Metadata-Based Video Steganography: Development of a New Model for Secure Information Embedding

Dedi Darwis
Faculty of Engineering and Computer Science, Universitas Teknokrat Indonesia, Indonesia
darwisdedi@teknokrat.ac.id (corresponding author)

Yusra Fernando
Faculty of Engineering and Computer Science, Universitas Teknokrat Indonesia, Indonesia
yusra.fernando@teknokrat.ac.id

Abhishek R. Mehta
Faculty of IT and Computer Science, Parul University, India
abhishek.mehta7067@paruluniversity.ac.in

Wamiliana
Department of Mathematics, Universitas Lampung, Indonesia
wamiliana.1963@fmipa.unila.ac.id

Setiawansyah
Faculty of Engineering and Computer Science, Universitas Teknokrat Indonesia, Indonesia
setiawansyah@teknokrat.ac.id

Received: 5 May 2025 | Revised: 3 July 2025 and 10 July 2025 | Accepted: 13 July 2025

 DOI: https://doi.org/10.48084/etasr.11937

ABSTRACT

This study introduces a novel video steganography model called Video Steganography Technique in Metadata (VSTM), which embeds secret messages within the metadata of MP4 video files. By utilizing the 'comments' field in the metadata, the method ensures that the visual and audio quality of the video remains unaffected. This model offers original contributions in maintaining the integrity and resilience of digital data against common manipulations, such as cropping, rotation, and social media compression. The VSTM model combines the Advanced Encryption Standard (AES) for encryption and ZLIB for data compression, enhancing the security and optimizing the data size. The tests demonstrate that the VSTM model maintains perfect video fidelity with no detectable pixel changes and achieves a high level of robustness against various manipulations, including video cropping, rotation, resizing, and sharing through most social media platforms. The test results showed that the VSTM was able to maintain the integrity of the video files, had resistance to visible detection, and was effectively used for small-scale confidential communications. This model offers a practical and secure solution in the field of digital steganography, and has the potential to be applied to a wide range of data protection needs in sensitive digital environments. However, metadata stripping by certain platforms, such as Instagram, affects the message retrieval. The method proves effective and reliable in securing digital information within the video files while preserving quality and ensuring the message integrity.

Keywords-Advanced Encryption Standard (AES); MP4; video steganography; Video Steganography Technique in Metadata (VSTM); ZLIB

I. INTRODUCTION

Digital services, whether web-based or mobile applications, need to be developed with robust security measures to ensure that the users feel confident about the confidentiality of their data and information [1, 2]. While the transmission and reception of data over the internet offer numerous advantages, there are also significant risks involved, such as cybercrimes

including eavesdropping, data alteration, and various forms of data manipulation [3–5]. As of 2023, it is estimated that there are approximately 5.3 billion internet users worldwide, representing about 66% of the global population [6, 7]. In the growing digital era, the data security and confidentiality are very crucial aspects. Personal information, company data, and digital transactions are now stored in systems that are vulnerable to cyber threats, such as hacking, identity theft, and data misuse. A breach of data security can not only harm individuals and organizations financially, but it can also damage reputations and lower the public trust. Therefore, the application of advanced cybersecurity technology, strict privacy policies, and user awareness of the importance of protecting personal data must be a top priority in facing the challenges of an increasingly complex digital world.

This technique allows sensitive information to be conveyed secretly without attracting the attention of unauthorized parties [8]. In the context of information security, steganography is often used as an additional layer of protection to strengthen the confidential communication systems and protect the data from eavesdropping or manipulation [9, 10]. Steganography is a well-established technique in the field of information security that involves concealing information within a medium in such a way that only authorized parties can detect its presence [8, 9]. This method differs from cryptography, where messages are encrypted but remain visible to anyone [11, 12]. In steganography, the existence of the message itself is not easily identifiable, as it is hidden within various media forms, such as images, text, audio, or video [13]. In today's digital landscape, steganography plays an increasingly vital role in safeguarding privacy and data security, particularly in light of rising threats, such as data theft and unauthorized surveillance [14].

Video has emerged as a preferred medium for steganography due to its substantial storage capacity and the complexity of the data it contains [15]. Videos facilitate the concealment of large amounts of data without compromising the visual quality, making detection by third parties more challenging. Various video steganography techniques have been developed, including Least Significant Bit (LSB), Discrete Cosine Transform (DCT), and numerous transformation-domain-based methods [14-17]. However, these techniques also exhibit certain vulnerabilities, such as susceptibility to detection attacks and limitations in their data-hiding capacity [18].

Conventional steganography methods that modify the main content of a video have some significant drawbacks. One of its main drawbacks is the potential for visual degradation, where changes to the pixels or frames of the video can give rise to visible artifacts, thus triggering suspicion. In addition, these methods are vulnerable to attacks of statistical analysis detection techniques, especially if modifications are made indiscriminately or repeatedly on a particular pattern [23, 24]. Another drawback is the sensitivity to compression or conversion of video formats, which can corrupt or remove hidden information if the data are not inserted with techniques that are resistant to such interference. Therefore, while effective in hiding information, these conventional methods require improvements in terms of durability, undetectability, and quality of the media used. As technology advances, existing video steganography techniques are beginning to reveal limitations, particularly regarding detection by increasingly sophisticated security algorithms [19]. Furthermore, there is a growing demand for greater data-hiding capacity [20]. This situation highlights the need for innovations in video steganography techniques that can address these limitations [21]. A new model that is both more efficient and secure is required to meet these challenges, particularly by leveraging elements within videos that have not been extensively utilized, such as metadata.

Metadata, which consist of descriptive information about video data, such as title, duration, format, and other attributes, present significant potential for use in steganography [20–22]. Unlike the main video data, metadata do not directly affect the visual content, allowing for the concealment of information without degrading the video quality [25]. Additionally, metadata are more challenging to detect using traditional analysis techniques, making them an ideal medium for steganography [26]. Utilizing metadata for steganography provides a more secure solution and enhances the data-hiding capacity compared to conventional methods [27]. Metadata have great potential as a space for inserting information in the context of steganography without compromising the visual or audio quality of a video. Unlike conventional methods that change pixels or frames, the insertion of metadata is done on the header or additional information that does not affect the main content. Metadata, such as time, location, codec, or even custom tags, can be modified to house confidential messages in a more secure and inconspicuous way. In addition to maintaining the media quality, the use of metadata also increases resistance to detection, as the changes are not directly visible in the final video result. However, it should be noted that metadata can often be removed or altered during the distribution or compression process, so this technique needs to be circumvented with an adaptive and manipulation-resistant insertion method [28].

This study aims to develop a new video steganography model that utilizes metadata as a concealment medium. The proposed model is named Video VSTM. VSTM is an innovative model that combines cryptographic techniques, data compression, and steganography. Cryptography is both a science and an art used to protect information by transforming it into an unreadable format for unauthorized parties [26, 27]. The primary goals of cryptography are to ensure the confidentiality, integrity, and authenticity of data. In contrast, data compression refers to the process of reducing the size of a file or dataset by eliminating non-essential information or altering the data representation to require less storage space [28, 29]. The main objective of data compression is to conserve the storage capacity and expedite the data transmission across networks.

The cryptographic technique employed is the AES. AES is a symmetric cryptographic algorithm utilized for the encryption and decryption of data. Established as a standard by the National Institute of Standards and Technology (NIST) in 2001. AES replaced the previously used Data Encryption Standard (DES) [30, 31] and is one of the most widely adopted

cryptographic algorithms. The compression technique utilized in this study is ZLIB, an extensively employed data compression library for both compressing and decompressing data. ZLIB finds applications in various file formats and programs, including PNG image files, PDF documents, and network protocols, like HTTP [32, 33]. The algorithm employed by ZLIB combines two primary compression methods: Deflate and LZ77. The proposed model is designed to address the shortcomings of the existing steganography techniques by enhancing the data-hiding capacity and resistance to detection. The primary contribution of this research lies in the development of a novel approach that can be applied across various data security applications, particularly concerning the privacy protection and the safeguarding of the digital information [37]. The proposed crypto-stego strategy in this work offers an efficient approach to maintaining the data confidentiality through hiding it in video media. This approach can strengthen the system's resilience against attacks, while also being a potential direction for the development of more adaptive and distributed works in secure communication systems.

