INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

A New Approach of Steganography on Image Metadata

Yusra Fernando [a], Dedi Darwis [a,*], Abhishek R Mehta [b], Wamiliana [c], Agus Wantoro [a]

[a] *Faculty of Engineering and Computer Science, Universitas Teknokrat Indonesia, Kedaton, Bandar Lampung, Indonesia*
[b] *Faculty of IT & Computer Science, Parul University, Vadodara, Gujarat, India*
[c] *Department of Mathematics, Universitas Lampung, Rajabasa, Bandar Lampung, Indonesia*
*Corresponding author: *darwisdedi@teknokrat.ac.id*

***Abstract*—In this paper, we introduce a novel method, Steganography on Image Metadata (SIM), to tackle the problem of robustness modification in steganography. The SIM method works by embedding messages into the metadata storage space of digital media. Metadata is information embedded in a file that explains the file's content. The advantage of this method is that it does not alter the pixel values in the image, ensuring no degradation in media quality, and the secret message remains secure even when robustness manipulations are applied to the stego-image. To enhance data security, this paper also suggests using Fernet cryptography for message encryption during the embedding process into the cover-image. According to experimental evaluations, the SIM technique can attain a maximum PSNR value of 100 dB and an outstanding MSE value of 0. All robustness manipulation issues in steganography can be effectively addressed using the SIM method. Test results demonstrate that the SIM method can withstand symmetric and asymmetric cropping manipulations down to a pixel size of 1x1, and the message can still be extracted. Testing with image rotation manipulation also proves that the message can be successfully extracted even when the stego-image is rotated up to 180 degrees. Experiments with image resizing manipulation also confirm that the message can be recovered even when the stego-image undergoes up to 90% compression. Testing with color effects applied to the image also does not affect message extraction results.**

***Keywords*— Cover-image; fernet; robustness; SIM; steganography.**

Manuscript received 8 Sep. 2023; revised 16 Nov. 2023; accepted 5 Jan. 2023. Date of publication 31 May 2024.

I. INTRODUCTION

Digital communication has recently become a fundamental part of our infrastructure, with numerous internet-based applications [1]–[3]. Generally, the internet does not use secure links, making transmitted information vulnerable to attacks from external sources[4]–[6]. It is now crucial to lessen the possibility of discovering information while being transmitted [7]. As a result, ensuring the security of information passing through open channels has become a fundamental concern [8]. The internet has allowed multimedia material to move quickly and widely in various formats, including text, audio, video, and photos[9]. In internet-based digital communication, all users have access to and can see anything [10]. Thus, information security is an essential consideration. Information security has three main goals: availability, integrity, and secrecy [11], [12].

Information can be protected using various methods, such as steganography, digital watermarking, reversible watermarking, encryption, and watermarking [12], [13]. This paper will focus on the technique of steganography. By hiding confidential messages within other media, steganography protects data by preventing unwelcome recipients from becoming suspicious of its existence. It is an extension of cryptographic techniques [14]. The message in the cover image when using picture steganography to hide the secret message from attackers—those who lack the legal right to access the data—is referred to as a stego-image. The goal of steganography is not only to prevent unauthorized parties from decoding the hidden message but also to ensure that the hidden message does not arouse suspicion [15]–[17].

The best method to evaluate the quality of steganography-produced images is to verify the accuracy of the methods or algorithms utilized for the extraction and embedding procedures. Such testing can be done in four ways: robustness, recovery, fidelity, and imperceptibility [18], [19]. The robustness manipulation process can pose challenges in image steganography. Typically, images containing hidden messages are not resilient to this type of attack, and the messages may become damaged during extraction after robustness manipulation [20], [21].

The first robustness manipulation involves cropping the image. This poses a challenge in image steganography because the cropped stego-image will experience damage to the secret message during the extraction process due to changes in pixel values. The second adjustment of robustness involves resizing the stego-image. Resizing involves changing the image's resolution size, which enlarges or compresses the stego-image relative to its initial size. The third robustness manipulation is rotating the image. Stego-images typically lack resistance to image rotation attacks because even a slight shift in the previous position can alter pixel values. The last robustness manipulation examined in this study is testing the resilience of the stego-image by applying a Gaussian blur attack, which involves applying a blur effect generated by a Gaussian function [22], [23].

The problem of stego-image resilience against robustness manipulations remains unsolved. Generally, steganography techniques are only resistant to a single kind of robustness manipulation attack, such cropping the image. Changes in pixel values and locations of the image prevent embedded messages from being retrieved during robustness editing. Research discussing the use of the Center Embedded Pixel Positioning (CEPP) method can withstand cropping attacks up to 70%. In this method, messages are embedded in the center of the cover image, ensuring the message remains secure if cropping occurs only on the left, right, top, and bottom positions. However, the drawback of this method is that it can only withstand cropping manipulation; if other image manipulations are performed, the message may not be extracted intact [24].

The problem to be investigated in this paper focuses on two aspects of steganography that still have weaknesses: the deterioration of cover-image quality due to pixel value changes and the resilience of stego-images against robustness manipulations. Many pixel value changes during the process of embedding digital data into the cover-image are the reason behind the reduction in the quality of the container media generated by stego-images and the problem of robustness manipulation. The objective of this paper is to develop a new method with novelty in the field of steganography, offering an alternative for secure digital data storage. This new method is called "Steganography on Image Metadata" (SIM). The SIM method works by embedding messages into the metadata storage space of digital media. Metadata is information embedded in a file, explaining the file itself [25], [26]. The advantage of this method is that it does not alter the existing pixel values in the image, ensuring no loss of media quality, and the embedded secret message remains secure even when stego-images undergo robustness manipulation. In this research, images in JPEG format are used as the cover-images because this media format offers metadata space that can be utilized for embedding text or txt file-based messages. To enhance data security, the paper also proposes using Fernet cryptography for the encryption process during the message embedding into the cover-image.

