Steganography Method of the Bigger Size in WebP Image Using M2PAM Algorithm for Social Applications

Omar A. Najm[a], Ahmed S. Nori[b]

University of Mosul (UOM),College of Computer Science and Mathematics,

Department of Computer Science, omar.csp55@student.uomosul.edu.iq,ahmed.s.nori@uomosul.edu.iq

Article History: *Submitted: 1 march 2022; Revised: 29 march 2022; Accepted: 4 April 2022; publication: 13 May 2022*

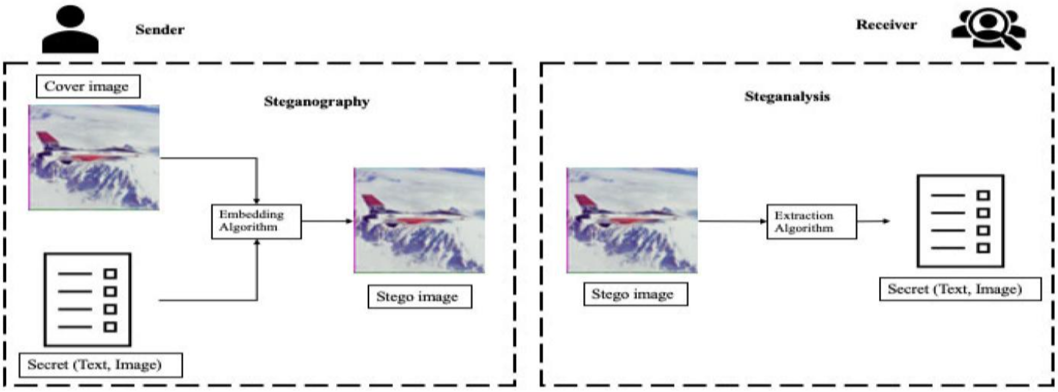
Abstract: Many facets of our lives, including business, education, and social activities, have been dominated by social media. Users of social media exchange a tremendous amount of information. Hackers have unauthorized access to this information. One of the most effective methods for data protection is steganography. Although there are many steganography algorithms in the literature, there is still a gap in terms of adopting high-efficiency techniques. This paper provides a steganography approach for secret data based on the M2PAM algorithm. WebP photos, which are commonly utilized in social media applications, and are from digital media to hide the messages The proposed method is based on the M2PAM algorithm and WebP images that are widely used in social media applications as cover to hide the messages. The proposed method is benchmarked against other methods in the literature. the proposed method showed outperformed the benchmarking in terms results of two metrics PNSR and MSE.

Keyword: Steganography, Security, WebP Image, Social Media

1. Introduction

Information security has become a core need in the current technological age and the popularity of applications **(Chinnasamy, P., Padmavathi, S., Swathy, R., & Rakesh, S. 2021)**. There are many different approaches for protecting data from malicious users to access private or sensitive information **(Islam, M. A., Kobita, A. A., Hossen, M. S., Rumi, L. S., Karim, R., & Tabassum, T. 2021)**. Also, data security becomes a significant concern for essential and thoughtful information **(GENÇOĞLU, M. T. (2021))(Ikhwan, A., Raof, R. A. A., Ehkan, P., Yacob, Y., & Syaifuddin, M. 2021)**. Practically, there are two procedures for data security namely, cryptography and steganography. The former is widely used to convert secret data into unreadable data by encoding it using keys. Besides, many algorithms were developed by cryptographers to protect users' data. However, the cryptography technique has disadvantages in that encrypted data is suspected if it is detected by malicious users **(Rout, H., & Mishra, B. K. 2014)**. The cryptography technique does not offer a satisfactory level of security that is required by some systems **(Panhwar, M. A., Khuhro, S. A., Mazhar, T., ZhongLiang, D., & Qadir, N. 2021)**. On the other hand, the latter, steganography technique is gathering secret data inside protocols or multimedia files as a cover (e.g., images, video, or sound files). In steganography, the malicious users will be able to view the cover file but it is not an easy task to retrieve. readable data. The previous practices showed that data security can be better performed using image encryption algorithms, then, a steganography approach inserts information into media consideration **(Oad, A., Yadav, H., & Jain, A. 2014)**. This permits steganography to produce data covered to enable other proposed users to distinguish the existence of information. The description of the working of steganography and its analysis is shown in Figure 1.

Figure.1. A general steganography steps.



Furthermore, many aspects should be taken into considerations when applying a steganography method. These aspects are represented indicators of the quality of the method used. The main indicators are presented as follows:

- The capacity hiding: It is always preferable to adopt approaches that lead to a high hiding capacity. The traditional approaches underperformed all the newly developed approaches when it comes to capacity hiding, followed by the Generative Adversarial Networks (GANs). The Convolutional Neural Networks (CNNs) approaches are considered the most efficient approaches in terms of hiding capacity according to **(Zhang, R., Dong, S., & Liu, J. 2019).**

- Robustness and Security: In fact, robustness is related to extracting the secret image while security is related to the process of embedding. These indicators are considered crucial factors in steganography since they relate to the core concept of the area **(Kuppusamy, P. G., Ramya, K. C., Rani, S. S., Sivaram, M., & Dhasarathan, V. 2020) (Subramanian, N., Elharrouss, O., Al-Maadeed, S., & Bouridane, A. 2021)**.

- Temper Resistance: altering a message is not an easy task after embedding it into an image (stegoimage) **(Rahmani I., Kumar A., & Manisha G. 2015).**

- Complexity: It is always preferred to adopt a simple approached that does not include complex computations. The cost of an approach is considered one of the important factors that developers try to minimize **(Jain, R., & Boaddh, J. 2016).**

Finally, the working domains when applying steganography techniques can be categorized into the following:

- Spatial Domain Approaches: These approaches perform some changes in the values of image pixels when hiding information.
- Transform Domain Approaches: They are more complex in implementation and many algorithms have been proposed that use transform domain (e.g., embedding in the frequency domain).
- Distortion Approaches: This kind of approach required knowledge about the “cover image” during the decoding process.
- Filtering and Masking Approaches: These approaches hide information using the same concept that is used in watermarking. The information is stored in the most significant areas

2.Significance Of The Study

According to the literature, many issues have been observed regarding three terms, namely, robustness, hiding capacity, and complexity. Most of the approaches in the literature struggled with the aforementioned issues. Hence, the contribution of this work is developing a method that is based on the M2PAM algorithm for hiding secret data in a WebP image that is widely used in social media. The proposed method is considered simple, robust, and secure more hiding capacity.

This article is organized as follows: Section 2 describes the proposed method at the receiver and sender sides. Section 3 presents the experimental results and the discussions. Section 4 concludes this work.

3. Literature Review

The traditional approaches of steganography have struggled with the security issue (e.g., the secret information is represented as a text). A summary of the traditional approaches is presented in Table 1.

