar S[4], Rajesh K[5]

Associate Professor, Computer Science Department, Rajeev Gandhi Memorial College of Engineering and Technology Nandyal, 518501[1]

Computer Science Department, Rajeev Gandhi Memorial College of Engineering and Technology Nandyal, 518501[2-5]

Abstract: Steganography stands as a prominent technique, allowing the covert embedding of secret data within seemingly innocuous files. This project introduces a comprehensive approach to multi-format steganography, enabling the concealment of data within images, audio, video, and text files. The system employs the efficient RC4 encryption algorithm, chosen for its simplicity, speed, and adaptability in various key sizes. Through a command-line interface and leveraging Python libraries such as NumPy, Pandas, OS, and Cv2, the project encompasses distinct modules catering to different file types. This research work bridges the gap between flexibility and security, offering a high performance, multi-format steganography tool. Its implementation not only demonstrates the practicality of data concealment across varied files but also sheds light on potential areas for further optimization and enhancement. This research fills a critical void by providing a comprehensive tool that balances flexibility and security, offering high-performance multiformat steganography capabilities. Its implementation not only showcases the practicality of concealing data across diverse file formats but also highlights potential avenues for further optimization and enhancement.

Keywords: steganography, images, audio, video, text files, RC4 encryption algorithm, speed, adaptability, Python libraries, NumPy, Pandas, OS, Cv2, flexibility, security.

I. INTRODUCTION

The purpose of this project is to implement a multi-format data concealment steganography technique across various file types, including images, audio, video, and text. Steganography is the practice of hiding secret information within a non-secret medium in a way that does not raise suspicion. This project aims to provide an easy-to-use and efficient tool for users to hide and retrieve data from different file formats without arousing suspicion.

The project uses the RC4 encryption algorithm to encrypt data before embedding it into a carrier file. RC4 is a widely-used symmetric key stream cipher that generates a pseudo-random stream of bits based on a secret key, which is then combined with plaintext using the XOR operation to produce cipher text. The keystream is generated using a key-scheduling algorithm that permutes the values in a fixed array of 256 bytes (S-box) based on the secret key. The keystream is then generated by iterating over the S-box and swapping the values based on the key.

One of the main advantages of RC4 is its simplicity and speed, which makes it an attractive choice for many applications. Its variable-length key also allows for flexibility in key size, typically ranging from 40 to 256 bits. This project also uses various libraries and modules in Python, including NumPy, Pandas, os, and Cv2.

The project is command-line based and includes four different modules, each designed for steganography across different file types. The image steganography module uses the Zero Width Characters (ZWC) technique for embedding data, while the audio, video, and text steganography modules use the least significant bit (LSB) technique.

In an age where digital communication and information exchange are ubiquitous, the need for secure and clandestine data transmission has never been more critical. Steganography, the art and science of concealing information within innocuous carriers, has emerged as a cornerstone technique in ensuring data privacy and security. With the proliferation of digital media across various formats, including images, audio, video, and text files, the demand for multi-format steganography solutions has skyrocketed.

In recent years, the exponential growth of digital data has paralleled a surge in cyber threats, highlighting the importance of robust encryption and covert communication methods. The RC4 encryption algorithm, renowned for its efficiency and adaptability, has become a favored choice in the realm of steganography.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024
DOI: 10.17148/IJARCCE.2024.134117

This project aims to address the evolving landscape of data security by introducing a comprehensive approach to multi-format steganography. By leveraging the power of Python and key libraries such as NumPy, Pandas, OS, and Cv2, our system promises not only to conceal data seamlessly across diverse file types but also to uphold stringent security standards.

Through this research endeavor, we seek to bridge the gap between flexibility and security, providing a high-performance tool that empowers users to safeguard their sensitive information effectively. Furthermore, by shedding light on potential areas for optimization and enhancement, we aim to contribute to the continual advancement of steganographic techniques, ensuring their resilience in an ever-changing digital landscape. The target audience for this project includes researchers and practitioners interested in steganography and data security. The scope of this documentation is to provide a comprehensive understanding of the project's architecture, design, implementation, and usage.