II. Materials and Method

Essentially, all digital images have metadata that serves to contain digital information. Information stored in image metadata is generally viewable by anyone and includes details such as file name, image dimensions, file size, image type, and more. In images with JPG/JPEG formats, there are specific attributes in the metadata where confidential information can be embedded, such as the program name. With this work, the method of embedding hidden messages into the image metadata—known as Steganography on image Metadata, or SIM—was developed. This method is an extension of previous developments, including the Center Sequential Technique (CST) and Center Embedded Pixel Positioning (CEPP), which address the protection of stego-images against robustness manipulation [24], [27]. The SIM method is combined with the Fernet cipher algorithm, which is used to create a stego-key and perform encryption before embedding the message into the image.

The Fernet cipher algorithm is a cryptography method from one of the Advanced Encryption Standard (AES) development packages [28]. It is typically employed for creating password keys and performing relatively uncomplicated encryption [29]. This study develops the Fernet algorithm utilizing existing Python programming libraries.

The essential idea behind image steganography is that after a message is embedded, the cover image's quality shouldn't alter noticeably. In other words, the smaller the pixel changes in the image, the better the adherence to the principles of effective steganography. This study aims to meet picture steganography quality standards by minimizing pixel alterations in the cover image. This study also highlights how strong stego-images are against image processing techniques such as cropping, resizing, rotating, and applying color, noise, and Gaussian blur effects [30].

A. Message Insertion Process

The message that has to be embedded can be either a *.txt file or a string. Before embedding the message, an encryption process is applied using the Fernet cipher algorithm to enhance message security. Additionally, the Fernet cipher algorithm is employed to create the stego-key. The procedure of message embedding is shown in Fig 1.

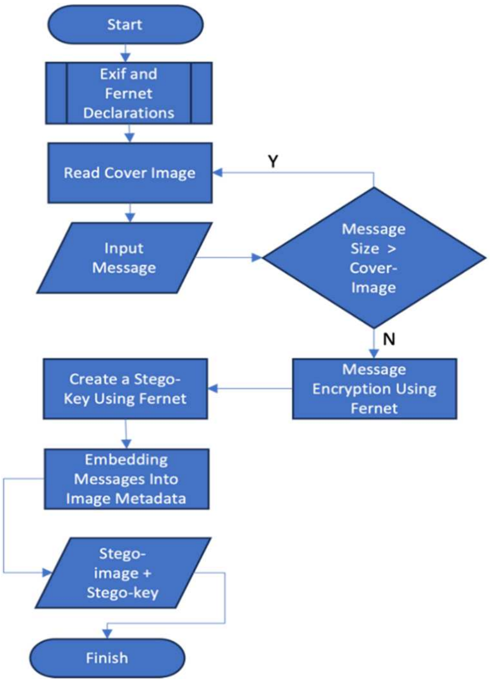

Start
Exif and Fernet Declarations
Read Cover Image
Y
Input Message
Message Size > Cover-Image
N
Create a Stego-Key Using Fernet
Message Encryption Using Fernet
Embedding Messages Into Image Metadata
Stego-image + Stego-key
Finish

Fig. 1 Flowchart Message Insertion Process

Declaring functions for Fernet cryptography and using Exif to read metadata are the first steps in the workflow for the proposed method's message embedding procedure. Subsequently, the image is read as the cover image, which can be either RGB or grayscale and is in JPG format. After that, a string representing the message is entered. If the message size exceeds the cover-image's size, the message embedding process cannot be finished.

An encryption process employing Fernet cryptography is carried out before the message being embedded into the image. Research conducted by Wazirali et al [31] explains that when using Exif for message embedding, the information becomes open and readable by others. Therefore, in the development of this study, cryptography techniques are combined to address the weaknesses of previous research. Furthermore, the stego-key, utilized in the message extraction procedure, is created using Fernet cryptography. The stego-key is generated symmetrically, meaning the same key is used for both message embedding and extraction.

B. Message Extraction Process

The receiver will receive the message once it has been successfully incorporated into the cover image. The recipient of the communication then uses the stego-key to extract the message. The message extraction flowchart from the stego-image is shown in Fig. 2.

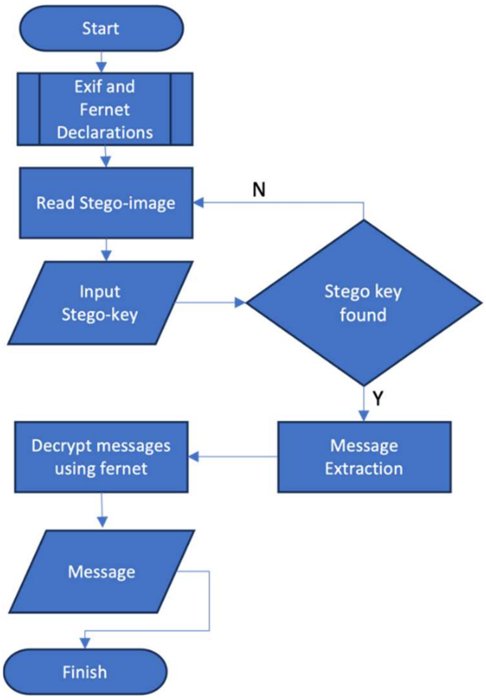


Fig. 2 Flowchart Message Extraction Process

Reading the stego-image is the first step in the message extraction process, after which functions for Fernet and Exif are declared. The stego-image reading procedure will then be repeated if the stego-key is inaccurate. Retrieving the message from the image metadata initiates the Exif procedure if the stego-key is accurate. The full message, or original plaintext, must be produced through a decryption procedure because the message received by invoking the Exif function is still in ciphertext form.