The video security presented is very promising, showing significant improvements in terms of data protection and resistance to attacks as a result of a complex embedding process [38]. In this context, research on image confidentiality makes an important contribution, particularly in explaining the relationship between the embedding capacity and the level of confidentiality achieved. The study shows that an increase in the embedding capacity does not always correspond to an increase in confidentiality, due to the trade-off that must be managed carefully so that the existence of hidden data (stego) remains difficult to detect [39]. Therefore, a video steganography approach that adopts principles from image secrecy research can produce a more adaptive and efficient multimedia security system in maintaining the privacy and information integrity [40]. The combination of video hiding methodologies and secrecy principles from image steganography opens up great opportunities to create robust and adaptive data protection mechanisms against the current digital security challenges [41, 42].

In the context of the real need for protection of secret data, the use of the AES encryption algorithm in combination with ZLIB compression in video steganography models, such as VSTM is a highly relevant and strategic approach. AES provides a high level of security for the embedded message, ensuring that even if the message location is successfully found, the content remains inaccessible without the decryption key. This is crucial in real-world situations, such as communication between investigative journalists and sources in regions with strict censorship, where information leaks can have fatal consequences. On the other hand, ZLIB compression is necessary to address the space limitations in metadata columns, such as 'comments' in MP4 files. This compression allows for the efficient insertion of longer messages without altering the structure of the video file, while also reducing the potential for detection through metadata analysis. Many previous stego-cryptography methods have failed in this aspect, as they did not take into account the space efficiency or relied solely on concealment without encryption protection. Without compression, the size of the encrypted data becomes too large and difficult to embed in limited metadata, whereas without encryption, the content of the hidden message becomes vulnerable if detected. Therefore, the integration of AES and ZLIB not only enhances the security and efficiency, but also addresses the fundamental weaknesses of previous stego-cryptography approaches, making it a robust solution for the secret communication needs in a digitally insecure environment.

The main purpose of the data compression is to save storage space and speed up data transmission over the network. This model is expected to overcome the weaknesses of the existing steganography techniques, by increasing concealment capacity and resistance to detection. The main contribution of this research is the development of new approaches that can be used in various data security applications, especially in the context of privacy protection and digital information security.

II. MATERIALS AND METHODS

In this study, the developed video steganography model involves embedding secret messages into the metadata of MP4 videos using a Python library known as Mutagen. This process not only inserts the message, but also employs cryptographic techniques to enhance the data security by implementing the AES algorithm. Before embedding, the message to be hidden undergoes a compression process using the ZLIB algorithm to reduce the size of the data being inserted, thereby minimizing any significant change to the video file size.

This method is a follow-up to the development of earlier established methods, including the Center Sequential Technique (CST), Center Embedded Pixel Positioning (CEPP), and Steganography on Image Metadata (SIM), which focused on the protection of stego-images concerning the manipulation robustness. The fundamental difference lies in the fact that while the previous research concentrated on image media, this study's primary focus is on using video files in the MP4 format for steganography. Essentially, the Model of VSTM involves the process of embedding messages into the metadata of a video by utilizing the 'comments' feature in the video file.

In the VSTM model, the information intended for concealment (such as text, images, or other data) is embedded into the metadata fields. For instance, this information can be inserted into attributes like 'title,' 'author,' 'description,' or 'comments.' However, the study specifically focuses on manipulating the 'comments' field, as it is easier to embed messages there. One advantage of the VSTM is that the data are stored outside the video content, ensuring that the visual and audio quality of the video remains intact without degradation typically associated with pixel alterations.

The message embedded in the metadata file can be visually perceived by the human eye, as it can be displayed in the file properties. Therefore, before embedding the file, it needs to be encrypted using the AES cryptographic technique. To address the limitations of the data storage space in the metadata file, the study also combines it with the ZLIB algorithm, which can compress the message by over 50% of its original size.

The first step in this process is to select the message to be embedded, which is in text format saved as *.txt. After choosing the message, the next step is to encrypt it using AES and compress it with the ZLIB algorithm. This compression is crucial as it reduces the size of the message, making it easier to be embed without affecting the overall size of the video.

The VSTM model is developed using Mutagen, a Python library designed for managing metadata in various audio and video formats. In the context of video steganography, Mutagen is utilized to read, write, and edit the metadata of MP4 video files. By using Mutagen, users can easily access the metadata sections and embed secret messages.

A. Message Embedding Process Using the VSTM Model

Steganography is effective in hiding confidential data. The VSTM model minimizes the possibility of detecting the inserted message. This approach is particularly suitable for confidential communication scenarios that prioritize the undetectability and media integrity. Figure 1 illustrates the message insertion flow using the VSTM model.

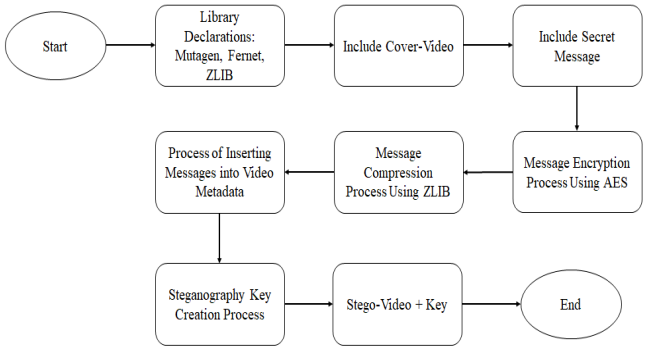


Fig. 1. Message embedding process using the VSTM model.

The initial functioning of the VSTM model relies on various libraries available in Python, including Mutagen, Fernet, and ZLIB. Metadata consist of additional information stored in the video, such as the title, description, and more. Next, Fernet is a cryptographic method that utilizes AES encryption for encrypting and decrypting messages. This ensures that only individuals with the encryption key can read the message. Finally, ZLIB is a method used for data compression. By compressing the message, its size can be reduced, making it more efficient to embed into the video.

The second step is to prepare a cover video in MP4 format. This video serves as the medium for concealing the secret message. It acts as a 'wrapper' that will hide the secret message within its metadata. The selected video should be one that is commonly used to avoid raising suspicion. The third step is to prepare the secret message to be embedded, which can be in the form of text, numbers, or other sensitive information stored in a .txt format. For instance, this message might include confidential company information, access keys, or even personal data that need to be kept private.

The fourth process involves encryption using AES. This encryption standard is known for being secure and difficult to break. The encryption process ensures that the secret message cannot be read directly by anyone attempting to access the metadata without authorization. Through encryption, the message is transformed into a coded format (ciphertext) that cannot be read without the decryption key.

The fifth process is the compression of the message using ZLIB. This compression aims to reduce the size of the encrypted message, making it more efficient to embed within the metadata. Consequently, the metadata do not become excessively large, thereby minimizing the risk of detection.

The sixth process involves embedding the message into the video metadata, which is the core of the developed model. The encrypted and compressed message is inserted into the video's metadata. In this process, the 'comments' field within the video metadata was manipulated. By embedding the message in the metadata, its presence is concealed further, as it does not alter the visual or audio appearance of the video itself.

The final step in message embedding is the creation of a steganography key. This key is the encryption key used to lock (encrypt) and unlock (decrypt) the secret message. It must be stored separately and securely, as only those with this key can read the embedded secret message.