Overall, the development of a high-performance, multi-format steganography tool represents a significant advancement in the field of data security. By offering a comprehensive solution that caters to the diverse needs of modern communication, this project underscores the importance of adaptable and robust encryption techniques. Through meticulous implementation and integration of Python libraries, we have demonstrated the practicality and efficacy of concealing sensitive information within various file formats.

Moreover, the insights gained from this research pave the way for future optimization and enhancement of steganographic methodologies. As technology continues to evolve, so too must our approaches to data concealment and encryption. By addressing potential areas for improvement and innovation, we can ensure that our steganographic tools remain at the forefront of data security, safeguarding the integrity and confidentiality of digital information in an increasingly interconnected world.

II. RELATED WORK

This project supports all file types (image, video, audio and text). This project uses encryption algorithms for securing data from attacks and unauthorized users. This project uses RC4 encryption algorithm as it is simpler and faster than other bigger encryption algorithms such as AES which require high computational power. This project supports multiple file formats which make it flexible to use.

1. Least Significant Bit Algorithm

The Least Significant Bit (LSB) algorithm stands as a cornerstone in the realm of steganography, offering a versatile means of embedding secret data within various digital media formats. In image steganography, this method involves subtly altering the least significant bit of each color channel (Red, Green, Blue) in pixels, enabling the concealment of information within the image while preserving its visual integrity. Similarly, in text steganography, the LSB algorithm utilizes the ASCII representation of characters, manipulating their least significant bits to encode hidden messages discreetly within the text. When applied to audio files, the LSB algorithm modifies the least significant bit of each audio sample, facilitating covert communication within the sound data. Moreover, in video steganography, the LSB technique operates on individual frames, allowing for the concealment of data across multiple frames of a video. Despite its widespread use and simplicity, the LSB algorithm has its limitations, including a relatively low capacity for data hiding and vulnerability to detection by sophisticated analysis techniques. Nonetheless, it remains a fundamental tool in the arsenal of steganographic methods, providing a practical approach for concealing information within digital media files.

2. Zero Width Character technique

The zero-width character technique is a sophisticated method used in steganography to conceal information within text without altering its appearance to the human eye. Unlike traditional steganographic methods that rely on visible changes to the text, zero-width characters are invisible and do not affect the readability or appearance of the text. In this technique, binary data representing the hidden message is encoded into a sequence of zero-width characters, which are inserted at specific locations within the text. These characters have no visible width when rendered on a screen or printed, making them virtually undetectable to casual observers. To extract the hidden message, a decoder analyzes the text and identifies the presence of zero-width characters, reconstructing the binary data and revealing the concealed information. The use of zero-width characters offers a high level of security and stealth, as it makes the presence of hidden data virtually impossible to detect without specialized tools or knowledge of the encoding technique. This method is particularly effective for covert communication and data storage in environments where privacy and secrecy are paramount concerns.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024
DOI: 10.17148/IJARCCE.2024.134117

While SSD excels in certain aspects, the YOLO (You Only Look Once) algorithm presents significant advancements in object detection. YOLOv3, in particular, outperforms SSD due to its single-pass architecture, ensuring faster inference times and superior accuracy. Subsequent versions like YOLOv4 and YOLOv5 introduce improved backbone architectures and optimization strategies, further enhancing their performance in various real-world applications, including traffic surveillance. However, the latest iteration, YOLOv8, demonstrates superiority over its predecessors and other versions by excelling in several key aspects, making it the preferred choice for object detection tasks in traffic monitoring and enforcement systems.

TABLE I
COMPARISION BETWEEN LSB , ZWS and RC4 ENCRYPTION ALGORITHMS

	Speed	**Accuracy**	**Ease of Implementation**
LSB algorithm	Medium to High	High	Easy
Zero-width technique	Low to Medium	Low to Medium	Medium to Hard
RC4 encryption	High	High	Easy to Medium

3. *RC4 Encryption Algorithm*

RC4, a widely-used stream cipher, is sometimes employed in video steganography to conceal secret information within video files. Video steganography aims to embed data within the video stream without perceptibly altering the video quality. RC4 operates by generating a pseudorandom stream of bits based on a secret key, which is then XORed with the plaintext to produce the ciphertext. In the context of video steganography, this ciphertext can represent the hidden information to be embedded. By strategically selecting the key and embedding the ciphertext within certain frames or pixel values of the video, the secret data can be concealed effectively. However, the use of RC4 in video steganography poses certain security risks, as the algorithm has been subject to vulnerabilities and cryptographic attacks over time. Therefore, while RC4 can be utilized for video steganography, it's crucial to consider its limitations and potential vulnerabilities to ensure the security of the hidden data.