III. Result and Discussion

A. Results

1) Implementation SIM Algorithm: As seen in Fig. 1, the message embedding procedure is carried out using the Python programming language. Here is a sample of pseudocode that may be used to process the cover image and secret image.

```
Cover Image Reading Pseudocode and Message
Encryption:
function hide_message(source, destination,
message)
    shutil.copyfile(source, destination)
    src ← cv2.imread(source, 1)
    write(destination, src)
    logging.info("Generate secret key ..")
    key ← Fernet.generate_key()
    fernet ← Fernet(key)
    encrypt_message ←
fernet.encrypt(message.encode())
end function
```

The pseudocode snippet represents the process of reading a cover-image based on the JPG/JPEG image type. This method also involves encryption using the Fernet cipher algorithm to transform plaintext messages into ciphertext.

2) Message Insertion Algoritm: The encrypted message will be embedded in the image metadata in JPG/JPEG format using the exif function.

```
Message Insertion Pseudocode:
function SetImageMetadata(make,
xResolution, yResolution, software)
    zeroth_ifd.Make ← make
    zeroth_ifd.XResolution ← xResolution
    zeroth_ifd.YResolution ← yResolution
    zeroth_ifd.Software ← software
end function

function SetExifMetadata(dateTimeOriginal,
lensMake, sharpness, lensSpecification,
userComment)
    exif_ifd.DateTimeOriginal ←
dateTimeOriginal
    exif_ifd.LensMake ← lensMake
    exif_ifd.Sharpness ← sharpness
    exif_ifd.LensSpecification ←
lensSpecification
    exif_ifd.UserComment ← userComment
end function

function SetGPSMetadata(gpsVersionID,
gpsAltitudeRef, gpsDateStamp)
    gps_ifd.GPSVersionID ← gpsVersionID
    gps_ifd.GPSAltitudeRef ← gpsAltitudeRef
    gps_ifd.GPSDateStamp ← gpsDateStamp
end function

function SetFirstImageMetadata(make,
xResolution, yResolution, software)
    first_ifd.Make ← make
    first_ifd.XResolution ← xResolution
    first_ifd.YResolution ← yResolution
    first_ifd.Software ← software
end function
```

```
SetImageMetadata("Canon", (96, 1), (96, 1), key)
SetExifMetadata("2099:09:29 10:10:10", key, 65535, ((1, 1), (1, 1), (1, 1), (1, 1)), encrypt_message)
SetGPSMetadata((2, 0, 0, 0), 1, "1999:99:99 99:99:99")
SetFirstImageMetadata("Canon", (40, 1), (40, 1), "piexif")
```

The pseudocode snippet represents the embedding process utilizing the exif image metadata. This process differs from the previously developed algorithms, namely CST and CEPP [24], [32]. LSB (Least Significant Bit) embedding was predominantly used in the previous method development. However, in the current method being developed, there is no change made to the pixel values at all. As a result, the encrypted message is directly included into the program name image metadata element. With this approach, the quality of the steganography result will remain high in terms of fidelity. Additionally, the stego-image will withstand alteration attempts like rotation, cropping, scaling, and Gaussian blur.

The quality of the steganography method employed will be higher if, in accordance with the principle of good steganography, the embedded message becomes less evident in the carrier media or if there are minimal or nonexistent changes in pixel values [33], [34].

3) Message Extraction Algorithm: Reading the stego-image from the exif image metadata is the first action taken.

```
Message Extraction Pseudocode:
function unhide_message(stego, key)
  exif_data ← piexif.load(stego)
  exif_ifd ← exif_data.get("Exif", {})
  embeded_key ← exif_ifd.get(42035, None)
  if not embeded_key
  begin
    write("There's no key embeded to this medium")
    return
  end
  if embeded_key.decode()[:5] = key:
  begin
    message ← exif_ifd.get(37510, None)
    fernet ← Fernet(embeded_key)
    result ← fernet.decrypt(message)
    decode()
    write(result)
  else
  begin
    write ("Key is not match")
  end
end function
```

The generated algorithm's pseudocode snippet represents the message extraction process. The process of this approach entails verifying that the supplied stego-key is accurate after reading The message, which is still in ciphertext form, will be decoded back into plaintext or the original message if the stego-key is correct.

B. Discussion

Before conducting the testing process on the developed method, several sample images are prepared for experimentation. There are a total of 5 images to be used as cover images, all in JPG format. JPG format images are chosen as cover images because they contain attributes that can be manipulated to embed messages. The sample images to be tested are shown in Fig 3 below.