The literature of steganography is rich with distinguished works. Many approaches have been proposed using a variety of algorithms and methods. **Thenmozhi and Chandrasekaran Thenmozhi, S., & Chandrasekaran, M. (2013).** The steganography technique to hide the secret messages within WebP images format by using the proposed algorithm named Mod 8 Plus Average Method (M8PAM). The proposed algorithm hides every three bits in one pixel of the cover file **Mahmood B.(2017).** proposed an implementation of a novel technique that was based on Discrete Wavelet Transform (DWT) for transforming original image steganography (covered) from a spatial domain to frequency domain. Thus, using (2-D DWT) on the cover image was improved, and using DWT is efficient but with a lowfrequency sub-band. Moreover, in the work of **Bandyopadhyay, D., Dasgupta, K., Mandal, J. K., & Dutta, P. (2014).** A secure data hiding technique for digital images based on the Least Significant Bit (LSB) technique was built. Hosting image files to hide sensitive information in the spatial domain including the LSB insertion method for image steganography was used. Therefore, image values and pinnacle of Signal-to-Noise Ratio (PSNR) presented good results. Besides, **Yu, X., Wang, C., & Zhou, X. (2018).** The issue of combining image steganography with pre-processing of Data

Encryption Standard (DES) encryption and LSB steganography algorithms was addressed. Their survey showed studies that using image steganography with pre-processing of DES encryption was better than using LSB steganography algorithms directly.

Moreover, another study performed by **Manjula, G. R., & Danti, A. (2015).** Suggest a way to hide a secrecy color photo in a color lid photo. The proposed technique takes data the that to be hidden consist of eight bits and sets them in the LSB of red, green and blue (RGB) pixel values of the cover image separately. Also, five (5) bits are placed in pixel G and R, and three (3) bits are placed in pixel B. Therefore, this technique gives better results compared to methods 3,3,2. In the same context. **Bawaneh, M. J., & Obeidat, A. A. (2016)** .Proposed a novel approach for data security called the Greyscale Steganography Process. The main idea behind the approach was using image segmentation for inserting secret message bit in LSB of a random pixel within the greyscale cover image. **Hussain, M., Wahab, A. W. A., Ho, A. T., Javed, N., & Jung, K. H. (2017).** Represented a new data hiding method for increasing visual quality, payload, and sustains steganography security. The main structure was consisting two methods, such as Parity-Bit Pixel Value Difference (PBPVD) and improved Right Most Digit Replacement (iRMDR). The iRMDR method presented the best closest stego-pixels for good visual quiet.

Table.1. Examples of traditional approaches

Reference	Data Used	Measurement	Limitations	Strengths
Al-Afandy et al.	Single RGB Images	Peak Signal to Noise Ratio (PSNR), Time	The text representation of secret info.	Fast and Robust
Arya, A., & Soni, S.	Lena and Baboon	Peak Signal to Noise Ratio (PSNR) and Mean Squared Error (MSE)	Less secure	Fast and accepts arbitrary image format
Patel, N., & Meena, S.	Lena	Peak Signal to Noise Ratio (PSNR)	Less secure	Fast

4.Objectives Of The Study

The main objective of this thesis is to design and implement a system for securely disguising WebP format images as cover images based on the M2PAM algorithm within social media applications. This was done using encryption and masking techniques with the mentioned algorithm. With these methods and techniques, the data will be transmitted in a secure manner so that no interceptor can detect, decrypt or tamper with it.

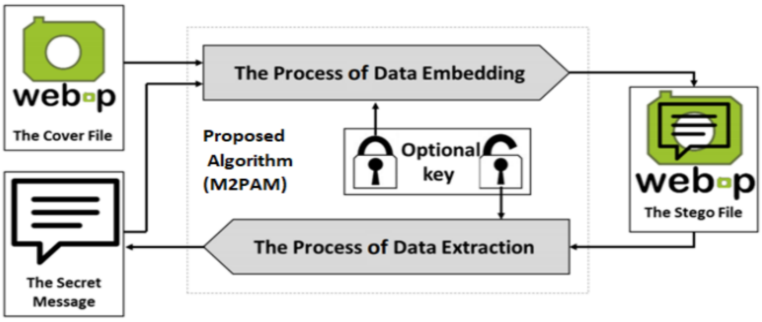
.RESEARCH METHOD

In this section, the proposed method using the M2PAM algorithm is described in three layers. In the first layer, non-sequential locations in the cover file are selected. These locations will be used to embed the secret data. In the second layer, the secret data is registered by changing its sequence (bits) using a random function. Besides, in this layer, the secret data is embedded and extracted using a symmetric key from both sides (sender and receiver). The third layer is used to hide the secret data using a public key in the sending process and a private key in the receiving process.

Mod 2 Plus Average Method (M2PAM)

The proposed cryptography method is described in **Figure.2.**

Figure.2. A diagram describes the proposed cryptography method.

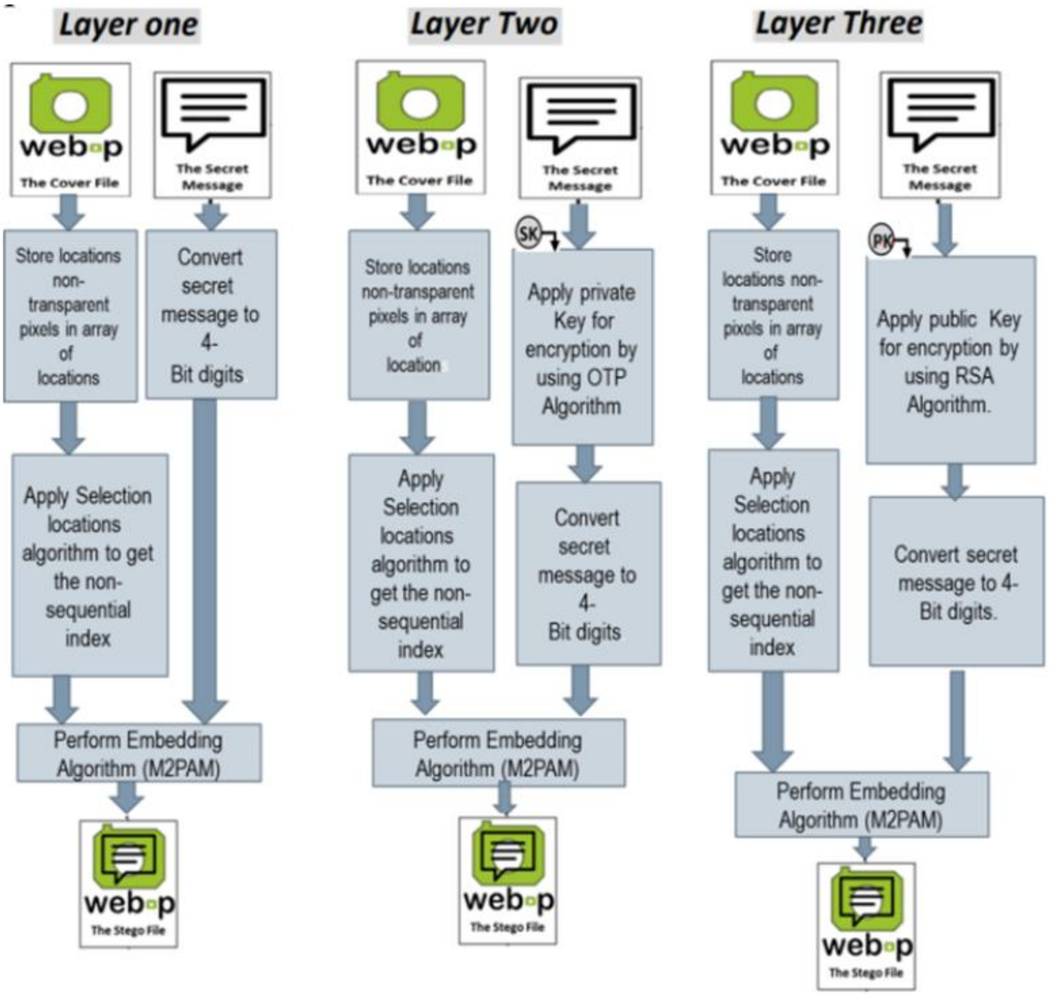

webp
The Cover File
The Process of Data Embedding
Proposed Algorithm (M2PAM)
Optional key
webp
The Stego File
The Process of Data Extraction
The Secret Message

