




Enhanced Efficiency and Security in LSB2 Steganography: Burst Embedding and Private Key Integration



Rashad J. Rasras^{1*}, Mutaz Rasmi Abu Sara², Ziad Alqadi¹

¹ Department of Electrical Engineering, Al-Balqa' Applied University, Amman 11134, P.O. Box 15008, Jordan

² IT Department, Faculty of Engineering and Information Technology, Palestine Ahliya University, Bethlehem 1041, Palestine

Corresponding Author Email: rashad.rasras@bau.edu.jo

<https://doi.org/10.18280/ts.400502>

ABSTRACT

Received: 24 March 2023

Revised: 24 July 2023

Accepted: 8 September 2023

Available online: 30 October 2023

Keywords:

private key, image key, LSB2 steganography, position, burst embedding

In the realm of digital colour image steganography, the utilisation of an image key for the extraction of essential covering stego-bytes from predetermined secret positions was explored. The traditional Least Significant Bit (LSB) and LSB2 methodologies were streamlined by substituting the logical operations within both the hiding and extraction functions with straightforward assignment operations. An enhancement was introduced to the LSB2 steganographic methods for secret message conveyance without compromising data hiding capacity. Instead of the conventional character-by-character embedding seen in the LSB2 method, the binary code of the concealed message was embedded in a burst manner within the covering image. Similarly, the extraction from the stego image was conducted in a burst fashion, leading to a reduction in the processes required for both data hiding and extraction. As a result, shorter hiding and extraction durations were achieved, culminating in augmented data steganography throughput. For bolstering message security against potential breaches, the proposed ULSB2 method integrated a confidential private key (PK) composed of two double values. This key provisioned the necessary key space to deter hacking endeavours. A comparative analysis was conducted between the outcomes derived from the ULSB2 method and those of prevailing techniques to delineate the enhancements in both quality and speed of message steganography.

1. INTRODUCTION

Steganography is understood to be a security technique that facilitates the discreet embedding of secret data within cover media. This concealment ensures the data remains imperceptible to human detection, preserving its anonymity. For the execution of steganography, three pivotal components are requisite: the cover object, the confidential data, and the steganographic algorithm. It has been noted that the cover object can manifest in various forms, such as images, audio, or video.

In the realm of systems security that grapples with challenges posed by information threats, two principal strategies have been identified: cryptography and steganography. Cryptography is delineated as the art of information encryption, encompassing processes of both encryption and decryption. During encryption, confidential data is transformed in a manner that renders it undecipherable to potential eavesdroppers. This transformation occurs when an encryption key is employed by the sender, enabling the message's transmission via public, and often insecure, channels. In contrast, decryption is recognised as the reconversion of this encrypted text back into its original readable form. It has been asserted that successful decryption can only transpire if the decryption key is available to the receiver. Parallely, steganography is characterised as the act of covertly embedding a secret message within a cover file, ensuring its presence remains wholly obscured [1-7].

A distinct differentiation emerges between steganography

and cryptography. While the former seeks to entirely conceal the data, the latter morphs it into an obfuscated version. In scenarios invoking cryptography, it is possible for third parties to discern the occurrence of a communication, though the content remains encrypted and thus, unintelligible. Conversely, with steganography, any external observer remains oblivious to the presence of covert data or the occurrence of clandestine communication, owing to the profound concealment effected by an intermediary [1-7].

This study's emphasis has been placed on safeguarding secret messages harbouring confidential data during its transmission over communication channels, with the ultimate aim to reinforce both the privacy and integrity of said data.

Historically, the concealment of messages, ensuring that their very transmission remains undetected, can be traced back to ancient civilisations. A notable practice from the era of the ancient Greek empire has been documented wherein secret messages were inscribed on the scalps of shaved slaves. Once their hair regrew, the message was obscured, enabling the slave to be dispatched to the recipient. Upon arrival, the recipient would unveil the concealed message by shaving the slave's head again. Such historical instances signify the rudimentary attempts at using intermediaries to clandestinely transmit information, ensuring the very act of communication remains concealed [8-13].

In contemporary times, steganography finds its application in various crucial sectors. It has been observed that smart identification cards benefit from steganographic techniques, with specific details hidden within individuals' images.

Furthermore, within TCP/IP network packets, steganography aids in the covert inclusion of passwords or unduplicated data, allowing for the intricate analysis of specific user network traffic.