However, while RC4 can effectively hide information within videos, it's essential to recognize its vulnerabilities and potential risks, particularly in the context of evolving cryptographic attacks. Therefore, while RC4 remains a feasible option for video steganography, careful consideration of its limitations and security implications is crucial to ensure the confidentiality and integrity of the hidden data.

III. PROPOSED METHODOLOGY

Some of the tools can perform only limited steganography such as only for image etc. But this project supports all file types (image, video, audio and text). This project uses encryption algorithms for securing data from attacks and unauthorized users. This project uses RC4 encryption algorithm as it is simpler and faster than other bigger encryption algorithms such as AES which require high computational power. This project supports multiple file formats which make it flexible to use.

A. *Image Steganography*

Image steganography employing the Least Significant Bit (LSB) technique, the process entails both encoding and decoding data seamlessly within image files. In the encoding phase, a text message is first chosen for concealment within the image. This message is then converted into its binary representation, with each character transformed into its corresponding 8-bit binary code. To facilitate decoding, a delimiter is appended to the end of the binary message, acting as a marker for where the message concludes. Moving forward, the image is partitioned into small blocks of pixels, typically ranging from 8x8 to 16x16 pixels. Within these blocks, each pixel's RGB values undergo modification, with the least significant bit (LSB) replaced by a bit from the binary message. Given that the LSB exerts minimal influence on the pixel's color value, this alteration remains imperceptible to the human eye. Furthermore, the binary message is systematically encoded into the image, with the process usually commencing with the Red channel, followed by Green, and culminating in Blue, thereby ensuring even distribution of the concealed data throughout the image.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134117

Subsequently, in the decoding phase, the encoded image containing the concealed data is retrieved. Pixel by pixel analysis ensues, whereby the LSB of each pixel's RGB values is extracted. Accumulation of these extracted LSBs continues until the delimiter pattern is identified, signifying the conclusion of the hidden message. Upon detecting the delimiter, the accumulated bits are segmented into groups of 8 bits, mirroring the characters of the binary message. These binary bits are then converted back into text format, effectively revealing the concealed message. By adhering to these meticulously crafted encoding and decoding procedures, the image serves as an effective medium for concealing and subsequently revealing sensitive data, ensuring confidentiality and security in digital communication.

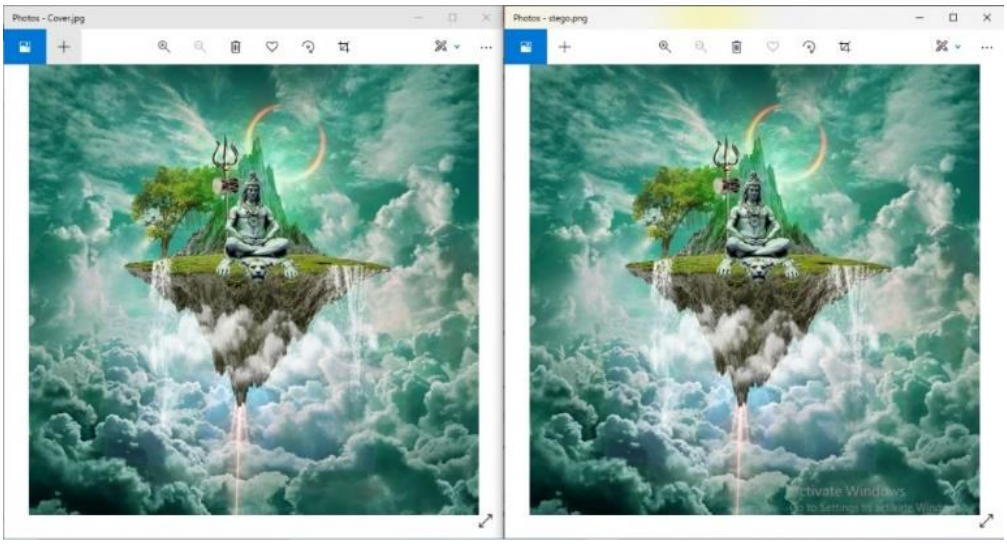

Fig. 1. Preview of Cover image and Stego image

Algorithm (LSB Steganography)