As a result of these processes, a stego-video along with its key is produced. This stego-video appears identical to the cover video in terms of size, duration, and visual and audio quality. However, within its metadata, it holds secret information that can only be accessed with the steganography key that has been created.

B. Message Extraction Process Using VSTM Model

Once the message is successfully embedded in the cover video, it is then sent to the recipient through various media. The recipient extracts the message using the same stego key used during the message insertion process. Figure 2 illustrates the flow of the message extraction from the stego-video.

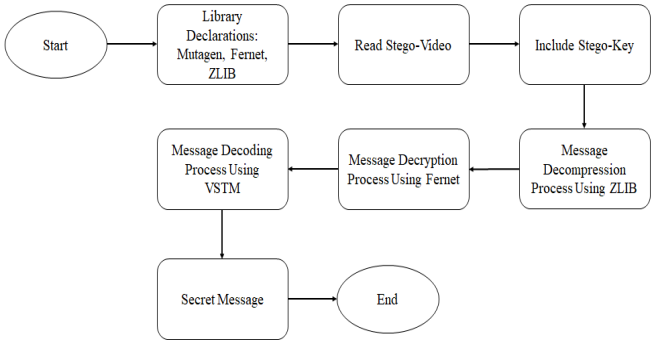


Fig. 2. Message extraction process using the VSTM model.

The message extraction process begins by declaring the Mutagen, Fernet, and ZLIB libraries, like the message embedding process. The next step is to read the stego-video containing the secret message, which is essential for identifying the presence of the secret message within the video to initiate the extraction process.

Following this, the stego-key input by the user is read. In this condition, the algorithm verifies the stego-key; if it is incorrect, the decryption process cannot proceed. If the stego-key is correct, the next step is to perform decompression using the ZLIB algorithm. Decompression is the process of restoring data to their original form before compression. This step is necessary to access the encrypted message in its original size, ensuring that the subsequent decryption process can be executed accurately.

The next stage involves decrypting the message using the Fernet/AES function. This process will revert the message, which was previously transformed into ciphertext (encrypted text), back to its original form (plaintext). Decryption is carried out using the steganography key (stego key) included in the previous steps. If the key used matches the one employed for encryption, the secret message is successfully read and restored to its original state.

The subsequent process is the extraction of the secret message from the results of the prior decompression and decryption. This stage ensures that the decrypted message can be read in the correct format. The VSTM model reads the data structure that has been decrypted and present it in the form of the actual message in its entirety. If this process runs smoothly, the hidden message within the video's metadata is revealed and ready for use. This process also serves to confirm that the message has not been damaged or altered during the embedding and extraction phases.

III. IMPLEMENTATION VSTM

The implementation of VSTM is a practical step in applying safe and inconspicuous steganography techniques to video media. VSTM leverages tools, such as FFmpeg or ExifTool, to efficiently insert and extract hidden information. With this approach, the messages can remain hidden and protected from visible detection, making VSTM an effective solution for securing communications in a digital environment that is prone to eavesdropping or data manipulation.

Based on the flow in Figure 1, the process of inserting messages in video metadata uses a pseudocode:

Message Insertion Pseudocode Using VSTM Model

```
- Import libraries: cryptography (Fernet), zlib,
argparse, mutagen (MP4), and shutil.
FUNCTION hide_message(cover_video_path,
secret_message_path, message_to_hide):
- Generate an encryption key using Fernet.
- Read the contents of the file that needs to be
hidden (message_to_hide).
- Compress the file content using ZLIB.
- Encrypt the compressed data using the generated
key.
- Create a file name based on the
secret_message_path.
- Create and save the encryption key file in the
`key/` directory with the name
`file_key_<filename>.txt`.
- Display the encrypted data and encryption key (for
record-keeping).
- Convert the encrypted data into a string format to
be inserted into the video metadata.
- Copy the cover_video_path to a new file with the
path specified in secret_message_path.
- Open the new video file using Mutagen (MP4).
- Insert the encrypted data string into the `©cmt`
metadata tag of the video.
- Save the video file with the modified metadata.
- Display a confirmation message that the encrypted
data has been successfully inserted.

FUNCTION main():
- Create an argument parser to receive input paths
for `cover_video_path`, `secret_message_path`, and
`message_to_hide`.
- Call the hide_message() function with arguments
obtained from user input.

IF the program is executed as __main__:
- Call the main() function.

END
```

The snippet of pseudocode represents the embedding process using Mutagen metadata in video files. This process differs from the algorithm development previously conducted. In earlier methods, such as CST and CEPP, LSB was primarily utilized for the embedding process. Additionally, the SIM method was used for embedding the metadata in the image. However, the current method does not alter the pixel values at all. Instead, the encrypted and compressed message is directly embedded in one of the video metadata attributes, specifically the 'comments' field. With this developed model approach, the quality of the resulting stego-video remains high in terms of fidelity. The stego-video will also be resistant to manipulation robustness attacks, such as cropping, resizing, rotation, and other types of attacks. According to the principles of good steganography, if the embedded message becomes less noticeable in the host medium or if the pixel value changes minimally or not at all, the quality of the steganography algorithm used will be higher.

The following is the process outlined in pseudocode form in carrying out the message extraction process using the VSTM model.

Message Extraction Pseudocode Using VSTM Model

```
PSEUDOCODE:
- Import libraries: cryptography (Fernet), zlib,
argparse, mutagen (MP4).

FUNCTION unhide_message(stego_video_path, key):
    - Convert the key into bytes format.
    - Create a Fernet cipher object using the
provided key.

    - Set the `output_video` variable to the path of
the stego video (stego_video_path).

    - Use Mutagen (MP4) to read the metadata of
`output_video`.

    - Retrieve the encrypted data from the metadata
tag `©cmt` of the video file.

    IF encrypted data is found in the metadata:
        - Decrypt the data using the Fernet cipher
with the provided key.
        - Decompress the decrypted data using ZLIB.

        - Modify the `output_file` path and filename
for saving the recovered data.
```

```
        - Open the `output_file` and write the
decompressed data to it.
        - Display the decrypted and decompressed
data.
    ELSE:
        - Print a message indicating no encrypted
data was found in the video metadata.

FUNCTION main():
    - Create an argument parser to receive input
arguments for `stego_video_path` and `key`.
    - Call the `unhide_message()` function using the
arguments received from user input.

IF the program is executed as __main__:
    - Call the `main()` function.

END
```

The pseudocode outlines the steps to extract and retrieve the secret message hidden within the metadata of a video (stego-video). A brief explanation of this process involves:

1. Function unhide_message (stego_video_path, key)

- This function takes two parameters: the path of the stego video (stego_video_path) and the encryption key (key).
- The encryption key is converted to byte format and used to create a Fernet object.
- The video metadata is read using Mutagen to retrieve the encrypted data from the metadata tag ©cmt.
- If the encrypted data are found, they are decrypted using the provided encryption key and then decompressed using ZLIB.
- The results of the decryption and decompression are then saved to a new file in the recovery folder.
- If no encrypted data are found, a warning message is displayed.

2. Function unhide_message (stego_video_path, key)

- This function creates an argument parser to receive the input path of the video (stego_video_path) and the encryption key (key) from the user.
- The unhide_message() function is then called with the arguments obtained from the user input.

3. Execution of the Main Program

- If the program is executed as main, the main() function is called to initiate the extraction process.

IV. RESULTS AND DISCUSSION

Before conducting tests on the developed method, several video samples were prepared for trial. A total of five videos in MP4 format were selected as cover videos. The MP4 format was chosen because it contains attributes that can be manipulated to embed messages. Table I presents the tested cover video samples.