According to the above figure, the proposed method can be summarized as follows:

- The proposed method works in two directions, embedding the secret data and extracting the data.
- Secret data: represents the data that should be protected from unauthorized access.
- Cover file: it is the file that will be used to carry the secret data. These files also will not be affected by the embedding process and maintain its resolution.
- Keys: they are used to protect the cryptography process. Key can be symmetric or asymmetric (public and private keys).
- the output of the embedding process :Stego file (stego file is the cover file before the embedding process).

Practically, data that hidden, cover file, and keys are considered the input of the embedding process. While stego file and keys are considered the inputs of the extraction process. **Figure.3.** demonstrates the embedding process.

Figure.3. A diagram describing the proposed method of embedding in the steganography system.


Layer one
Layer Two
Layer Three
webp
The Cover File
The Secret Message
Store locations non-transparent pixels in array of locations
Convert secret message to 4-Bit digits
Apply Selection locations algorithm to get the non-sequential index
Perform Embedding Algorithm (M2PAM)
The Stego File
SK
Apply private Key for encryption by using OTP Algorithm
PK
Apply public Key for encryption by using RSA Algorithm.
Convert secret message to 4-Bit digits.

The proposed method is based on partitioning the data into 4 bits. At the same time, the none transparent pixels of the cover file is determined that, in turn, will carry the bit values of the secret data. The detailed steps of the embedding method are described in **Algorithm.1.**

Algorithm.1. The proposed embedding process in the cover file.

Goal: to hide data into a cover file, in our case the cover file is an image of type lossless webP.

Input: the data to hide in the image as an array of bits, in our case the data is text, and the webP image to hide the data in as RGBA (8888) array.

Output: output: the input webP image after hiding the data in it as RGBA (8888) array.

Steps:

Step1: allocate the index of non-transparent pixels in the RGBA array.

Step2: apply the random selection algorithm on the non-transparent pixels.

Step3: grouping the input text bits into groups of 4 bits.

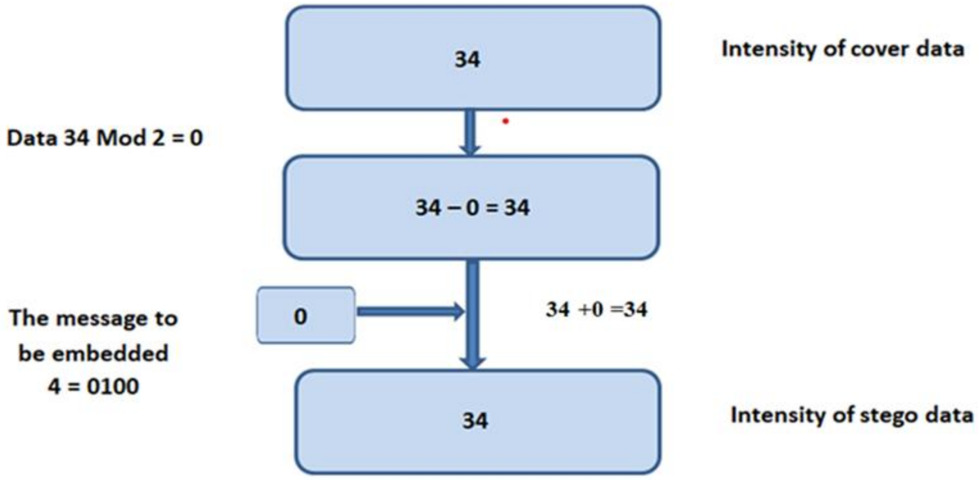
Step4: hide each 4 bits in one of the random selected non-transparent pixel each bit in one layer of the RGBA as follow: 1.find the modules of 2 on the R layer. 2.subtract the mod2 from the R layer. 3.add the result of the subtraction to the original value of R layer. 4.repeat the above 3 steps for the GBA layers to one bit in each.

Step5: Convert the new RGBA (8888) to the webP image to get the stegano file.

Step6: end.

In step 2, the value of 2 enables to embed one bit of the secret data in a location that was selected in the cover file see **Figure.4.**

Figure.4. Illustration of the embedding process using M2PAM.



On the other hand, the extraction process in the proposed method is demonstrated in **Figure.5.** The extraction process allows retrieving one bit of the secret data from a location in the stego file through dividing by 2, then subtracting the change in the remainder of the division operation. After retrieving all the bit values of the secret data, they are converted into a matrix of bytes and then converted into the understandable form of the secret data.

Algorithm 2: The process extractionof secret data in using proposed method the proposed steganography system.

Goal: extract the hidden data from the stegano file webp image.

inputs: stegano file webp image as an array of RGBA 8888.

Output: the secret message is hidden in the stegano file.

Steps:

Step1: allocate the index of non-transparent pixels in the RGBA array.

Step2: apply the random selection algorithm on the non-transparent pixels.
Step3: extract 4 bits from each pixel (one bit from each layer RGBA) by applying the modulus of 2.
Step4: Repeat the previous step to extract all the hidden data as an array of bits
Step5: Convert the extracted array of bits to bytes each 8 bits represent the ASCII code of the text hidden in the webp image.
Step6: end.

Figure.5. A diagram describing the proposed method of extraction in the steganography system.