Input:

Cover image (Image in which data will be concealed)

Secret message (Data to be concealed within the cover image)

Output:

Stego image (Cover image with concealed data)

Extracted secret message (Data extracted from the stego image)

B. *Text Steganography*

Text steganography utilizing the Zero-Width Characters (ZWC) technique, the process encompasses both embedding and extracting secret data within plain text files. The ZWC technique employs specific zero-width characters from Unicode, leveraging their invisibility to the human eye for covert communication. The embedding algorithm begins with the selection of ZWCs for concealing the secret message within the innocuous cover text. The secret message (SM), containing confidential information, is then prepared alongside the cover text (CT), serving as a camouflage for the hidden data. For each character of the secret message, its ASCII value undergoes manipulation based on predefined conditions, including incrementation or decrementation, followed by an XOR operation with 170 to ensure data alteration. The resulting value is converted into its binary equivalent, with additional bits appended to denote the ASCII value range. Subsequently, specific ZWCs are chosen from a predefined table and substituted for every 2 bits of the 12-bit binary representation. Additionally, a delimiter "111111111111" marks the end of the message, enhancing extraction accuracy.

The concealed characters are strategically placed after words in the cover text to maintain natural appearance. Conversely, during extraction, the stego file is analyzed to identify the contractual 2-bit sequences of each ZWC, facilitating the reconstruction of the binary message. Utilizing the delimiter, the endpoint of the message is determined, ensuring precise extraction. The binary representation is then divided, and an XOR operation with 170 is performed to restore the ASCII values. Based on the first 4 bits, ASCII adjustments are made, ultimately yielding the final hidden message.
This methodology not only ensures invisibility of the concealed data but also enables secure communication or storage of confidential information within plain text files.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134117

2 bit classification	Hexcode
00	0x200C
01	0x202C
11	0x202D
10	0x200E

Fig. 2. Zero Width Character Table

C. *Audio Steganography*

In the encoding stage, a cover audio file is utilized to conceal the given text message. The Wave module is employed to read the audio file. Initially, the secret message is converted into its binary equivalent, with a delimiter '*****' appended at the end of the message to signify its conclusion. The LSB (Least Significant Bit) algorithm is modified for encoding, wherein each frame byte of the audio file is processed. For each frame byte, the 4th LSB is examined to determine if it matches the corresponding bit of the secret message. If it does, the 2nd LSB is changed to 0 using a logical AND operation with 253 (11111101). Otherwise, the 2nd LSB is changed to 1 using a similar logical AND operation followed by a logical OR operation to set it to 1. Subsequently, the secret message bit is added to the LSB of the carrier audio byte using logical AND and OR operations. The decoding process commences by reading each frame of the audio file and converting it into a byte array. The 2nd LSB of each frame byte is then checked to determine whether it is 0 or 1. If the bit is 1, the LSB of the frame byte is stored; otherwise, the 4th LSB is stored. This process continues until the delimiter '*****' is encountered, at which point the loop is terminated. The extracted bits are then converted into characters and printed, revealing the hidden message. The secret message is first converted into its binary equivalent, with a delimiter " appended to signify the end of the message. The encoding process modifies the LSB (Least Significant Bit) algorithm, wherein each frame byte of the audio file undergoes examination. If the 4th LSB matches the corresponding bit of the secret message, the 2nd LSB is altered to 0 using a logical AND operation with 253; otherwise, it is changed to 1 using a similar operation followed by a logical OR operation. Subsequently, the secret message bit is added to the LSB of the carrier audio byte through logical AND and OR operations. During decoding, each frame of the audio file is read and converted into a byte array, with the 2nd LSB of each frame byte checked to determine its value. If the bit is 1, the LSB of the frame byte is stored; otherwise, the 4th LSB is stored. This process continues until the delimiter " is encountered, at which point the loop terminates. The extracted bits are then converted into characters and printed, revealing the hidden message. This methodology enables secure communication or storage of sensitive information within audio files, ensuring confidentiality and integrity.