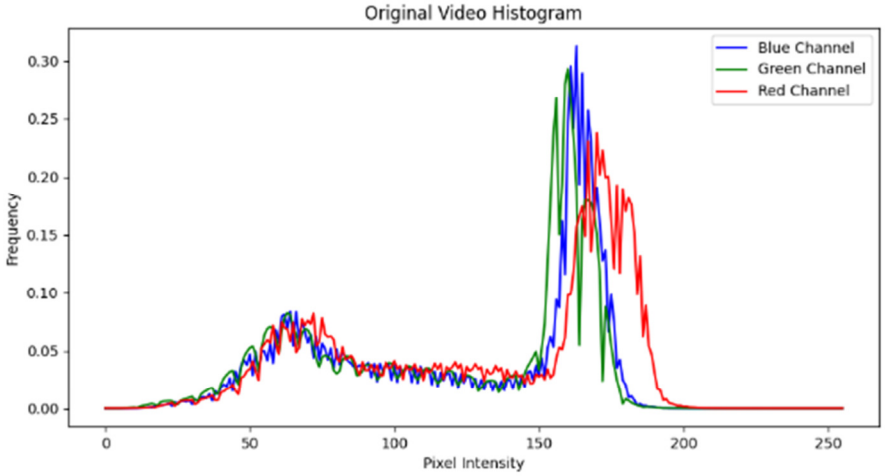
TABLE I. COVER VIDEO SPECIFICATIONS

Cover video	Size (MB)	Dimensions	Duration (s)
elephant.mp4	42.8	1280 × 720	33
giraffe.mp4	57.3	1280 × 720	39
seafood.mp4	17.1	1280 × 720	13
butterflies.mp4	67.1	1280 × 720	51
animals.mp4	3.20	360 ×640	60

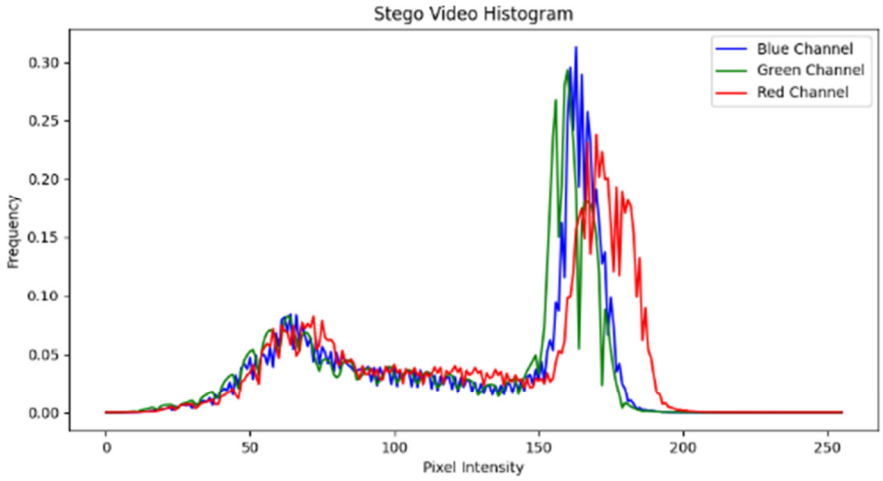
The process involves embedding a secret message in the form of text into the cover video. The first experiment was conducted to observe any changes to the cover video after the message was embedded. The embedded message consists of a string in .txt format with a size of 1,713 bytes. The results of this experiment are presented in Table II.

TABLE II. MESSAGE INSERTION TEST RESULTS

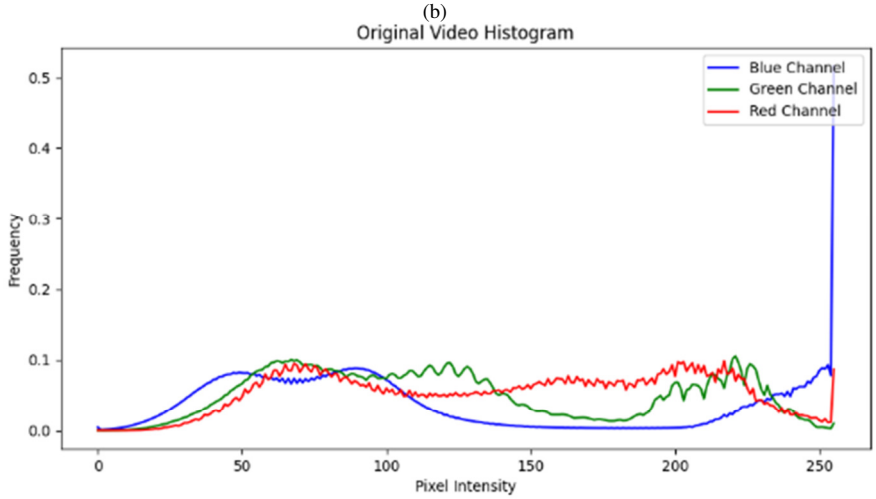
Cover video	Cover video size (MB)	Stego-video size (MB)	Number of pixel changes	Message extraction results
elephant.mp4	42.8	42.8	0	Succeed
giraffe.mp4	57.3	57.3	0	Succeed
seafood.mp4	17.1	17.1	0	Succeed
butterflies.mp4	67.1	67.1	0	Succeed
animals.mp4	3.13	3.13	0	Succeed



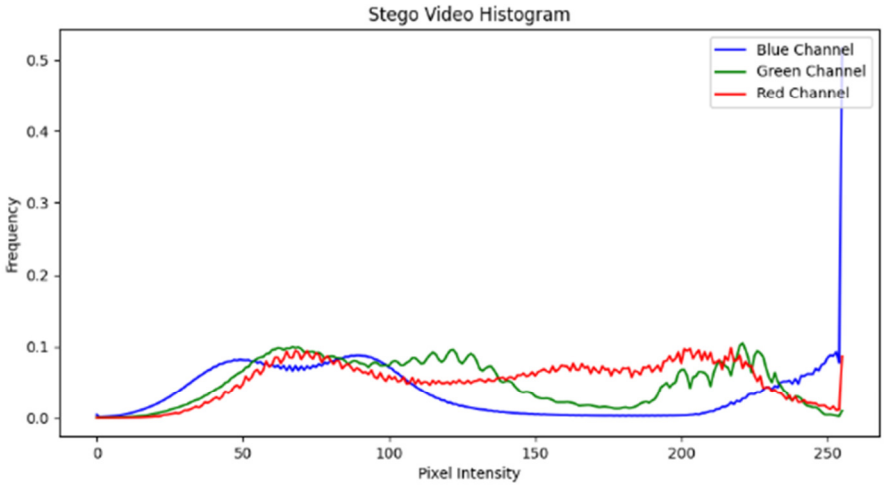
(a)



(b)



(c)



(d)

Fig. 3. Cover video and stego-video: (a) elephant.mp4 (cover video), (b) elephant.mp4 (stego-video), (c) giraffe.mp4 (cover video), (d) giraffe.mp4 (cover video).

The results of the message embedding test presented in Table II indicate that the size of the stego-video remains unchanged. This is due to the absence of any pixel modifications after the message is embedded and the use of the ZLIB compression algorithm to reduce the message size. The extraction results also demonstrate that all embedded messages

can be fully recovered without loss. A visual comparison between the cover video and the stego-video can be observed through the histograms portrayed in Figure 3.

The histograms displayed in Table III show that there is no change in the pixel intensity values for the red, green, and blue channels of the video. This lack of change is because the embedded message does not alter the pixel bit values. Instead, the message is inserted through the video's metadata.