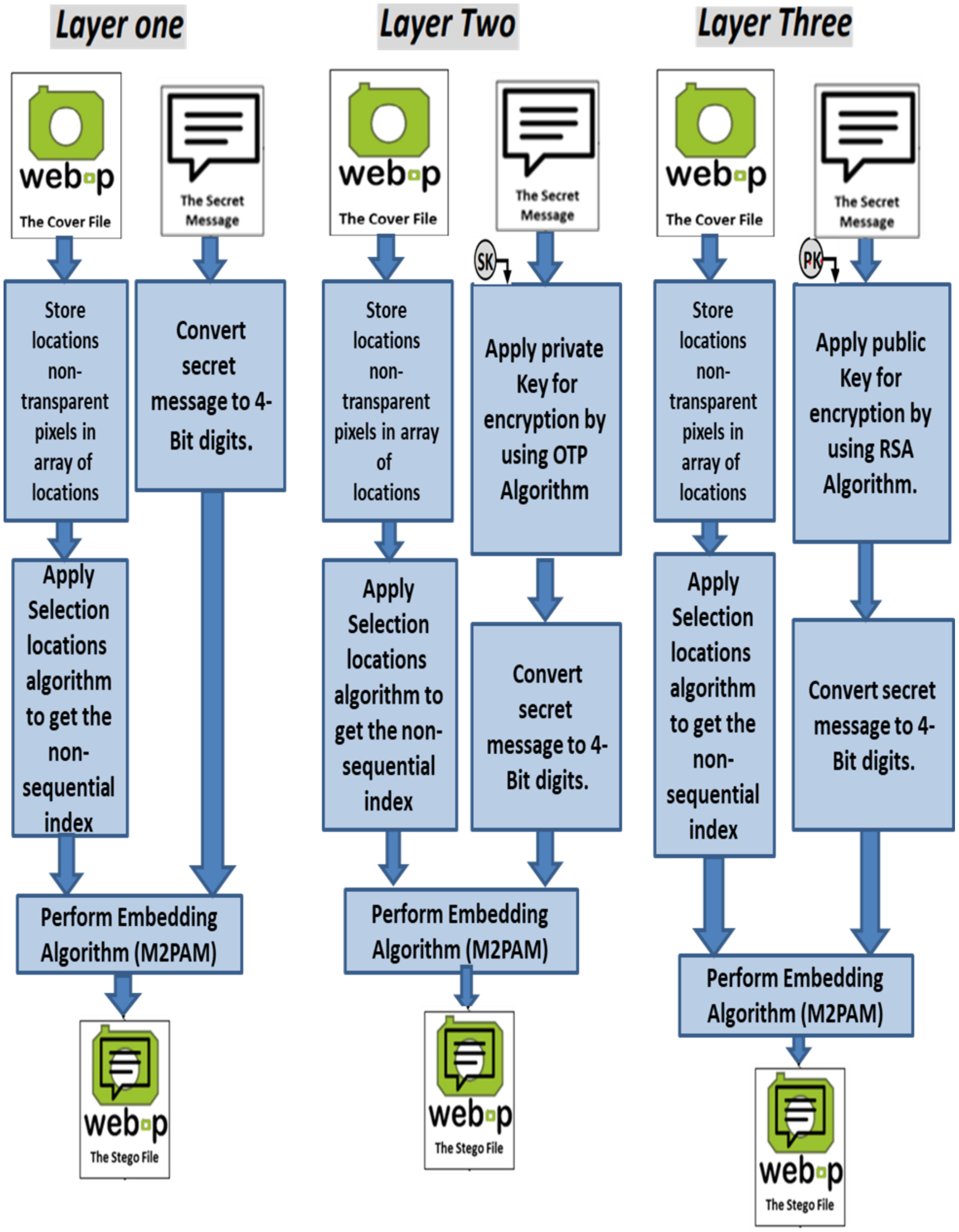
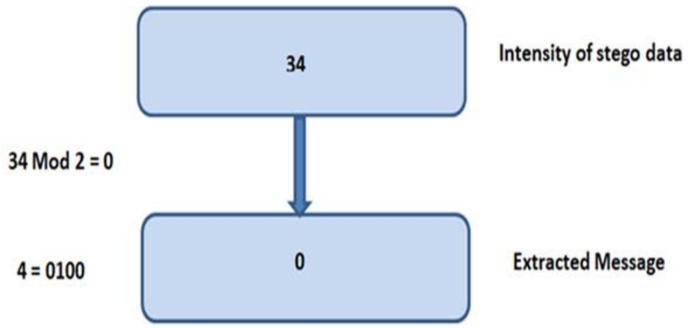

Layer one
Layer Two
Layer Three
webp
The Cover File
The Secret Message
Store locations non-transparent pixels in array of locations
Convert secret message to 4-Bit digits.
Apply Selection locations algorithm to get the non-sequential index
Perform Embedding Algorithm (M2PAM)
The Stego File
SK
Apply private Key for encryption by using OTP Algorithm
Apply Selection locations algorithm to get the non-sequential index
PK
Apply public Key for encryption by using RSA Algorithm.
Convert secret message to 4-Bit digits.

Figure.6. illustrates the extraction of the secret data from a stego file using Algorithm 2.

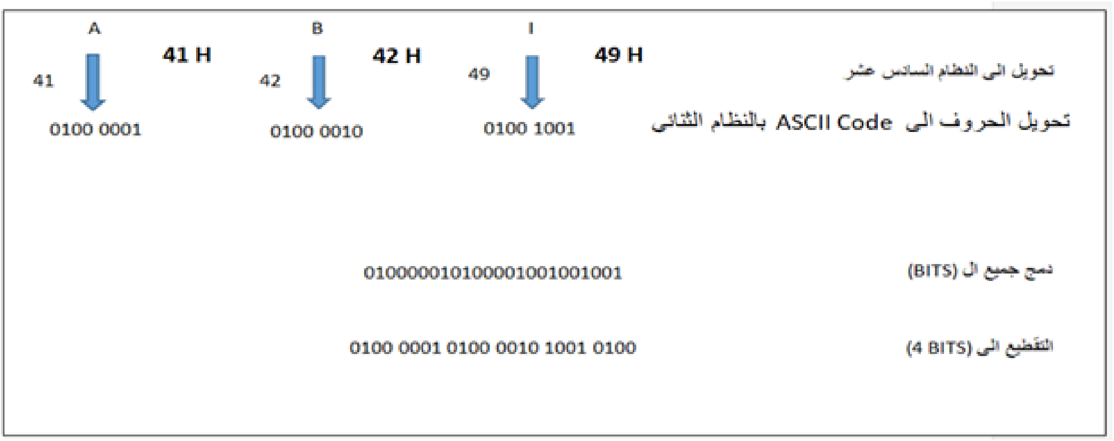
Figure.6. Illustration of the extraction process using M2PAM.



Secret Data Preprocessing

To illustrate the conversion of the secret data into bit values it is needed to perform some steps as shown in the following scenario: Assume having a secret message ("ABI") and it is needed to send it securely using the proposed M2PAM method. The first step is to convert the data into a byte form then to a bit form based on ASCII standard. Each letter is converted to its corresponding code that is in a hexadecimal form. In the example, the letter "A" corresponds to 41, the letter "B" corresponds to 42, the letter "I" corresponds to 49, and so on. Then, these numbers will be converted into a binary form. After that, the data will be partitioned into chunks of 4 bits.

Secret data is: ABI



After completing the aforementioned processes, a new matrix is generated that represents a stream of bits of the secret data. This matrix will be sent into the embedding process. Algorithm 3 illustrates the preprocessing steps of the secret data.

Algorithm 3: Secret data preprocessing.
Goal: convert the secret text to an array of bits to be able to hide it in the webp image. **Input:** secrecy message as text. **Output:** array of bits.
Steps:
Step 1: convert the input text to an array of ASCII code.
Step 2: add 6 special characters represent the tail of data.
Step3: convert each byte from the array of ASCII code to 8 bits and append it to the bits array.
Step4: End.