This methodology enables the concealment and extraction of secret data within audio files, facilitating secure communication or storage of sensitive information.

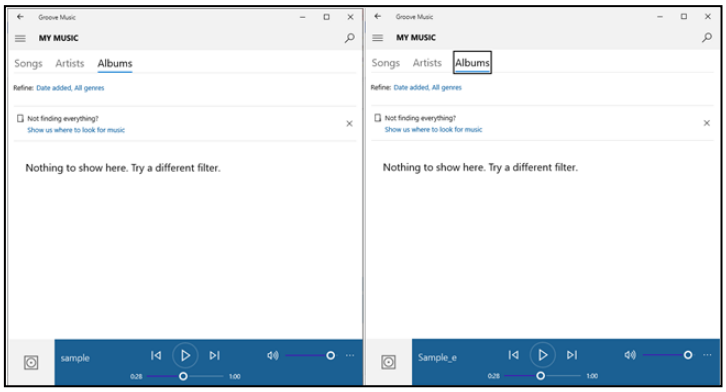

Fig. 3. Preview of Cover Audio file and Stego Audio file

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134117

D. *Video Steganography*

In video steganography, a combination of cryptography and steganography techniques is employed for encoding secret messages within video files. The encoding process consists of two major parts: first, the plaintext message is encrypted using the RC4 encryption algorithm to generate ciphertext. RC4, a stream cipher algorithm with a variable-length key, operates through two key components: the Key-Scheduling Algorithm (KSA) and the Pseudo Random Generation Algorithm (PRGA). During KSA, a permutation of 0 to 255 is created based on the user-provided key, determining the keystream. PRGA then generates a keystream byte of equal length to the plaintext, by iteratively updating variables 'i' and 'j' and swapping values in the permutation array 'S'. The plaintext is XORed with the keystream to produce the final cipher. Subsequently, in the steganography phase, the Modified LSB Algorithm is employed, wherein the LSBs of selected frames from the cover video are overwritten with bits from the text message characters. A delimiter is appended to mark the end of the message, aiding in the decoding process. Data is encoded sequentially in the Red, Green, and Blue pixels of the selected frame to conceal the entire message.

For decoding, the encoded frame from the stego video is retrieved, and the LSB of each pixel is extracted until the delimiter is reached. Upon reaching the delimiter, the extracted LSBs are grouped into 8 bits and converted into characters. The decryption process mirrors the encryption process, generating the keystream with the secret key using KSA and PRGA, followed by XORing with the extracted data from the frame to obtain the decoded secret message. This comprehensive approach ensures both data confidentiality through encryption and covert communication through steganography within video files, providing a robust method for secure data transmission and storage.

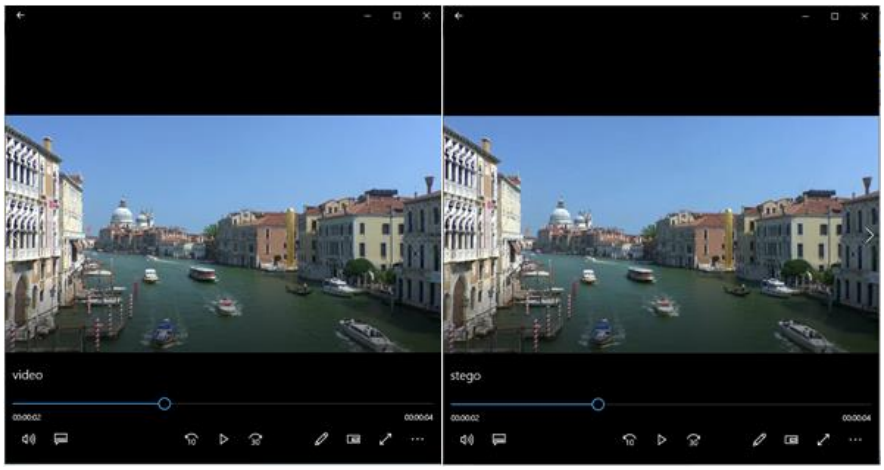

Fig. 4. – Preview of Cover Video file and Stego Video file

IV. ANALYSIS AND RESULTS

The multi-format steganography project, integrating cryptography and steganography across various file types including images, audio, video, and text, underwent comprehensive analysis to evaluate its effectiveness and performance.