The steganography model developed in this study is more efficient in terms of the size of the embedded secret message compared to SIM. This improvement is due to the addition of the ZLIB compression algorithm. ZLIB is a lossless compression algorithm designed to reduce the data size without any loss of information [35]. It is an implementation of the deflate algorithm, combining two primary compression techniques: LZ77 (Lempel-Ziv 1977) and Huffman coding. The algorithm is known for its simplicity, speed, and compression efficiency, making it a popular choice in various software compression implementations [33, 34]. The results of the secret message embedding tests on the cover video are depicted in Table III.

TABLE III. ZLIB COMPRESSION TEST RESULTS

File name	Original file size (KB)	Compressed file size (KB)	Compression ratio (%)
message_secret1.txt	10	4	60
message_secret2.txt	50	20	60
message_secret3.txt	200	80	60
message_secret4.txt	1,000	400	60

Based on the test results shown in Table IV, the ZLIB algorithm provides stable and efficient compression, achieving approximately 60% compression for text files of varying sizes. This compression efficiency is possible because *.txt files typically contain numerous repeated characters or sequences, allowing the LZ77 and Huffman coding algorithms used by ZLIB to effectively detect and compress them. However, the compression results may vary if the characters in the *.txt file have fewer repeating patterns, resulting in a compressed file size that is not significantly different from the original file size.

A. Fidelity Testing

The fidelity test in video steganography is used to evaluate the extent to which the quality of the stego-video is preserved after embedding the secret message. Fidelity is a critical parameter in measuring the performance of a steganography method, as it indicates the level of distortion introduced to the original video during the embedding process [43]. The metrics used to assess fidelity include calculating the Mean Square Error (MSE) and the Peak Signal-to-Noise Ratio (PSNR) [36, 37]. The MSE value is obtained by:

$$MSE_{AVG} = \frac{MSE_R + MSE_G + MSE_B}{X.Y} \quad (1)$$

where MSE_{AVG} is the cover video's average *MSE* value, MSE_R is the red *MSE* value, MSE_G is the green *MSE* value, MSE_B is the red *MSE* value, and *X, Y* are the video dimensions.

The PSNR value is determined by:

$$PSNR = 10_{log10}\left(\frac{255^2}{MSE}\right) \quad (2)$$

The results of the fidelity test are presented in Table IV.

TABLE IV. FIDELITY TEST RESULTS

Stego-video	MSE	PSNR (dB)
stego_elephant.mp4	0	Inf (∞)
stego_giraffe.mp4	0	Inf (∞)
stego_seafood.mp4	0	Inf (∞)
stego_butterflies.mp4	0	Inf (∞)
stego_animals.mp4	0	Inf (∞)

Based on the fidelity testing results using the MSE and PSNR values, it can be concluded that all the stego-videos tested—stego_elephant.mp4, stego_giraffe.mp4, stego_seafood.mp4, stego_butterflies.mp4, and stego_animals.mp4—showed an MSE of 0 and a PSNR value of infinity (∞).

An MSE of 0 indicates that there is no difference (error) between the stego-video and the cover video. This suggests that the data embedding process in the stego-video does not alter the video pixels, resulting in perfect fidelity when compared to the original video. An infinite PSNR value (∞) signifies that the quality of the stego-video is exceptionally high, with no signal degradation compared to the original video. A high PSNR typically reflects a better image quality, and in this case, the PSNR reaches infinity because there are no errors or alterations between the stego-video and the original video. Thus, it can be concluded that all the tested stego-videos exhibit perfect fidelity based on the MSE and PSNR metrics, indicating that the data embedding in the video has no impact on the visual quality.

B. Robustness Testing

Robustness testing is conducted to assess the resilience of the stego-video, which has been embedded with a message, against various video processing attacks and to verify whether the message can still be extracted from the stego-video. In this study, the attacks tested include video cropping, video rotation, resizing, and sharing the stego-video through social media platforms, such as email, WhatsApp, Telegram, Instagram, and Facebook. Additionally, tests were performed by targeting the metadata file using several video editing tools.

The video cutting test is carried out by cutting the video duration using the Python programming language. Table V presents the results of the video cropping testing using stego-video 'elephant.mp4' along with the message extraction results. Testing is carried out by cutting the video with a duration percentage, namely: 25%, 50%, 75%, and 90%.

TABLE V. VIDEO CROPPING TEST RESULTS

Crop percentage (%)	Video length after cutting (s)	Result
25	24.75	Succeed
50	16.50	Succeed
75	8.250	Succeed
90	3.300	Fail

Based on the testing results presented in Table VI, which involved trimming the video duration and extracting the message, several insights can be gathered regarding the robustness of the employed steganography method. The testing was conducted by cutting the video duration by varying percentages: 25%, 50%, 75%, and 90%. The extraction results indicate that: when 25% of the video duration was cut, the message extraction was successful; when 50% was cut, the extraction was successful; when 75% was cut, the extraction was also successful; however, with a 90% cut, the extraction failed.

The successful extraction at 25%, 50%, and 75% cuts demonstrates that the steganography method is quite robust and capable of maintaining the integrity of the hidden message, even after a significant portion of the video has been removed. In contrast, the failure to extract the message after a 90% cut indicates that the message's integrity could not be preserved, leading to extraction failure. The successful extraction of messages at various cutting percentages suggests that the steganography method exhibits good robustness if the embedded data remain within the remaining parts of the video. At 25%, 50%, and 75% cuts, this method effectively preserves the hidden information, allowing for complete message extraction. However, with a 90% cut, only 10% of the original video duration (3.3 s out of the total duration) remains. This implies that a large portion of the video content has been lost, including segments that may have contained the hidden message. When the area where the message was embedded is in the trimmed section, the message becomes fragmented and unextractable.

From these results, it can be concluded that the greater the percentage of the video duration cut is, the higher is the likelihood of losing the section containing the message, leading to extraction failure. The failure to extract the message after a 90% cut indicates that this steganography method relies heavily on the number of frames or the duration of the video. In other words, if the remaining video area is too small, the information contained within is no longer sufficient to reconstruct the original message. This suggests that embedding messages in the video is more effective when the messages are spread throughout the entire video rather than confined to specific areas, thereby increasing the resilience against the duration cuts. In previous studies that used video steganography, the method they developed could not withstand the video rotation attacks [44-46]. The messages cannot be extracted if the video is rotated. Table VI illustrates the results of video steganography robustness testing by rotating the video position from 450, up to 1800 playbacks. The stego-video used is 'stego_giraffe.mp4'.

In the tests presented in Table VII, the robustness of the video steganography was evaluated by rotating the stego-video at various angles: 45°, 90°, and 180°. The results indicate that the hidden message could be perfectly extracted at each rotation performed. Video rotation is a form of video processing attack that can alter the visual orientation and frame structure within the video. Typically, such changes in orientation can affect the pixel distribution used for embedding the steganographic message. However, the test results demonstrate that all messages can be fully extracted after the stego-video is rotated at 45°, 90°, and 180 degrees: the message is intact.

TABLE VI. ROTATE MANIPULATION FOR STEGO_GIRAFFE.MP4

Degree of rotation	Result
45^0	Succeed
90^0	Succeed
180^0	Succeed

TABLE VII. VIDEO RESIZE TEST RESULTS

Resize percentage (%)	File size (MB)	Result
25	12.8	Succeed
50	8.50	Succeed
75	4.20	Succeed
90	1.70	Succeed

The successful extraction of messages at each rotation angle indicates that the steganography method employed is highly robust against the changes in orientation. This suggests that the technique used for embedding messages within the stego-video does not rely on the specific orientation of pixels or frames, thereby minimizing the risk of message loss during video rotation.

The fact that the message can be fully extracted after rotation indicates that the deployed steganography method possesses good resistance to the changes in frame orientation. This is because the embedded message does not interfere with the existing frames and pixels in the video. The study developed this method by embedding the secret message into the video's metadata features.