Selecting the cover letter

This work proposes the WebP image type to be used as a cover file. This kind of image is considered lossless and widely used in social media applications (e.g., stickers). Moreover, this type of image is widely supported by

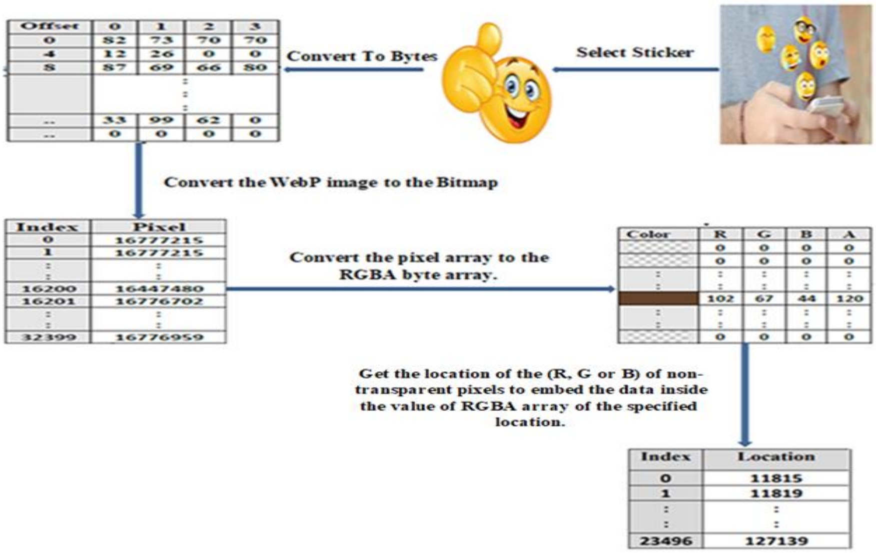
Google applications and can be used with a wide range of APIs and libraries. Besides, it can be handled using different programming languages such as C, C++ , and Java that are compatible with the Android operating system and its tools. Practically, each pixel in the image is taken in the form of 4 bits. Then, the WebP format is converted into a Bitmap image format. The next step is to make a backup copy and put it in temporary storage (buffer) aiming at having a copy before and after the change in the image. The none transparent pixels are detected and a random function is used to select the locations that will be used in the embedding process. Algorithm 4 illustrates the steps followed for selecting the locations that will be used in the embedding process.

Algorithm 4: Detecting image locations that secret data will be embedded in
Goal: select the non-transparent pixels to hide data in them. **Input:** cover file Webp image as an array of RGBA 8888. **Output:** outputs: indices of non-transparent pixels as array A.
Steps:
Step 1: create a pixel pointer start with index 0.
Step 2: select four values from the input array RGBA 8888 that pixel pointer point to, these four values represent the RGBA layer of each pixel.
Step3: compare the RGBA value with the threshold value.
Step4: if the RGBA values greater than threshold value adds the pixel pointer value to the A array.
Step 5: increase the pixel value by 4 to compare the next pixel.
Step 6: end

Figure.7. depicts the process of detecting the hiding locations in the cover image.

As mentioned before, there are three layers in the proposed steganography system. Each layer has its own parameters and can be associated with other layers. Some of the parameters can be defined by users or can be generated using particular algorithms. **Figure.8.** demonstrates the integration of the layers in the proposed steganography system.

Figure.7. Illustration of detecting the hiding locations in the cover image.



Convert To Bytes
Select Sticker
Convert the WebP image to the Bitmap
Convert the pixel array to the RGBA byte array.
Get the location of the (R, G or B) of non-transparent pixels to embed the data inside the value of RGBA array of the specified location.

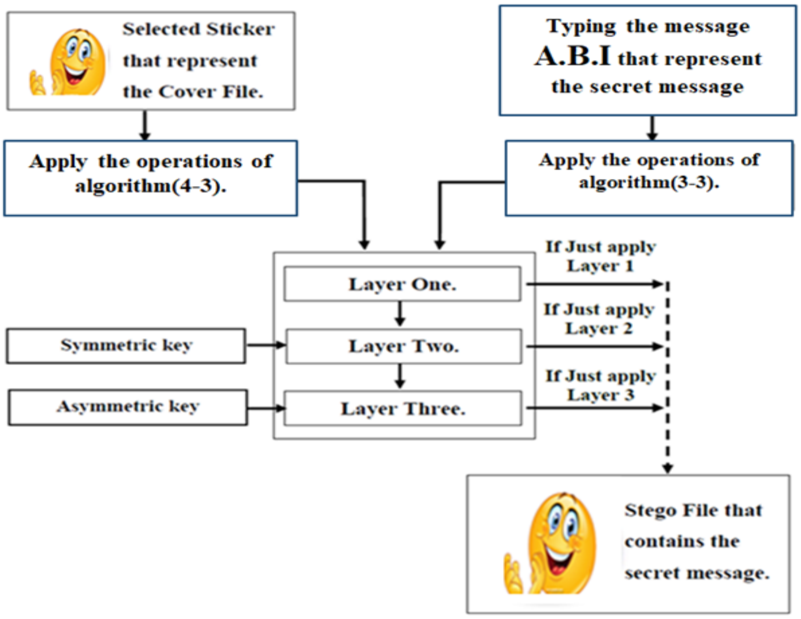
Offset	0	1	2	3
0	82	73	70	70
4	12	26	0	0
8	87	69	66	80
	:	:	:	:
--	33	99	62	0
--	0	0	0	0

Index	Pixel
0	16777215
1	16777215
: :	: :
16200	16447480
16201	16776702
: :	: :
32399	16776959

Color	R	G	B	A
▨	0	0	0	0
▨	0	0	0	0
: :	: :	: :	: :	: :
■	102	67	44	120
: :	: :	: :	: :	: :
▨	0	0	0	0

Index	Location
0	11815
1	11819
: :	: :
23496	127139

Figure.8. The integration of the three layers in the proposed steganography system.


Selected Sticker that represent the Cover File.
Typing the message A.B.I that represent the secret message
Apply the operations of algorithm(4-3).
Apply the operations of algorithm(3-3).
Layer One.
Layer Two.
Layer Three.
Symmetric key
Asymmetric key
If Just apply Layer 1
If Just apply Layer 2
If Just apply Layer 3
Stego File that contains the secret message.