Fig. 3 Cover Image

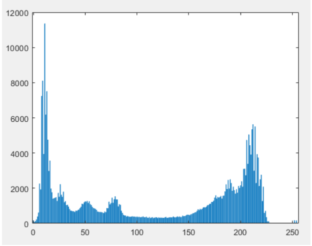
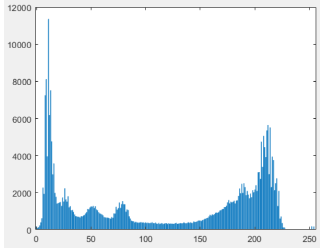
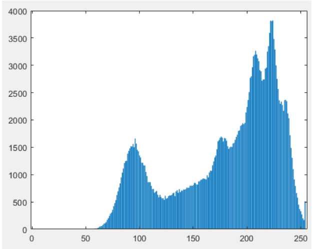
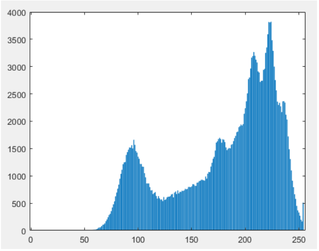
The technique involves adding words to the cover image representing a secret message. The first experiment aims to determine how the cover image changes once the message is incorporated. The embedded message is in the form of a string of 126 bytes. The result of the experiment is shown in Table I below.

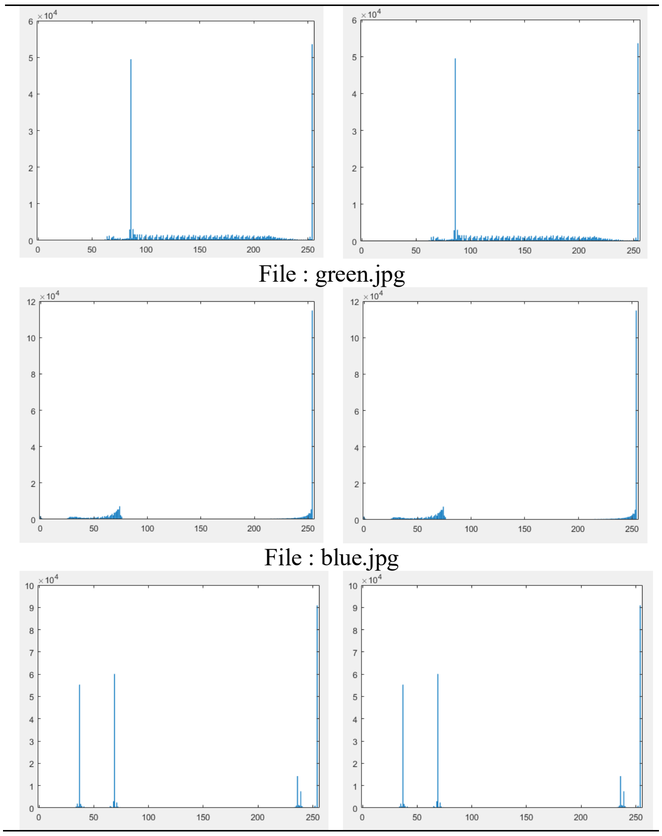
TABLE I
MESSAGE INSERTION TEST RESULTS

Cover Image	Cover Dimensions	Cover Size (kb)	Stego-Image Size (kb)	Number of Pixel Changes
Foto.jpg	680 x 480	28.8	30.5	0
Lenna.jpg	512 x 512	81.1	82.8	0
Red.jpg	512 x 512	15.3	16.9	0
Green.jpg	512 x 512	13.1	14.7	0
Blue.jpg	512 x 512	14.0	15.7	0

There have been minor modifications to the stego-image size shown in Table I, with an average increase of 1.5 KB. Whether the message being embedded is larger or smaller will determine how much the stego-image size increases. The results are shown in Table II below so that you can see changes in pixel values in the cover image and stego-image.

TABLE II
EXAMINATION OF STEGO IMAGE HISTOGRAMS AND COVER IMAGE COMPARISON

Histogram Cover-Image	Histogram Stego-Image
File: foto.jpg	
	
File: lenna.jpg	
	
File : red.jpg	



There is no difference in intensity values between the pixels of the cover image and the generated stego-image, as indicated by the histogram displayed in Table II. This happens because the embedded message modifies the image metadata rather than the bit values of the pixels.

In the developed method, the maximum message size that can be hidden in the image metadata is 8,000 characters or 8 KB. Therefore, this SIM method is only suitable for secret messages that are not too large.

1) Fidelity Testing: The term fidelity describes the capacity to test images without information loss or visual distortion precisely [35], [36]. This procedure will measure the Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) data to perform fidelity testing. One metric used to assess the final image's quality is PSNR [37]. The PSNR method calculates the difference between the created stego image's pixel values and the cover image's pixel values [38]. To determine PSNR, the average square absolute error between the cover image and the stego image is first calculated, or mean square error, or MSE value [38]. Equation 1 contains the formula used to get the MSE value.

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

Information:

MSE_{AVG} = Cover image's average MSE value.
MSE_R = The red MSE value.
MSE_G = The green MSE value.
MSE_B = The blue MSE value.
$X.Y$ = Image dimensions.

Equation 2 contains the formula used to determine the PSNR value.

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

Information:

$PNSR$ = Digital image PSNR value.
MSE = Digital image PSNR value.

The Fidelity test results are displayed in Table III.