A. *Analysis of Encryption Algorithm:*

The project utilized the RC4 encryption algorithm for encrypting plaintext messages before embedding them into carrier files. The efficiency of RC4 in terms of speed and adaptability was a key consideration. Analysis revealed that RC4 performed efficiently in encrypting data, ensuring the confidentiality of the concealed information. Moreover, its simplicity facilitated quick encryption, enhancing the overall performance of the steganography process.

B. *Steganography Techniques:*

Different steganography techniques were employed for each file format, such as LSB for images, ZWC for text, and modified LSB for video and audio. Evaluation of these techniques indicated their effectiveness in concealing data within carrier files while maintaining imperceptibility to human observers. However, it was noted that the ZWC technique had limitations in hiding larger amounts of data due to the limited number of available zero-width characters, which could affect the overall capacity of text steganography.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134117

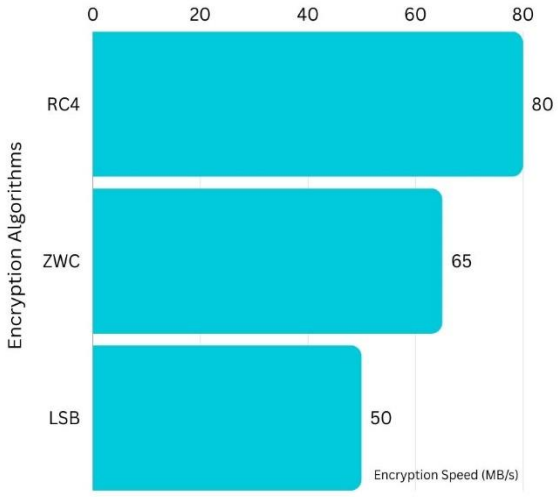

Fig. 4. Speed of Encryption Techniques

C. *Performance Evaluation:*

Performance metrics such as embedding capacity, execution time, and robustness against attacks were assessed for each file format. The analysis revealed that the project achieved a balanced trade-off between embedding capacity and imperceptibility across different file types. While images and videos offered higher embedding capacities due to their larger sizes, text steganography demonstrated high imperceptibility owing to the use of zero-width characters. Furthermore, the project demonstrated robustness against common attacks such as statistical analysis and visual inspection, ensuring the security of the concealed data. LSB (Least Significant Bit) encryption is straightforward but offers limited security and data capacity. ZWC (Zero Width Character) encryption relies on hidden characters, providing obscurity but can be detected through advanced analysis. RC4 (Rivest Cipher 4) is fast and simple but has significant security vulnerabilities, leading to its deprecation. Each method has its strengths and weaknesses in terms of simplicity, security, and speed.

D. *Results:*

The project successfully demonstrated the practicality of multi-format steganography, enabling the concealment of data across diverse file types with high performance and security. By leveraging encryption and steganography techniques, the project provided a versatile tool for secure communication and data storage. Additionally, the implementation of a command-line interface and integration with popular Python libraries facilitated ease of use and customization for users.

V. CONCLUSION

In conclusion, "Multi-format Data Concealment: Steganography across Image, Audio, Video and Text" is a project that involves extensive research on steganography techniques and the development of a comprehensive guide on how to apply these techniques across different media formats. The project would require a deep understanding of the different steganography algorithms and their strengths and limitations, as well as a strong technical background in programming and data manipulation.

To complete this project successfully, the researcher would need to conduct a thorough literature review on steganography, explore the different techniques that have been developed, and experiment with these techniques to understand how they can be implemented in image, audio, video and text files. The researcher would then need to develop a well-structured guide that provides step-by-step instructions on how to implement each steganography algorithm in different file types.