Video resizing is a process for compressing or compressing the stego-video. This test was carried out by compressing the stego-video using the Python programming language. Testing was carried out by reducing the video size from 25%, 50%, 75%, and 90%. Table VII lists the results of the stego-video resize test and the extraction results. The stego-video tested is 'stego_seafood.mp4'.

In the tests presented in Table VII, the VSTM model was utilized to embed a hidden message in the video's metadata. The testing involved resizing the video to various percentages of its original size: 25%, 50%, 75%, and 90%. The results indicate that the message could be perfectly extracted from each resized video. The VSTM model operates by embedding the message in the metadata section of the video file rather than in the main video data, such as pixels or frames. Metadata contain additional information that detail the video, including the title, description, codec, timestamp, and other parameters that do not directly impact the visual content of the video. Consequently, when the video is resized, the VSTM model exhibits greater resistance to modifications because the metadata do not necessarily change during the resizing process.

This success demonstrates that the VSTM model does not rely on the content of the video altered by resizing; instead, it depends on the metadata structure, which remains unchanged or undergoes minimal changes during the resizing process. The robustness of the VSTM model against video resizing can be

explained by considering several characteristics of the metadata and the resizing process.

- Changing the video size does not change the metadata directly: When a video is resized, changes only occur in the visual content, such as frame resolution and video bitrate. Metadata, such as information about the author, title, or even some other technical parameters, do not undergo changes during the video resizing. This keeps the messages embedded in metadata intact and can be extracted without any problems.
- Resizing the video changes the visual content, not the metadata structure: The video resizing process generally focuses on changing the number of pixels per frame or the video quality by adjusting the bitrate and resolution. Even though the video file size is reduced, the metadata structure is not compromised because the resizing only impacts the visual content. In this way, the messages inserted in the metadata can still be maintained even if the video file size changes.

Based on the data evidenced in Table VII, even though the file size was reduced by 90% of the original size (to only 1.7 MB), the extraction results still showed success. This indicates that the steganographic messages inserted using the VSTM model are not affected by the reduction in the video file size. The successful extraction of messages at all resizing levels shows that the VSTM model has excellent resistance to video resizing. With high resistance to video resizing, the VSTM model can be an effective solution for hiding information in videos that are susceptible to size modification, either intentionally or unintentionally.

Social media are often used to send messages through the 'inbox' feature. To evaluate the resilience of the embedded message in the video using the VSTM model, the study conducted a trial by embedding a message into the video and then sharing it on various social media platforms, including email, WhatsApp, Telegram, Instagram, and Facebook. The video was then downloaded and extraction was performed to verify whether the message could still be retrieved. The results of the stego-video testing conducted via social media are presented in Table VIII. The stego file, 'stego_animals.mp4' was embedded and utilized for this test.

A fidelity test was conducted to evaluate the resilience of the steganography method in videos that have been manipulated using various video editing tools or applications. The results outlined in Table IX indicate that only a few applications successfully preserved the embedded message, while others failed to extract the message intact. The findings show that the success of extracting the steganographic message is highly dependent on the video editing application used. Applications, such as VSDC Free Video Editor, Filmora, Shotcut, and HitFilm Express, successfully maintained the metadata, allowing for the complete extraction of the message. In contrast, applications, like Adobe Premiere Pro and Final Cut Pro, demonstrated a failure to preserve the metadata, resulting in the loss of the message. The extraction failure after editing the stego-video occurred because these applications altered the 'comments' attribute to the name label of the application used.

TABLE VIII. TEST RESULTS THROUGH SOCIAL MEDIA

Social media	Extraction results
E-mail	Success
WhatsApp	Success
Telegram	Success
Facebook	Success
Instagram	Failure

TABLE IX. MESSAGE DURABILITY TEST RESULTS ON VIDEO METADATA

Name of tools / applications for manipulation	Result	Information
VSDC Free Video Editor	Succeed	Can preserve metadata but may be lost if the export settings are changed.
Filmora	Succeed	It does not damage metadata, but if the export settings are done incorrectly, messages cannot be extracted
Shotcut	Succeed	Preserves metadata well, especially if the compression settings are not changed
HitFilm Express	Succeed	Generally, preserves metadata, but must be careful with export settings
Adobe Premiere Pro	Fail	Incorrect export settings can corrupt metadata
Final Cut Pro	Fail	Cannot retain the message content in the metadata
DaVinci Resolve	Fail	Too many editing options will cause the metadata to be damaged
iMovie	Fail	The resulting large export file causes the metadata file to be damaged
Windows Movie Maker	Fail	May corrupt metadata when saving or exporting videos
CyberLink PowerDirector	Fail	Corrupts metadata if certain export settings are used

The fidelity test results indicate that the visual quality of videos that have undergone information hiding processes remains within acceptable tolerance limits, with changes that are undetectable by human vision (imperceptible). The testing was conducted using standard metrics, such as PSNR and MSE, which collectively evaluate the fidelity between the original video and the stego-video. The high PSNR values indicate that the distortion occurring is minimal, thus the hidden information does not affect the user's visual experience. These results prove that the deployed hiding method is not only safe and effectively concealed, but also maintains the integrity of the original media, making it suitable for applications in secret communication systems and visual data protection.

C. Discussion

Based on the results of the experiments and tests conducted, the message embedding method using the "mutagen" video metadata, combined with Fernet cryptography and ZLIB compression, successfully embeds messages into the metadata space of the video under the 'comments' attribute. The development of this method aims to address resilience issues, such as cropping, rotation, resizing, transmission of video

through social media, and testing the robustness of messages using video editing tools. The fidelity analysis showed that the MSE for all stego-videos is 0, indicating no changes between the cover video and the stego-video. Meanwhile, the PSNR obtained from this method across five video samples resulted in an infinite value (∞) for all stego-videos, due to the absence of changes in the pixel intensity and bit arrangement. This method demonstrates an improvement over the previous techniques. Additionally, this method produces better video quality compared to several benchmark studies [39–45]. A compression test was also conducted on the messages to be embedded into the stego-video using ZLIB compression, achieving an average file compression rate of 60% for *.txt files.

For robustness testing, the videos were manipulated in several ways, including cropping, rotating, resizing, and sending the stego-videos through social media platforms, such as email, WhatsApp, Telegram, Instagram, and Facebook. The metadata file was also tested using various video editing tools. The cropping tests indicated that the message could be fully extracted from the stego-video even after 25%, 50%, and 75% cropping. However, the message could not be extracted after 90% cropping because significant cuts compromised the metadata file.

Further testing of video manipulation through rotation demonstrated that the message in the stego-video could still be extracted, regardless of whether the video was rotated at 45°, 90°, or 180°. The compression tests demonstrated that the bmessages could be fully extracted even when compressing the video from 25% to 90%, as long as the metadata containing the message remained intact. Additionally, a secret message was embedded into the stego-video and then shared on various social media platforms. The results showed that the message could be successfully extracted from platforms, like email, WhatsApp, Telegram, and Facebook, but not from Instagram, likely due to Instagram's aggressive compression algorithms that often strip metadata from videos.

Videos were also tested using editing tools that could damage the metadata file. The results indicated that some tools, such as Adobe Premiere Pro, Final Cut Pro, DaVinci Resolve, iMovie, Windows Movie Maker, and CyberLink PowerDirector, could corrupt the metadata. However, tools, like VSDC Free Video Editor, Filmora, Shotcut, and HitFilm Express, were proven not to damage the metadata, allowing for the successful extraction of the hidden messages.

The developed steganography model has proven to be a viable alternative for securing digital data through video. This method demonstrates resilience against manipulation and can be utilized for message delivery on social media platforms. Furthermore, the developed method maintains the quality of the stego-video, showing no significant changes when compared to the original stego-video.