TABLE III
FIDELITY TEST RESULTS

Stego Image	MSE	PSNR
Foto.jpg	0	100 db
Lenna.jpg	0	100 db
Red.jpg	0	100 db
Green.jpg	0	100 db
Blue.jpg	0	100 db

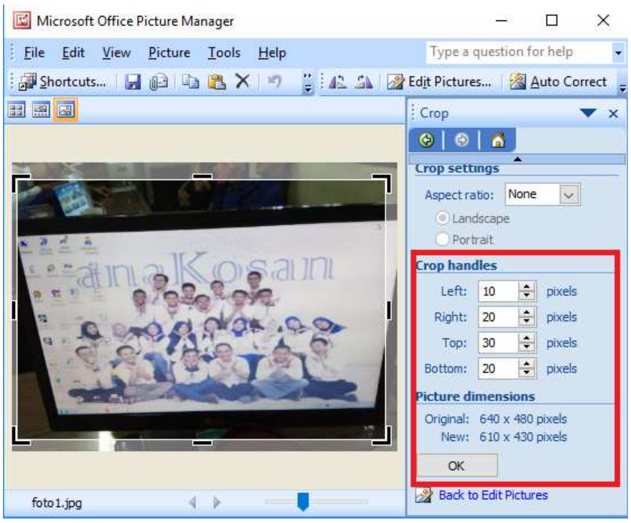
Comparing Table III to the prior technique, the MSE and PSNR values are good [39]. The resulting MSE score of 0 shows that the stego-image's pixel values have not changed. The maximum value of the PSNR is 100 dB. The exceptional quality of the generated stego-image in this procedure can be attributed to the absence of changes in pixel intensity values and dimensions.

2) Robustness Testing

To determine whether the message included in the stego-image can withstand image processing attacks and whether the message can be recovered from the stego-image, robustness testing is carried out. This study discusses image processing attacks such as cropping, rotation, resizing, applying color effects to the image by reducing or altering the stego image pixels, and attacks on the message stored in the metadata. Robustness testing for the developed method is performed using Microsoft Office Picture Manager.

- Cropping Testing

The first robustness test involves symmetric and asymmetric cropping of the stego-image from various directions and different cropping percentages. The cropping test scenario will involve cutting pixels using the Crop Handles feature in Microsoft Office Picture Manager by entering pixel reduction values, as shown in Figure 4.



Crop settings
Aspect ratio: None
○ Landscape
○ Portrait

Crop handles
Left: 10 pixels
Right: 20 pixels
Top: 30 pixels
Bottom: 20 pixels

Picture dimensions
Original: 640 x 480 pixels
New: 610 x 430 pixels

OK

Fig. 4 Use of Crop Handles for a Cropping Test

The testing was conducted on the stego-image named "red.jpg," which has a pixel size 512 x 512. Table IV below presents the results of the symmetric cropping test of the stego-image from the left, right, up, and down directions.

TABLE IV
TEST OF CROP (SYMETRY)

Pixel Value Reduction Amount				Stego-Image Dimension Change	Extraction Results
Left	Right	Up	Down		
25	25	25	25	462 x 462	✓
50	50	50	50	412 x 412	✓
100	100	100	100	312 x 312	✓
150	150	150	150	212 x 212	✓
250	250	250	250	12 x 12	✓
255	255	255	255	2 x 2	✓
255	256	255	256	1 x 1	✓

As the message can be extracted even after cropping the stego-image to a pixel size of 1 x 1, the experimental findings shown in Table IV clearly show that the cropping manipulation assault on the stego-image is unsuccessful. This happens because the image's metadata contains unaltered and complete information (✓ success; ✕ fail).

A second cropping manipulation test was performed by asymmetrically cutting the stego-image with the name "red.jpg" to demonstrate the validity of the established method. The outcomes are shown in Table V.

TABLE V
TEST OF CROP (ASYMETRY)

Amount of Pixel Value Reduction				Stego-Image Dimension Change	Extraction Results
Left	Right	Up	Down		
10	20	40	60	482 x 412	✓
25	50	100	200	437 x 212	✓
50	100	150	200	362 x 162	✓
100	150	200	250	262 x 62	✓
230	280	240	271	2 x 1	✓

Asymmetric cropping tests presented in Table V also demonstrate the reliability of the developed method, as even when pixels are cropped down to a size of 1 x 1, the message can still be extracted.

- Rotate and Flip Testing

In the previous paper that utilized the CST and CEPP methods [24], [32], those methods were unable to withstand rotate and flip manipulation attacks. Messages could not be extracted when the image underwent rotation, even if it was slight. Below are the robustness test results for rotation in Table VI and flip in Table VII. The development of this method can address the robustness issues related to rotation and flip manipulations. The test results show that even when the stego-image is flipped both horizontally and vertically or rotated up to 180 degrees, the message may be effectively recovered intact.

TABLE VI
ROTATE TEST RESULTS

Rotation Degrees	Rotation Results	Pixel Dimension Change	Extraction Results
45^0		724 x 724	✓
90^0		512 x 512	✓
180^0		512 x 512	✓

TABLE VII
FLIP TEST RESULTS

Flip Direction	Flip Results	Extraction Results
Horizontal		✓
Vertical		✓

- Image Resize Testing

The process of resizing a picture is known as stego-image compression or compacting. This testing is conducted by applying compression percentage values to one of the stego-images named "green.jpg". The results of the robustness manipulation test for resizing the image are presented in Table VIII below.