Steganography's multifaceted utility is also evident in securing online voting systems, facilitating the confidential interchange of data between governmental bodies, both nationally and internationally. This technique also proves indispensable in ensuring the safety of online banking transactions, safeguarding intelligence and military communications, and bolstering the clandestine transfer of sensitive data amongst defence establishments.

Various algorithms, each with its distinct characteristics, merits, and demerits, are employed to embed information within images. Notable among these are tools such as Jsteg, Outguess, and JPHide – all accessible for download online. Analogously, text and audio files have also been utilised to harbour concealed data. To the unassuming observer, such text or audio manifests as entirely innocuous [14-19].

A representation of the steganographic data model, as depicted in Figure 1, comprises the covering image, the covert message, the steganographic data method, and the resultant stego image [14-19].

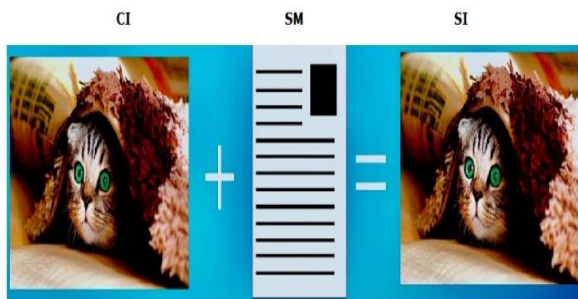


Figure 1. Depiction of the data steganography framework

The digital colour image, as delineated in references [20-27], stands as a colossal data reservoir, largely attributed to its high resolution, qualifying it as an optimal covering medium. Such images comprise a vast ensemble of pixels, configured in a three-dimensional matrix. This matrix is further subdivided into individual two-dimensional matrices, each representing a distinct colour—red, green, and blue—as showcased in Figure 2. An evaluation of the image's quality can be conducted through visual inspection of either the image itself or its associated colour histograms. These histograms, when amalgamated, result in the cumulative histogram of the image.

Several factors underpin the recommendation for utilising digital colour images in data steganography:

- Their expansive size, which facilitates the concealment of extensive messages.
- The relative ease with which the three-dimensional image matrix can be processed.
- Their ubiquity and cost-free generation.
- Pixel values' congruence with ASCII values, falling within the 0-255 range.
- The simplicity of deploying colour matrices and the feasibility of utilising an image segment for steganography purposes.

As illustrated in Figure 2, the representation of the covering image becomes evident. Paramount to the steganography process is the assurance that the stego image remains of

impeccable quality, closely mirroring its original counterpart, the covering image. The criteria that a steganographic method must fulfil are enumerated in Table 1.



Figure 2. Schematic representation of the covering image

Table 1. Enumerated quality benchmarks for steganography [1-8]

Quality Parameters	Measured Between Cover and Stego Images
MSE	Low
PSNR	High
CC	Closed to 1
NSCR	Closed to 0

Table 1 encapsulates the requisite quality benchmarks [1-8]. However, a salient issue is observed: steganographic methods, in their current form, do not guarantee security. It has been identified that adept hackers, possessing refined programming acumen, might retrieve concealed messages if aware of valuable information ensconced within the stego image. An efficacious countermeasure to this vulnerability is the integration of a clandestine PK. This key plays an instrumental role in both the obfuscation and retrieval of messages, a process depicted in Figure 3 [28, 29].

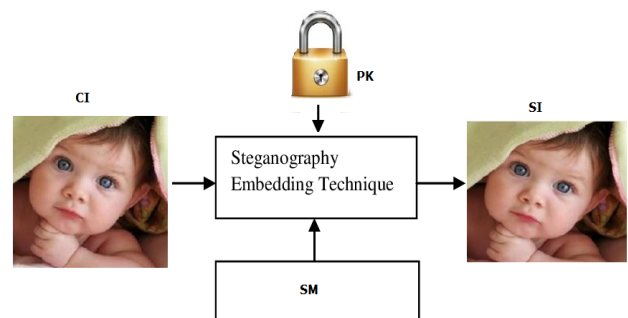


Figure 3. Illustration of secure data steganography mechanism

2. RELATED WORKS

One commonly employed method in data steganography is the LSB technique. This approach embeds information into a cover image by allocating 8 pixels from the cover image to

accommodate one character (byte) from the concealed message [30-34]. In the case of RGB colour images, which utilise a 24-bit colour encoding, the LSB from each of the red, green, and blue components can be utilised to host binary bits from the clandestine message. A depiction of this method, using three pixels of a 24-colour image consuming 9 bytes of memory, with the covert message represented as 01000001, can be observed in Figure 4. Alterations to this bit have been shown not to compromise the integrity of the cover image and are imperceptible to the human eye.