V. CONCLUSION

The study developed a novel Video Steganography Technique in Metadata (VSTM) that uses video metadata to embed secret messages, improving both the data-hiding capacity and security. By combining Advanced Encryption Standard (AES) encryption and ZLIB compression, the VSTM model ensures that the visual and audio quality of the video remains intact while concealing the message securely within the metadata fields, specifically the 'comments' attribute. The method was tested for robustness against various manipulations, including cropping, rotation, resizing, and social media sharing. The results showed that the method could successfully recover messages under most conditions, except when the video was cropped beyond 90% or shared via Instagram, which removes metadata. Video editing tools, like Filmora and Shotcut preserved the metadata, while others, like Adobe Premiere corrupted them. The fidelity tests indicated a perfect quality retention, with no pixel changes detected, demonstrating that the VSTM model effectively preserves the video integrity. It is concluded that VSTM offers a reliable and efficient solution for digital data security.

The main contribution of this research lies in the development of a new model that utilizes video metadata as a space for embedding secret information in a hidden and secure manner, without altering the main visual content of the video itself. This approach introduces a new paradigm in video steganography, where message integration occurs at the descriptive level of the file (metadata) such as tags, headers, or other technical information, which are generally overlooked by conventional media processing. This model not only enhances the imperceptibility as it does not affect the visual quality of the video, but also expands the insertion capacity and reduces the likelihood of detection by traditional steganalysis techniques. Furthermore, this model demonstrates high efficiency in terms of extraction speed and cross-platform compatibility, making it a potential solution for fast, lightweight, and secure secret information delivery applications.

ACKNOWLEDGMENT

The authors extend their gratitude to the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia for supporting this project with funding under contract numbers 104/E5/PG.02.00.PL/2024, 1132 /LL2/KP/PL/2024, 004/UTI/LPPMI/E.1.1/VI/2024.

REFERENCES

[1] T. Qiao, X. Luo, B. Pan, Y. Chen, and X. Wu, "Toward Steganographic Payload Location via Neighboring Weight Algorithm," *Security and Communication Networks*, vol. 2022, pp. 1–17, Feb. 2022, https://doi.org/10.1155/2022/1400708.

[2] A. Gupta, H. Shukla, and M. Gupta, "A secure image steganography using X86 assembly LSB," *NEU Journal for Artificial Intelligence and Internet of Things*, vol. 1, no. 1, pp. 38–47, 2022.

[3] Z. Zhou *et al.*, "Secret-to-Image Reversible Transformation for Generative Steganography," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 5, pp. 4118–4134, Sep. 2023, https://doi.org/10.1109/tdsc.2022.3217661.

[4] B. Yamini and R. Sabitha, "Image steganalysis: real-time adaptive colour image segmentation for hidden message retrieval and Matthew's correlation coefficient calculation," *International Journal of Information and Computer Security*, vol. 17, no. 1/2, p. 83, 2022, https://doi.org/10.1504/ijics.2022.121292.

[5] H. G. Zaini, "Image Segmentation to Secure LSB2 Data Steganography," *Engineering, Technology & Applied Science Research*,

vol. 11, no. 1, pp. 6632–6636, Feb. 2021, https://doi.org/10.48084/etasr.3859.

[6] N. Alsharari, "Integrating Blockchain Technology with Internet of things to Efficiency," *International Journal of Technology, Innovation and Management (IJTIM)*, vol. 1, no. 2, pp. 01–13, Dec. 2021, https://doi.org/10.54489/ijtim.v1i2.25.

[7] A. Munshi, "Randomly-based Stepwise Multi-Level Distributed Medical Image Steganography," *Engineering, Technology & Applied Science Research*, vol. 13, no. 3, pp. 10922–10930, Jun. 2023, https://doi.org/10.48084/etasr.5935.

[8] F. Al-Shaarani and A. Gutub, "Securing matrix counting-based secret-sharing involving crypto steganography," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 9, pp. 6909–6924, Oct. 2022, https://doi.org/10.1016/j.jksuci.2021.09.009.

[9] T. AlKhodaidi and A. Gutub, "Refining image steganography distribution for higher security multimedia counting-based secret-sharing," *Multimedia Tools and Applications*, vol. 80, no. 1, pp. 1143–1173, Jan. 2021, https://doi.org/10.1007/s11042-020-09720-w.

[10] X. Lv and M. Li, "Application and Research of the Intelligent Management System Based on Internet of Things Technology in the Era of Big Data," *Mobile Information Systems*, vol. 2021, pp. 1–6, Jun. 2021, https://doi.org/10.1155/2021/6515792.

[11] J. Qin, Y. Luo, X. Xiang, Y. Tan, and H. Huang, "Coverless Image Steganography: A Survey," *IEEE Access*, vol. 7, pp. 171372–171394, 2019, https://doi.org/10.1109/access.2019.2955452.

[12] Z. Qu, H. Sun, and M. Zheng, "An efficient quantum image steganography protocol based on improved EMD algorithm," *Quantum Information Processing*, vol. 20, no. 2, Feb. 2021, https://doi.org/10.1007/s11128-021-02991-8.

[13] X. Duan, D. Guo, N. Liu, B. Li, M. Gou, and C. Qin, "A New High Capacity Image Steganography Method Combined With Image Elliptic Curve Cryptography and Deep Neural Network," *IEEE Access*, vol. 8, pp. 25777–25788, 2020, https://doi.org/10.1109/access.2020.2971528.

[14] D. Darwis, A. Thyo Priandika, A. Surahman, A. Ferico Octaviansyah Pasaribu, A. Junaidi, and Wamiliana, "Combination of Modified LSB Steganography and Huffman Compression for Data Security," in *2021 International Conference on Computer Science, Information Technology, and Electrical Engineering*, Banyuwangi, Indonesia, Oct. 2021, pp. 218–224, https://doi.org/10.1109/icomitee53461.2021.9650312.

[15] D. R. I. M. Setiadi, "PSNR vs SSIM: imperceptibility quality assessment for image steganography," *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 8423–8444, Mar. 2021, https://doi.org/10.1007/s11042-020-10035-z.

[16] N. A. Taha, A. Al Saffar, A. A. Abdullatif, and F. A. Abdullatif, "Image Steganography using Dynamic Threshold based on Discrete Cosine Transform," *Journal of Physics: Conference Series*, vol. 1879, no. 2, May 2021, Art. no. 022087, https://doi.org/10.1088/1742-6596/1879/2/022087.

[17] T. J. Siddiqui and A. Khare, "Chaos-based Video Steganography Method in Discrete Cosine Transform Domain," *International Journal of Image and Graphics*, vol. 21, no. 02, Apr. 2021, Art. no. 2150015, https://doi.org/10.1142/s0219467821500157.

[18] M. Nazari and M. Mehrabian, "A novel chaotic IWT-LSB blind watermarking approach with flexible capacity for secure transmission of authenticated medical images," *Multimedia Tools and Applications*, vol. 80, no. 7, pp. 10615–10655, Mar. 2021, https://doi.org/10.1007/s11042-020-10032-2.

[19] M. Suresh and I. S. Sam, "Exponential fractional cat swarm optimization for video steganography," *Multimedia Tools and Applications*, vol. 80, no. 9, pp. 13253–13270, Apr. 2021, https://doi.org/10.1007/s11042-020-10395-6.

[20] T. T. K. Hue, N. T. Linh, M. Nguyen-Duc, and T. M. Hoang, "Data Hiding in Bit-plane Medical Image Using Chaos-based Steganography," in *2021 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, Hanoi, Vietnam, Oct. 2021, pp. 1–6, https://doi.org/10.1109/mapr53640.2021.9585243.