TABLE VIII
RESIZE IMAGE TEST RESULTS

Resize Percentage	Pixel Dimension Change	Extraction Results
25	384 x 388	✓
50	256 x 256	✓
75	128 x 128	✓
90	51 x 51	✓

By compressing the image from 25% to 90%, manipulation testing of image resizing was carried out. Based on the observations and experiments conducted, the original message can be fully extracted even though the image has been compressed up to 90%.

- Color Contrast Effect Giving Test

In this test, contrast effects were applied to the stego-image, causing a change in color using the Microsoft Office Picture Manager application. Subsequently, the stego-image with the applied contrast effect was saved as a new stego-image to test whether the message could be extracted or not. The stego-image tested was named "foto.jpg." The results of the robustness manipulation test through the application of color contrast effects are presented in Table IX.

Amount and hue are used to adjust color shifts, while saturation controls the sharpness and fading of colors in the image. In the testing process presented in Table IX, the maximum values were applied: amount = 100, hue = 180, and saturation = 100. The message can still be fully extracted based on the analysis and observations.

TABLE IX
COLOR CONTRAST EFFECT TEST RESULTS

Amount	Hue	Saturation	Manipulation Results	Message Extraction
100	0	0		✓
0	180	0		✓
0	0	100		✓
100	180	100		✓

- Message Robustness Testing on Metadata

In this testing phase, attacks were carried out on the message within the metadata using several tools/applications commonly used for image editing. The manipulation involved modifying the stego-image in ways that could alter or damage the message stored in the metadata. The results of the message's resilience in the metadata are presented in Table X.

TABLE X
MESSAGE DURABILITY TEST RESULTS ON METADATA

Tools	Message Extraction
Microsoft Picture Manager	✓
Paint	✓
CorelDraw	✓
Paint 3D	✗
Windows Photo Editor	✗
Adobe Photoshop	✗

It is clear from the test results displayed in Table X that stego-image manipulation with Paint 3D, Windows Photo Editor, and Adobe Photoshop does not allow the message to be retrieved. These applications can change the program name attribute into the label of the application used.

C. *Evaluation of Test Results*

The method of message embedding utilizes exif image metadata and a combination of Fernet cryptography to execute the message embedding process in the image metadata space within the program name attribute based on the tests and testing carried out. While maintaining the goal of creating high-quality stego-images, the development of this approach concentrates on tackling robustness challenges, such as cropping, rotation, scaling, and color impacts on the stego-image.

The MSE values for every stego-image show no difference between the cover-image and the stego-image based on the fidelity testing study. In the meantime, all of the stego-pictures produced by this method's PSNR values for the five sample images used come out to be 100 dB. This is because there is no change in pixel intensity values, and the bit arrangement remains unaltered. This development method surpasses the previous techniques conducted by the author and also produces better image quality than several benchmark studies[19], [20], [40]–[42].

Furthermore, the analysis results from experiments on robustness processes for cropping, rotation, resizing, and applying color effects to images demonstrate that message embedding in exif metadata allows for successful message extraction even when the stego-image undergoes cropping manipulation, both symmetrically and asymmetrically, reducing the stego image dimensions to 1 x 1. Similarly, testing with rotation manipulation shows that the original message can still be extracted even when the stego-image is rotated up to 180 degrees. Testing with image resizing manipulation also proves that the message can be retrieved even when the stego-image undergoes up to 90% compression.

The testing of color effects applied to the image also does not affect message extraction results. According to the study observations and comparisons with previous studies, it is evident that earlier studies focused on exif image metadata without emphasizing robustness but solely focused on steganography image quality. Therefore, the fundamental difference in this study is the focus on robustness testing while still considering fidelity aspects in achieving high-quality stego-images.

One limitation of this SIM method is that the maximum size of the hidden text message in the metadata is limited to 8,000 characters. This method is only suitable when embedding relatively small messages. Another limitation is that if the stego image is manipulated using specific tools like Paint 3D, Windows Photo Editor, and Adobe Photoshop, the message cannot be extracted because the "program name" attribute will automatically be replaced with the name label of the respective application.

IV. CONCLUSION

The approach devised in this work focuses on how resilient stego-images are to changes that affect robustness. The strategy makes use of Fernet cryptography and the exif image metadata mechanism. The Steganography on Image Metadata (SIM) technique uses pre-existing fields, such as program name, to embed a concealed message within the metadata of an image file. With this technique, a maximum PSNR value of 100 dB and an outstanding MSE value of 0 can be attained. These numbers show that the steganographic image generated by the SIM technique has exceptional quality.

All robustness manipulation issues in steganography can be addressed using the SIM method. The results of the tests show that the message can still be retrieved using the SIM approach even after symmetric and asymmetric cropping adjustments down to a pixel size of 1x1. Testing with image rotation manipulation also proves that the message can be successfully extracted even when the stego-image is rotated up to 180 degrees. Experiments with image resizing manipulation also confirm that the message can be recovered even when the stego-image is compressed up to 90%. Testing with color effects applied to the image also does not affect message extraction results.

A limitation of SIM is that the maximum size of the hidden text message in the metadata is 8,000 characters. This method is only suitable for embedding relatively small messages. Compressing the text message before embedding it into the cover image's metadata is recommended for future study development to increase the message storage capacity. Another limitation is that if the stego image is manipulated using specific tools like Paint 3D, Windows Photo Editor, and Adobe Photoshop, the message cannot be extracted because the "program name" attribute is automatically replaced with the application's label name. To address this limitation, we propose an improved method by not only hiding the message in the "program name" attribute but also allowing it to be stored in other attributes.