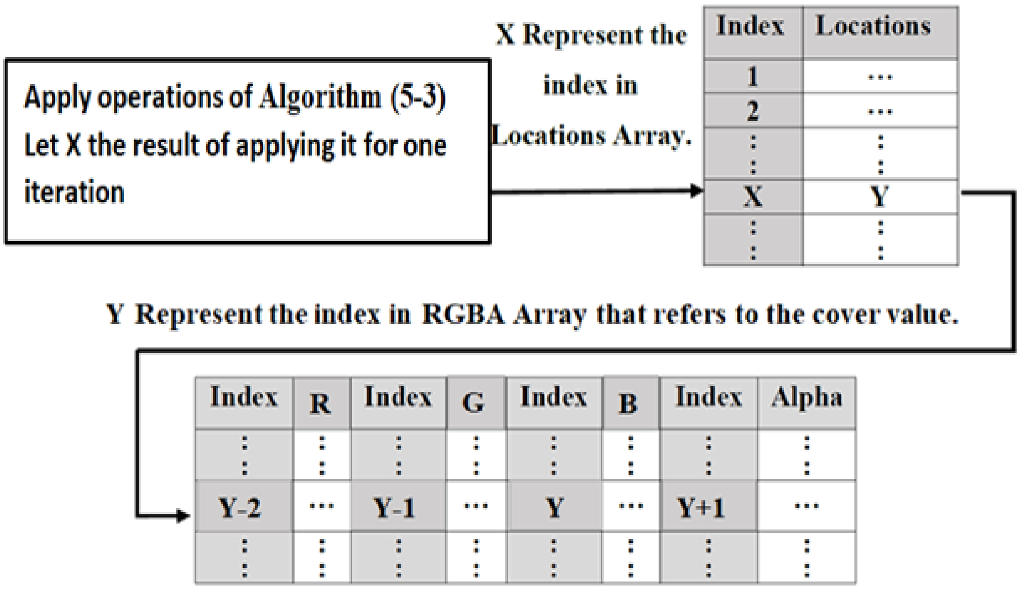
Embedding Process: Layer One

The embedding process that hides the secret data is applied on the predefined none-transparent locations in an un-sequential way in the cover file. After that, the values of the components (R, G, or B) are stored in addition to the alpha byte. The proposed method generates two matrices as follows:

- Location array: includes the indices X of the locations.
- RGBA Array: Includes the indices Y that contains the locations of pixel components that will carry secret data.

Figure.9. shows a diagram that shows the relations between the two matrices and Algorithm 5 describes the steps of generating the indices.

Figure.9. The relations between Locations Array and BGRA Array.


Apply operations of Algorithm (5-3)
Let X the result of applying it for one iteration
X Represent the index in Locations Array.
Y Represent the index in RGBA Array that refers to the cover value.

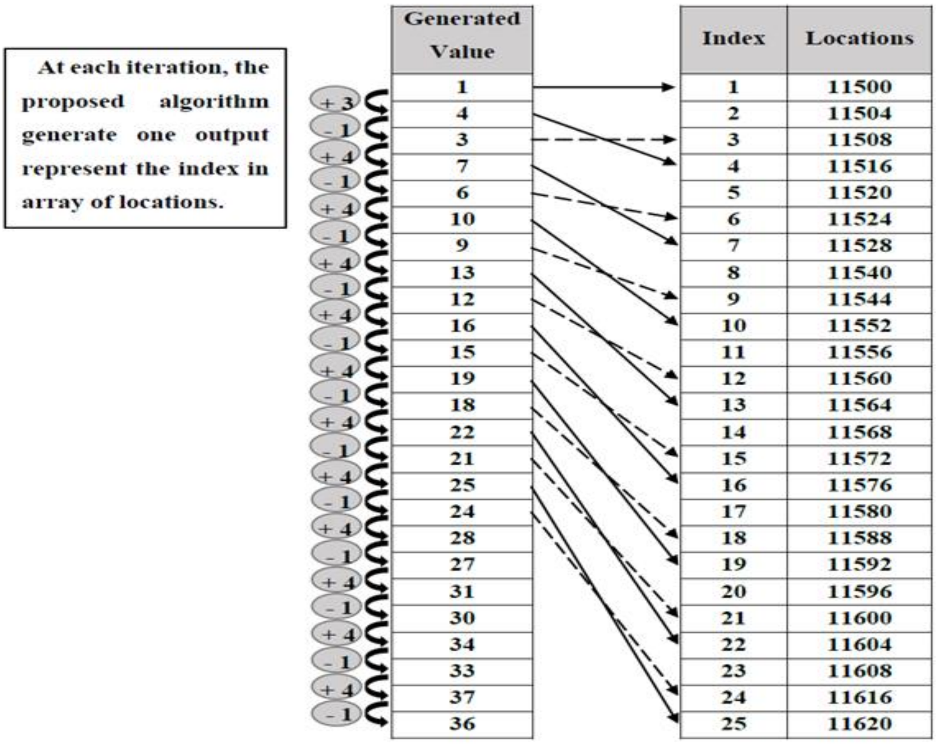
Index	Locations
1	…
2	…
⋮	⋮
X	Y
⋮	⋮

Index	R	Index	G	Index	B	Index	Alpha
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Y-2	…	Y-1	…	Y	…	Y+1	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

The process of generating indices is repeated until all the secret data is embedded (see the example in Figure 10).

Algorithm 5: Steps for generating the indices
Goal: chose the non-transparent pixels in non-series order. **Input:** indices of non-transparent pixels as array A. Output: **outputs:** indices of non-transparent pixels in non-series order as array B.
Steps:
Step 1: create pointer I = 0 points to indices of non-transparent pixels at array A.
Step 2: create variable V = 6.
Step3: add the index that I point at array A to array B.
Step4: apply equation I = I + V.
Step 5: if V modulus 6 equals to 0 apply equation V = V + 7.
Step 6: apply equation V = V – 6.
Step 7: apply equation V = V * (-1).
Step 8: Repeat the steps from 3 until we got the indices of non-transparent pixels in non-series order.
Step 9: end.

Figure.10. Example of generating un-sequential indices.



At each iteration, the proposed algorithm generate one output represent the index in array of locations.

Generated Value
1
4
3
7
6
10
9
13
12
16
15
19
18
22
21
25
24
28
27
31
30
34
33
37
36

Index	Locations
1	11500
2	11504
3	11508
4	11516
5	11520
6	11524
7	11528
8	11540
9	11544
10	11552
11	11556
12	11560
13	11564
14	11568
15	11572
16	11576
17	11580
18	11588
19	11592
20	11596
21	11600
22	11604
23	11608
24	11616
25	11620

Embedding Process: Layer Two

In this layer, several operations are applied during the embedding process

Embedding Process: Layer Three

This layer is used to add a data security algorithm to the system. The algorithm used in this layer is the RSA, which generates two keys; public and private see **Figure.11.**