[21] A. V. Sergeev and P. B. Khorev, "Overview on Hiding Data in Microsoft Office Documents," in *2021 3rd International Youth Conference on Radio Electronics, Electrical and Power Engineering*, Moscow, Russia, Mar. 2021, pp. 1–4, https://doi.org/10.1109/reepe51337.2021.9388038.

[22] R. Wazirali, Z. Chaczko, and J. Gibbon, "Steganographic Image Sharing App," in *2017 25th International Conference on Systems Engineering (ICSEng)*, Las Vegas, NV, Aug. 2017, pp. 494–499, https://doi.org/10.1109/icseng.2017.62.

[23] L. Hartmann and S. Wendzel, "How Feasible are Steganographic and Stealth Attacks on TIA Project Metadata of ICS: A Case Study with Real-world Data," in *European Interdisciplinary Cybersecurity Conference*, Virtual Event Romania, Nov. 2021, pp. 83–84, https://doi.org/10.1145/3487405.3487661.

[24] C. M. Kondasinghe, "A System to Preserve Metadata using Steganography," M. S. Thesis, University of Colombo School of Computing, Colombo, Sri Lanka, 2017.

[25] H. Jain, "Constant Sound Steganography," *The American Journal of Interdisciplinary Innovations and Research*, vol. 03, no. 07, pp. 5–9, Jul. 2021, https://doi.org/10.37547/tajiir/volume03issue07-02.

[26] S. D. Muyco and A. A. Hernandez, "A Modified Hash Based Least Significant Bits Algorithm for Steganography," in *Proceedings of the 2019 4th International Conference on Big Data and Computing*, Guangzhou, China, 2019, pp. 215–220, https://doi.org/10.1145/3335484.3335514.

[27] R. Rejani, D. Murugan, and D. V. Krishnan, "Pixel Pattern Based Steganography on Images," *ICTACT Journal on Image and Video Processing*, vol. 5, no. 3, pp. 991–997, Feb. 2015, https://doi.org/10.21917/ijivp.2015.0146.

[28] N. A. Al-Juaid, A. A. Gutub, and E. 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, https://doi.org/10.26735/16587790.2018.006.

[29] B. S. Shashikiran, K. Shaila, and K. R. Venugopal, "Minimal Block Knight's Tour and Edge with LSB Pixel Replacement Based Encrypted Image Steganography," *SN Computer Science*, vol. 2, no. 3, May 2021, Art. no. 139, https://doi.org/10.1007/s42979-021-00542-7.

[30] A. Sweigart, *Hacking Secret Ciphers with Python: a beginner's guide to cryptography and computer programming with Python*, 1. ed. Scotts Valley, California, US: Createspace, 2013.

[31] H. K. Tayyeh and A. S. Ahmed AL-Jumaili, "A combination of least significant bit and deflate compression for image steganography," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, Feb. 2022, Art. no. 358, https://doi.org/10.11591/ijece.v12i1.pp358-364.

[32] S. Moufid, F. Z. Lachgar, C. El Ainzroudi, and A. A. Ben El Arbi, "Improving Steganography using Image Compression JPEG," in *2021 15th International Conference on Advanced Technologies, Systems and Services in Telecommunications*, Nis, Serbia, Oct. 2021, pp. 405–410, https://doi.org/10.1109/telsiks52058.2021.9606250.

[33] S. Roy and Md. M. Islam, "A Hybrid Secured Approach Combining LSB Steganography and AES Using Mosaic Images for Ensuring Data Security," *SN Computer Science*, vol. 3, no. 2, Mar. 2022, Art. no. 153, https://doi.org/10.1007/s42979-022-01046-8.

[34] C. A. Sari, G. Ardiansyah, D. R. I. M. Setiadi, and E. H. Rachmawanto, "An improved security and message capacity using AES and Huffman coding on image steganography," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 17, no. 5, Oct. 2019, Art. no. 2400, https://doi.org/10.12928/telkomnika.v17i5.9570.

[35] E. Hartnett, C. S. Zender, W. Fisher, and D. Heimbigner, "Quantization and Next-Generation Zlib Compression for Fully Backward-Compatible, Faster, and More Effective Data Compression in NetCDF Files," in *AGU Fall Meeting 2021*, New Orleans, LA, Dec. 2021, pp. IN15A-05.

[36] A. P. Maulidina, R. A. Wijaya, K. Mazel, and M. S. Astriani, "Comparative Study of Data Compression Algorithms: Zstandard, zlib & LZ4," in *Communications in Computer and Information Science*, Cham: Springer Nature Switzerland, 2024, pp. 394–406.

[37] E. S. B. Hureib and A. A. Gutub, "Enhancing Medical Data Security via Combining Elliptic Curve Cryptography with 1-LSB and 2-LSB Image Steganography," *International Journal of Computer Science and Network Security*, vol. 20, no. 12, pp. 232–241, Dec. 2020, https://doi.org/10.22937/IJCSNS.2020.20.12.26.

[38] A. Gutub and F. Al-Shaarani, "Efficient Implementation of Multi-image Secret Hiding Based on LSB and DWT Steganography Comparisons," *Arabian Journal for Science and Engineering*, vol. 45, no. 4, pp. 2631–2644, Apr. 2020, https://doi.org/10.1007/s13369-020-04413-w.

[39] B. O. Al-Roithy and A. A. Gutub, "Trustworthy image security via involving binary and chaotic gravitational searching within PRNG selections," *International Journal of Computer Science and Network Security*, vol. 20, no. 12, pp. 167–176, Dec. 2020, https://doi.org/10.22937/IJCSNS.2020.20.12.18.

[40] A. A.-A. Gutub, "Adopting counting-based secret-sharing for e-Video Watermarking allowing Fractional Invalidation," *Multimedia Tools and Applications*, vol. 81, no. 7, pp. 9527–9547, Mar. 2022, https://doi.org/10.1007/s11042-022-12062-4.

[41] Z. Saeidi, A. Yazdi, S. Mashhadi, M. Hadian, and A. Gutub, "High performance image steganography integrating IWT and Hamming code within secret sharing," *IET Image Processing*, vol. 18, no. 1, pp. 129–139, Jan. 2024, https://doi.org/10.1049/ipr2.12938.

[42] Z. Y. Hasan and A. K. Idrees, "EDiTOK: Energy-Efficient Distributed Transmission Optimization-Based Kalman–Zlib Approaches for Monitoring Smart Farming Irrigation in Internet of Things Networks," in *Lecture Notes in Networks and Systems*, Singapore: Springer Nature Singapore, 2024, pp. 117–130.

[43] J. M. Reddy, A. Voma, and P. Themdeo, "Audio Steganography Using Multi-Level DWT to Improve PSNR, Hiding Capacity and Imperceptibility," in *Algorithms for Intelligent Systems*, Singapore: Springer Singapore, 2021, pp. 309–314.

[44] I. A. Sabilla, M. Meirisdiana, D. Sunaryono, and M. Husni, "Best Ratio Size of Image in Steganography using Portable Document Format with Evaluation RMSE, PSNR, and SSIM," in *2021 4th International Conference of Computer and Informatics Engineering*, Depok, Indonesia, Sep. 2021, pp. 289–294, https://doi.org/10.1109/ic2ie53219.2021.9649198.

[45] S. M. Affiq, S. I. Shaiden, and K. Subramaniam, "Android based Digital Steganography Application using LSB and PSNR Algorithm in Mobile Environment," *Evergreen*, vol. 8, no. 2, pp. 421–427, Jun. 2021, https://doi.org/10.5109/4480724.

[46] H. Paraskevov, S. Zhelezov, and B. Uzunova-Dimitrova, "Robustness of the secret message in stego file against flip and rotation attack," *Annals of the Academy of Romanian Scientists: Series on Mathematics and its Applications*, vol. 9, no. 1, pp. 5–16, Aug. 2025.