Acknowledgment

We express our gratitude to the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia for providing funding for this project under contract number 178/E5/PG.02.00.PL/2023, 234/LL2/AL.04/2023.

References

[1] O. F. A. Wahab, A. A. M. Khalaf, A. I. Hussein, and H. F. A. Hamed, "Hiding Data Using Efficient Combination of RSA Cryptography, and Compression Steganography Techniques," IEEE Access, vol. 9, pp. 31805–31815, 2021, doi: 10.1109/access.2021.3060317.

[2] Z.-L. Yang, S.-Y. Zhang, Y.-T. Hu, Z.-W. Hu, and Y.-F. Huang, "VAE-Stega: Linguistic Steganography Based on Variational Auto-Encoder," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 880–895, 2021, doi: 10.1109/tifs.2020.3023279.

[3] W. Tang, B. Li, M. Barni, J. Li, and J. Huang, "An Automatic Cost Learning Framework for Image Steganography Using Deep Reinforcement Learning," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 952–967, 2021, doi:10.1109/tifs.2020.3025438.

[4] N. Subramanian, O. Elharrouss, S. Al-Maadeed, and A. Bouridane, "Image Steganography: A Review of the Recent Advances," IEEE Access, vol. 9, pp. 23409–23423, 2021, doi:10.1109/access.2021.3053998.

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

[6] 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, doi:10.1007/s11128-021-02991-8.

[7] X. Liao, J. Yin, M. Chen, and Z. Qin, "Adaptive Payload Distribution in Multiple Images Steganography Based on Image Texture Features," IEEE Transactions on Dependable and Secure Computing, pp. 1–1, 2021, doi: 10.1109/tdsc.2020.3004708.

[8] 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, doi:10.1109/access.2020.2971528.

[9] S. Li, M. Xue, B. Zhao, H. Zhu, and X. Zhang, "Invisible Backdoor Attacks on Deep Neural Networks via Steganography and Regularization," IEEE Transactions on Dependable and Secure Computing, pp. 1–1, 2020, doi: 10.1109/tdsc.2020.3021407.

[10] A. A. AlSabhany, A. H. Ali, F. Ridzuan, A. H. Azni, and M. R. Mokhtar, "Digital audio steganography: Systematic review, classification, and analysis of the current state of the art," Computer Science Review, vol. 38, p. 100316, Nov. 2020, doi:10.1016/j.cosrev.2020.100316.

[11] M. Dalal and M. Juneja, "Steganography and Steganalysis (in digital forensics): a Cybersecurity guide," Multimedia Tools and Applications, vol. 80, no. 4, pp. 5723–5771, Oct. 2020, doi:10.1007/s11042-020-09929-9.

[12] M. Hassaballah, M. A. Hameed, A. I. Awad, and K. Muhammad, "A Novel Image Steganography Method for Industrial Internet of Things Security," IEEE Transactions on Industrial Informatics, vol. 17, no. 11, pp. 7743–7751, Nov. 2021, doi: 10.1109/tii.2021.3053595.

[13] S. Dhawan, C. Chakraborty, J. Frnda, R. Gupta, A. K. Rana, and S. K. Pani, "SSII: Secured and High-Quality Steganography Using Intelligent Hybrid Optimization Algorithms for IoT," IEEE Access, vol. 9, pp. 87563–87578, 2021, doi: 10.1109/access.2021.3089357.

[14] Y. Zhao, R. Wang, W. Jia, W. Zuo, X. Liu, and W. Gao, "Deep Reconstruction of Least Significant Bits for Bit-Depth Expansion," IEEE Transactions on Image Processing, vol. 28, no. 6, pp. 2847–2859, Jun. 2019, doi: 10.1109/tip.2019.2891131.

[15] C.-C. Chang, "Adversarial Learning for Invertible Steganography," IEEE Access, vol. 8, pp. 198425–198435, 2020, doi:10.1109/access.2020.3034936.

[16] A. A. Abd EL-Latif, B. Abd-El-Atty, and S. E. Venegas-Andraca, "A novel image steganography technique based on quantum substitution boxes," Optics &amp; Laser Technology, vol. 116, pp. 92–102, Aug. 2019, doi: 10.1016/j.optlastec.2019.03.005.

[17] M. Kalita, T. Tuithung, and S. Majumder, "A New Steganography Method Using Integer Wavelet Transform and Least Significant Bit Substitution," The Computer Journal, vol. 62, no. 11, pp. 1639–1655, Mar. 2019, doi: 10.1093/comjnl/bxz014.

[18] L. AlFaqawi, M. AbuHaya, and ..., "Alpha channel-Based Indicator For Robustness Forward/Backward LSB Steganography," ... *Conference on Information* ..., 2021, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9636891/

[19] Ms. B. Bhatu and H. Y. Shah, "Customized Approach To Increase Capacity And Robustness In Image Steganography," in *International Conference on Inventive Computation Technologies (ICICT)*, 2016.

[20] 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, 2017.

[21] L. Ke and Z. Yin, "On the security and robustness of 'Keyless dynamic optimal multi-bit image steganography using energetic pixels,'" Multimedia Tools and Applications, vol. 80, no. 3, pp. 3997–4005, Sep. 2020, doi: 10.1007/s11042-020-09807-4.