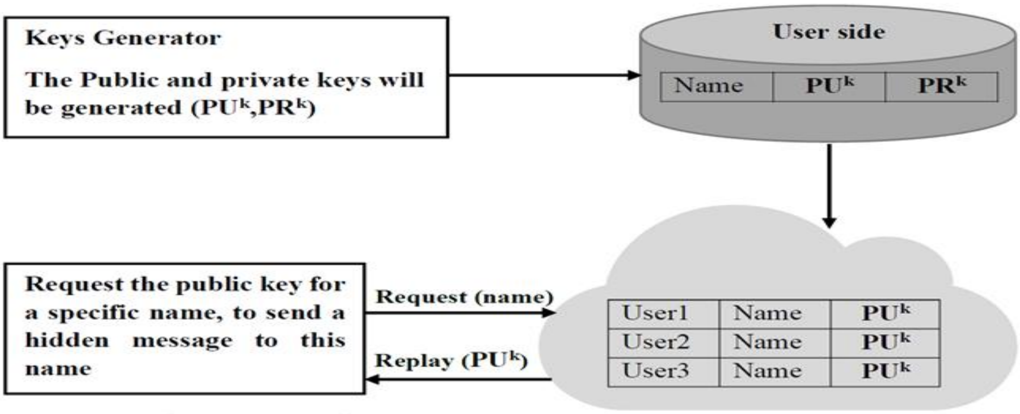
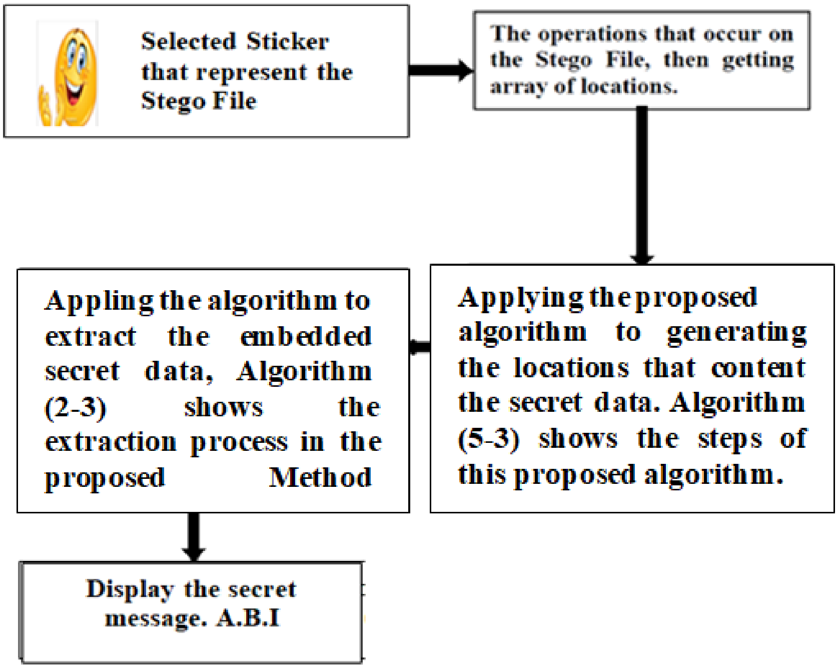

Keys Generator
The Public and private keys will be generated (PU^k,PR^k)
User side
Name
PU^k
PR^k
Request the public key for a specific name, to send a hidden message to this name
Request (name)
Replay (PU^k)
User1
User2
User3
Name
PU^k

Figure 10: Generating keys in layer three

Extraction Process: Layer One

In this layer, the un-sequential pixels are detected in the stego file., the values of the components RGBA are stored (see **Figure.12.**).

Figure.12. Extracting secret data from the stego file in layer one.


Selected Sticker that represent the Stego File
The operations that occur on the Stego File, then getting array of locations.
Applying the proposed algorithm to generating the locations that content the secret data. Algorithm (5-3) shows the steps of this proposed algorithm.
Appling the algorithm to extract the embedded secret data, Algorithm (2-3) shows the extraction process in the proposed Method
Display the secret message. A.B.I

Extraction Process: Layer Two

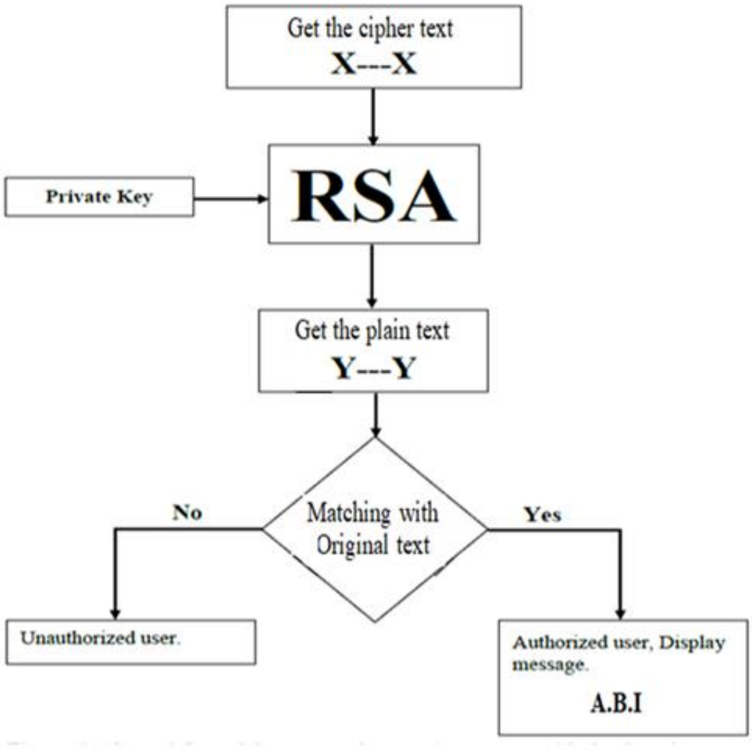
The purpose of this layer is to re-generate the original number of the secret data using the same parameters used during the embedding process as the following steps:

- Convert each digit into 4 bits
- Combine all the bits in a form of a stream.
- Apply the proposed method.
- Show the secret data (e.g., message).

Extraction Process: Layer Three

This layer is dedicated to secure the network during the key generation process. When the unreadable secret data is opened. **Figure.13.** shows the details of the process.

Figure.13. User validation.



MSE and PSNR Metrics:

These two metrics were applied to the proposed method (M2PAM). For benchmarking purposes, the metrics were also applied to M8PAM and M16PAM aiming at evaluating the proposed method against the other methods in the literature. The Mean Squared Error (MSE) reflects the squared difference between the pixels in the original and the affected images to the total number of pixels and can be calculated using the following equation **Van Wieringen, W. N. (2017)**:

$$MSE = \frac{1}{H * W} \sum_{x=1}^{W} \sum_{y=1}^{H} (lm_{original}(x,y) - lm_{affected}(x,y)^2) \quad \ldots(1)$$

Where lmoriginal represents the original image, lmaffected is the influenced image, and H*W is image size.

The Peak Signal to Noise Ratio (PSNR) measures the amount of noise in the affected image and can be formulated as follows **Liu, N., & Zhai, G. (2017)**:

$$PSNR = 10 \log_{10}\left(\frac{Max^2}{MSE}\right) \ldots(2)$$

Where Max denotes the maximum values of pixels.

3. EXPERIMENTAL RESULTS

We note that M2PAM has outperformed the other two methods in terms of MSE, PSNR and message size. This means that the proposed method generates fewer errors, which is desirable. As shown in **Table-2, Table-3 and Table-4**.

Table.2. shows the performance comparison results for the three algorithms (M2PAM, M8PAM, M16PAM) for a message size of 1173 bytes.

الغطاء	الخوارزمية	MSE	PSNR	Message size
	M8PAM	0.026	63.931	1173 bytes
	M16PAM	0.024	64.665	1173 bytes
	M16PAM	0.027	63.704	1173 bytes
	M16PAM	0.032	63.056	1173 bytes
	M2PAM	0.003	72.332	1173 bytes
	M2PAM	0.004	71.621	1173 bytes
	M2PAM	0.003	72.210	1173 bytes

Table.3. shows the performance comparison results for the three algorithms (M2PAM, M8PAM, M16PAM) for a message size of 1013 bytes.