Figure 4. Implementation of the LSB method

The LSB method introduces minor pixel adjustments, which might involve no change, an addition of 1 to the pixel value, or a subtraction of 1. Such modifications result in a low Mean Square Error (MSE) value and a high Peak Signal-to-Noise Ratio (PSNR) value. Thus, any changes in the image introduced by this method are not discernible to human vision.

The capacity limit for the LSB method is determined by dividing the image size by 8. Notably, the security level offered by the LSB technique is considered low, although enhancements in security can be achieved by integrating a reference, which serves as a PK. When using the LSB technique for data steganography, it is ensured that the clandestine message is embedded into the cover image byte by byte. A portion of the cover image, equivalent in size to the message size multiplied by 8, must be earmarked to accommodate the concealed message, meaning the capacity of the LSB technique is defined as the cover image size divided by 8.

In embedding messages using the LSB method, the LSBs of sequential holding bytes store the concealed message bits. The initial byte of the message is followed by the second byte, and so forth. This process is further illustrated in Figure 5, where the stego colour image is interpreted as an array of RGB pixel values columns. Here, the LSBs of the columns are supplanted by the binary values of the ASCII representations of hidden messages, such as '5fk'.

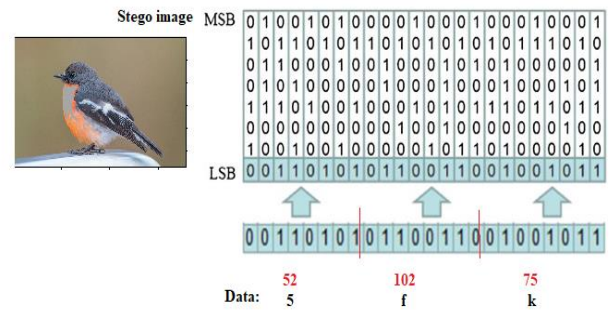


Figure 5. Message byte concealment using the LSB technique

Building upon the foundational principles of the LSB method, the LSB2 technique in data steganography has been introduced. This enhanced method reserves 4 bytes from the cover image to accommodate a single character from the concealed message, doubling the capacity compared to its predecessor [35-38]. Consequently, the character bits of the concealed message are stored in the two LSBs of the cover image, as depicted in Figure 6. The maximum capacity of this approach is determined by dividing the image size by 4.

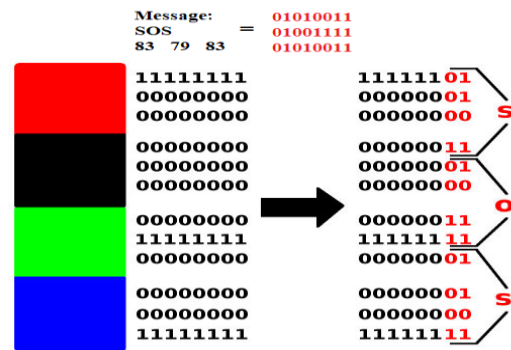


Figure 6. Data hiding using the LSB2 method

Variations in the byte values of the cover image, subsequent to the data hiding process, have been observed to range between -3 to +3. Notably, such alterations remain imperceptible to the human eye. The classical LSB2 method, as documented in [19], carries out data hiding and extraction by executing a sequence of logical operations. This approach embeds the concealed message byte-by-byte into consecutive bytes of the cover image.

Two illustrative examples further elucidating the treatment of each byte from the concealed message can be observed in Tables 2 to 5.