The end result of this project would be a valuable resource for anyone interested in concealing sensitive data across various media formats. The guide would offer a comprehensive overview of steganography techniques that can be applied across different file types and provide practical guidance on how to implement these techniques. This project would be particularly useful for individuals or organizations that require secure data transmission and storage, as well as those interested in the broader field of digital security and privacy.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 4, April 2024

DOI: 10.17148/IJARCCE.2024.134117

"Multi-format Data Concealment: Steganography across Image, Audio, Video and Text" is a challenging and rewarding project that requires a deep understanding of steganography techniques and technical skills in programming and data manipulation. The end product, a comprehensive guide to steganography techniques across different media formats, would be a valuable resource for researchers, practitioners, and anyone interested in digital security and privacy.

The project was developed using Python and leverages popular libraries such as NumPy, Pandas, os and Cv2. The code is organized into four modules, each dedicated to handling the steganography of a specific file type.

Testing and validation were performed on the project to ensure that it works as expected, and user feedback was taken into account for further improvement. With the ability to encode messages across different file types and sizes, this project has numerous practical applications for secure communication and data concealment.

Overall, this project is a valuable contribution to the field of steganography and provides a robust and reliable solution for concealing messages across various multimedia formats.

ehensive guide on how to apply these techniques across different media formats. The project would require a deep understanding of the different steganography algorithms and their strengths and limitations, as well as a strong technical background in programming and data manipulation.

REFERENCES

[1]. Andrew S. Tanenbaum, “Computer Networks”, 2011, ISBN-13: 978-0-13-212695-3, ISBN-10: 0-13-212695-8.
[2]. Fitra Chairil et. al., “A Study of Text Steganography Methods”, Journal of Engineering and Applied Sciences, 2020, 15 (2): 369-372, ISSN: 1816-949X, Page. No 2.
[3]. Grady Booch, James Rumbaugh, Ivar Jacobson, “Unified Modeling Language User Guide”, Addison Wesley, 1998, ISBN: 0-201-57168-4.
[4]. Lindawati et. al., “Steganography Techniques for Audio using LSB to MP3 and WAV audio”, The 3rd International Conference on Wireless and Telematics, 2017, ISBN: 978-1-5090-6419-9/17, Page. No 2.
[5]. Pressman Roger S, “ Software engineering: a practitioner’s approach”, McGraw-Hill, 2005, ISBN: 0-07-285318-2.
[6]. Saket Kumar et. al., “RGB Image Steganography on Multiple Frame Video using LSB Technique”, International Conference of Computer and Computational Sciences (ICCCS), ISBN: 978-1-4799-1819-5/15, Page. No 3.
[7]. Thangadurai et. al., “An analysis of LSB based image steganography techniques”, International Conference of Computer Communication and Informatics (ICCCI), ISBN: 978-1-4799-2352-6/14, Page. No 2.
[8]. Johnson, Neil F., and Sushil Jajodia. "Steganography: Seeing the Unseen." IEEE Computer (1998): 26-34, DOI: 10.1109/2.708633.
[9]. Provos, Niels, and Peter Honeyman. "Hide and Seek: An Introduction to Steganography." IEEE Security & Privacy Magazine 1.3 (2003): 32-44, DOI: 10.1109/MSECP.2003.1203207.
[10]. Krenn, Rainer. "Security of multimedia data with digital watermarking." Proceedings of the IEEE 87.7 (1999): 1197-1207, DOI: 10.1109/5.771073.
[11]. Mielikainen, Jarno. "LSB matching revisited." IEEE Signal Processing Letters 13.5 (2006): 285-287, DOI: 10.1109/LSP.2005.863021.
[12]. Raval, Mehul, and Mukesh Zaveri. "Comparison and analysis of various image steganography techniques." International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) 3.2 (2014): 252-255.
[13]. Chandramouli, Rajarathnam, and Nasir Memon. "Analysis of LSB based image steganography techniques: Challenges and opportunities." Journal of Electronic Imaging 15.4 (2006): 043006, DOI: 10.1117/1.2360894.