[22] S. K. Moon, "Software and hardware-based audio-video crypto steganalysis model for enhancing robustness and imperceptibility of secured data," Multimedia Tools and Applications, vol. 81, no. 15, pp. 21047–21081, Mar. 2022, doi: 10.1007/s11042-022-12353-w.

[23] D. Darwis, N. B. Pamungkas, and Wamiliana, "Comparison of Least Significant Bit, Pixel Value Differencing, and Modulus Function on Steganography to Measure Image Quality, Storage Capacity, and Robustness," Journal of Physics: Conference Series, vol. 1751, no. 1, p. 012039, Jan. 2021, doi: 10.1088/1742-6596/1751/1/012039.

[24] D. Darwis, A. Junaidi, D. A. Shofiana, and Wamiliana, "A New Digital Image Steganography Based on Center Embedded Pixel Positioning," Cybernetics and Information Technologies, vol. 21, no. 2, pp. 89–104, Jun. 2021, doi: 10.2478/cait-2021-0021.

[25] 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," European Interdisciplinary Cybersecurity Conference, Nov. 2021, doi: 10.1145/3487405.3487661.

[26] C. M. Kondasinghe, "A System to Preserve Metadata using Steganography." 2021.

[27] D. Darwis, A. Junaidi, and Wamiliana, "A New Approach of Steganography Using Center Sequential Technique," Journal of Physics: Conference Series, vol. 1338, no. 1, p. 012063, Oct. 2019, doi: 10.1088/1742-6596/1338/1/012063.

[28] "Introduction to Cryptography and Python," Implementing Cryptography Using Python®, pp. 1–30, Jul. 2020, doi:10.1002/9781119612216.ch1.

[29] R. Rahaeimehr, "Novel Cryptographic Authentication Mechanisms for Supply Chains and OpenStack Novel Cryptographic Authentication Mechanisms for Supply Chains and OpenStack," 2019.

[30] M. Suryadi, Y. Satria, and A. Hadidulqawi, "Implementation of the Gauss-Circle Map for encrypting and embedding simultaneously on digital image and digital text," Journal of Physics: Conference Series, vol. 1821, no. 1, p. 012037, Mar. 2021, doi: 10.1088/1742-6596/1821/1/012037.

[31] R. Wazirali, Z. Chaczko, and J. Gibbon, "Steganographic Image Sharing App," 2017 25th International Conference on Systems Engineering (ICSEng), Aug. 2017, doi: 10.1109/icseng.2017.62.

[32] D. Darwis, A. Junaidi, and Wamiliana, "A New Approach of Steganography Using Center Sequential Technique," Journal of Physics: Conference Series, vol. 1338, no. 1, p. 012063, Oct. 2019, doi: 10.1088/1742-6596/1338/1/012063.

[33] O. Juarez-Sandoval, M. Cedillo-Hernandez, G. Sanchez-Perez, K. Toscano-Medina, H. Perez-Meana, and M. Nakano-Miyatake, "Compact image steganalysis for LSB-matching steganography,"

2017 5th International Workshop on Biometrics and Forensics (IWBF), Apr. 2017, doi: 10.1109/iwbf.2017.7935103.

[34] S. D. Muyco and A. A. Hernandez, "A Modified Hash Based Least Significant Bits Algorithm for Steganography," Proceedings of the 2019 4th International Conference on Big Data and Computing - ICBDC 2019, 2019, doi: 10.1145/3335484.3335514.

[35] 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, Nov. 2020, doi: 10.1007/s11042-020-10035-z.

[36] S. D. Muyco and A. A. Hernandez, "Least Significant Bit Hash Algorithm for Digital Image Watermarking Authentication," Proceedings of the 2019 5th International Conference on Computing and Artificial Intelligence, Apr. 2019, doi: 10.1145/3330482.3330523.

[37] S. Dhawan and R. Gupta, "Analysis of various data security techniques of steganography: A survey," Information Security Journal: A Global Perspective, vol. 30, no. 2, pp. 63–87, Aug. 2020, doi:10.1080/19393555.2020.1801911.

[38] M. S. Taha, M. S. Mohd Rahim, S. A. Lafta, M. M. Hashim, and H. M. Alzuabidi, "Combination of Steganography and Cryptography: A short Survey," IOP Conference Series: Materials Science and Engineering, vol. 518, no. 5, p. 052003, May 2019, doi: 10.1088/1757-899x/518/5/052003.

[39] O. Evsutin, A. Melman, and R. Meshcheryakov, "Digital Steganography and Watermarking for Digital Images: A Review of Current Research Directions," IEEE Access, vol. 8, pp. 166589–166611, 2020, doi: 10.1109/access.2020.302277979.

[40] G. Swain and S. K. Lenka, "A novel steganography technique by mapping words with LSB array," International Journal of Signal and Imaging Systems Engineering, vol. 8, no. 1/2, p. 115, 2015, doi: 10.1504/ijsise.2015.067052.

[41] O. Juarez-Sandoval, A. Fierro-Radilla, A. Espejel-Trujillo, M. Nakano-Miyatake, and H. Perez-Meana, "Cropping and noise resilient steganography algorithm using secret image sharing," Sixth International Conference on Graphic and Image Processing (ICGIP 2014), Mar. 2015, doi: 10.1117/12.2179745.

[42] M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. S. Ho, and K.-H. Jung, "Image steganography in spatial domain: A survey," Signal Processing: Image Communication, vol. 65, pp. 46–66, Jul. 2018, doi:10.1016/j.image.2018.03.012.