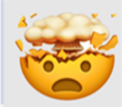
الغطاء	الخوارزمية	MSE	PSNR	Message size
	M8PAM	0.027	63.690	1013 bytes
	M16PAM	0.011	63.979	1013 bytes
	M16PAM	0.032	62.997	1013 bytes
	M16PAM	0.026	63.926	1013 bytes
	M2PAM	0.003	73.012	1013 bytes
	M2PAM	0.001	72.931	1013 bytes
	M2PAM	0.003	72.720	1013 bytes

Table.4. shows the performance comparison results for the three algorithms (M2PAM, M8PAM, M16PAM) for a message size of 1343 Bytes.

الغطاء	الخوارزمية	MSE	PSNR	Message size
	M8PAM	0.036	62.460	1343 bytes
	M16PAM	0.015	66.232	1343 bytes
	M16PAM	0.034	62.714	1343 bytes
	M16PAM	0.037	62.355	1343 bytes
	M2PAM	0.004	71.673	1343 bytes
	M2PAM	0.004	71.564	1343 bytes
	M2PAM	0.001	77.539	1343 bytes

4. CONCLUSIONS

This work proposes a method that is based on the M2PAM algorithm to perform steganography on secret data. The cover medium of the secret data is based on using WebP images that are widely used in social media applications. The proposed method is benchmarked against other methods in the literature. the proposed method showed outperformed the benchmarking in terms results of two metrics PNSR and MSE. Moreover, the experimental results showed that the hiding capacity of an image does not count on, the size of the image but on the number of non-transparent pixels in the image. Finally, the Mode metric is important to be investigated since it affects the image in a way that makes it doubtful to users. The use of WebP is considered more secure in hiding secret information. Also, this type is considered common social media and can be transferred fast.

As future works, it is planned to work on different versions of the WebP image type. The proposed approach can be developed to work efficiently with video or sound secret data. Finally, the proposed method can be developed to use more than one cover file to hide secret data.

References

Chinnasamy, P., Padmavathi, S., Swathy, R., & Rakesh, S. (2021). Efficient Data Security Using Hybrid Cryptography on Cloud Computing. In Inventive Communication and Computational Technologies (pp. 537-547). Springer, Singapore.

Islam, M. A., Kobita, A. A., Hossen, M. S., Rumi, L. S., Karim, R., & Tabassum, T. (2021). Data Security System for A Bank Based on Two Different Asymmetric Algorithms Cryptography. In Evolutionary Computing and Mobile Sustainable Networks (pp. 837-844). Springer, Singapore.

GENÇOĞLU, M. T. (2021). Enhancing The Data Security by using Audio Steganography with Taylor Series Cryptosystem. Turkish Journal of Science and Technology, 16(1), 47-64.

Ikhwan, A., Raof, R. A. A., Ehkan, P., Yacob, Y., & Syaifuddin, M. (2021, February). Data Security Implementation using Data Encryption Standard Method for Student Values at the Faculty of Medicine, University of North Sumatra. In Journal of Physics: Conference Series (Vol. 1755, No. 1, p. 012022). IOP Publishing.p. 11, doi: 10.1088/1742-6596/1755/1/012022, 2021.

Rout, H., & Mishra, B. K. (2014). Pros and cons of cryptography, steganography and perturbation techniques. IOSR Journal of Electronics and Communication Engineering, 76-81.

Panhwar, M. A., Khuhro, S. A., Mazhar, T., ZhongLiang, D., & Qadir, N. (2021). Quantum Cryptography: A way of Improving Security of Information. Computer Science, 16(1), 9-21.

Oad, A., Yadav, H., & Jain, A. (2014). A review: image encryption techniques and its terminologies. International Journal of Engineering and Advanced Technology (IJEAT) ISSN, 2249-8958.

Duan, X., Jia, K., Li, B., Guo, D., Zhang, E., & Qin, C. (2019). Reversible image steganography scheme based on a U-Net structure. IEEE Access, 7, 9314-9323.

[9] Zhang, R., Dong, S., & Liu, J. (2019). Invisible steganography via generative adversarial networks. Multimedia tools and applications, 78(7), 8559-8575.

Kuppusamy, P. G., Ramya, K. C., Rani, S. S., Sivaram, M., & Dhasarathan, V. (2020). A novel approach based on modified cycle generative adversarial networks for image steganography. Scalable Computing: Practice and Experience, 21(1), 63-72.

Subramanian, N., Elharrouss, O., Al-Maadeed, S., & Bouridane, A. (2021). Image Steganography: A Review of the Recent Advances. IEEE Access.

K. I. Rahmani, A. Kumar, and G. Manisha, “Study of Cryptography and Steganography System Study of Cryptography and Steganography System,” Int. J. Eng. Comput. Sci., vol. 4, no. August, pp. 10–13, doi: 10.18535/ijecs/v4i8.12, 2015.

Jain, R., & Boaddh, J. (2016, February). Advances in digital image steganography. In 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH) (pp. 163- 171). IEEE.

Al-Afandy, K. A., Faragallah, O. S., ELmhalawy, A., El-Rabaie, E. S. M., & El-Banby, G. M. (2016, October). High security data hiding using image cropping and LSB least significant bit steganography. In 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt) (pp. 400-404). IEEE.

Arya, A., & Soni, S. (2018). Performance evaluation of secrete image steganography techniques using least significant bit (LSB) method. vol, 6, 160-165.

Patel, N., & Meena, S. (2016, November). LSB based image steganography using dynamic key cryptography. In 2016 International Conference on Emerging Trends in Communication Technologies (ETCT) (pp. 1-5). IEEE.

Thenmozhi, S., & Chandrasekaran, M. (2013, January). A novel technique for image steganography using nonlinear chaotic map. In 2013 7th International Conference on Intelligent Systems and Control (ISCO) (pp. 307-311). IEEE.

Mahmood B., " WebP Image Steganography Using M8PAM for Android Applications", (2017), M.Sc. Thesis.

Bandyopadhyay, D., Dasgupta, K., Mandal, J. K., & Dutta, P. (2014). A novel secure image steganography method based on chaos theory in spatial domain. International Journal of Security, Privacy and Trust Management (IJSPTM), 3(1), 11-22.

Yu, X., Wang, C., & Zhou, X. (2018). A survey on robust video watermarking algorithms for copyright protection. Applied Sciences, 8(10), 1891.

Manjula, G. R., & Danti, A. (2015). A novel hash based least significant bit (2-3-3) image steganography in spatial domain. arXiv preprint arXiv:1503.03674.

Bawaneh, M. J., & Obeidat, A. A. (2016). A secure robust gray scale image steganography using image segmentation. Journal of Information Security, 7(03), 152.

Hussain, M., Wahab, A. W. A., Ho, A. T., Javed, N., & Jung, K. H. (2017). A data hiding scheme using parity-bit pixel value differencing and improved rightmost digit replacement. Signal Processing: Image Communication, 50, 44-57.

Van Wieringen, W. N. (2017). On the mean squared error of the ridge estimator of the covariance and precision matrix. Statistics & Probability Letters, 123, 88-92.

Liu, N., & Zhai, G. (2017). Free energy adjusted peak signal to noise ratio (FEA-PSNR) for image quality assessment. Sensing and Imaging, 18(1), 11.