Table 2. Methodology of data concealment: Example 1

Covering byte	Hiding Process Operations	Holding byte
b(1)=217	s(1)=unit8(bitor(bitand(b(1),252),bitshift(a1,-6)))=217 a=bitand(a1,48)=0	217
b(2)=200	a=bitshift(a,2)=0 s(2)=unit8(bitor(bitand(b(2),252),bitshift(a,-6)))=200 a=bitand(a1,12)=0	200
b(3)=120	a=bitshift(a,4)=0 s(3)=unit8(bitor(bitand(b(3),252),bitshift(a,-6)))=120 a=bitand(a1,3)=1	120
b(4)=190	a(bitshift(a,6)=64 s(4)=unit8(bitor(bitand(b(4),252),bitshift(a,-6)))=189	189

Table 3. Data extraction procedure: Example 1

Holding byte	Hiding Process Operation	Character Weight
s(2)=217	d1=bitand(s(1),3)=1 d1=bitshift(d1,6)=64	64
s(2)=200	d2=bitand(s(2),3)=0 d2=bitshift(d2,4)=0	0
s(3)=120	d3=bitand(s(3),3)=0 d3=bitshift(d3,2)=0	0
s(4)=189	d4=bitand(s(4),3)=1	1
Sum	Extracted character	65

Table 4. Methodology of data concealment: Example 2

Converting byte	Hiding Process Operation	Holding byte
b(1)=217	s(1)=unit8(bitor(bitand(b(1),252),bitshift(a1,-6)))=219 a=bitand(a1, 48)=16 a=bitshift(a, 2)=64	219
b(2)=200	s(2)=unit8(bitor(bitand(b(2),252),bitshift(a,-6)))=200 a=bitand(a1,12)=12 a=bitshift(a,4)=192	201
b(3)=120	s(3)=unit8(bitor(bitand(b(3),252),bitshift(a,-6)))=123 a=bitand(a1,3)=3 a(bitshift(a,6)=192	123
b(4)=190	s(4)=unit8(bitor(bitand(b(4),252),bitshift(a,-6)))=189	191

Table 5. Data extraction procedure: Example 2

Holding byte	Hiding Process Operation	Character Weight
s(2)=219	d1=bitand(s(1),3)=3 d1=bitshift(d1,6)=192	192
s(2)=201	d2=bitand(s(2),3)=1 d2=bitshift(d2,4)=16	16
s(3)=123	d3=bitand(s(3),3)=3 d3=bitshift(d3,2)=12	12
s(4)=191	d4=bitand(s(4),3)=3	3
Sum	Extracted character	223

Example 1:

The objective was to conceal the character ‘A’, which possesses a decimal value denoted as a1=65. Its binary equivalent has been determined to be: 01000001. When presented with covering bytes represented as [b= [217 200 120 190]], the complement of 252 was found to be 3. The intricacies of the hiding process, executed via MATLAB functions, can be discerned in Table 2.

Subsequent to the data hiding process, data extraction can be undertaken. By leveraging a series of logical operations via MATLAB functions, the extraction process was realised, as depicted in Table 3.

Example 2:

With an aim to hide the character represented by the decimal a1=223, its binary form was identified as 11011111. Given the covering bytes b=[217,200,120,190], the complement of 3 was established to be 252. The hiding process, once again facilitated through MATLAB functions, is showcased in Table 4.

The subsequent stage involves the extraction of the embedded data. Delving into MATLAB functions, the extraction procedure was executed, the details of which are illustrated in Table 5.

3. PROPOSED ULSB2 STEGANOGRAPHIC TECHNIQUE

The refined ULSB2 steganographic method has been

introduced to embed message bits in a burst-like fashion, commencing from the MSB of each character byte. A comparison between the conventional LSB2 and the ULSB2 data hiding techniques is depicted in Figure 7. In the presented example, the characters 'ABC' — corresponding to the decimal ASCII codes (65, 66, 67) and their binary equivalents 01000001, 01000010, and 01000011 — serve as the secret message. Employing the novel ULSB2 approach, the binary representations of the characters 'ABC' can be conceptualised as a 3*8 matrix of binary digits. This matrix is subsequently transformed into a 12*2 matrix utilising the MCM MATLAB function, as illustrated in Figure 7.

For the implementation of the ULSB2 technique, a PK comprising two double data type values is utilised. The inaugural value represents the message length, whereas the latter indicates the starting position in the cover image for the processes of data embedding and extraction.

The data embedding procedure, as facilitated by the ULSB2 methodology, is elucidated in Figure 8. The outlined process can be segmented into the subsequent steps:

- (1) The message is read.
- (2) The message undergoes encoding to its decimal equivalent. As demonstrated in Figure 8, the message 'SOS' translates to the values (83,79,83).
- (3) The resultant decimal message from the preceding step is converted to its binary representation.
- (4) This binary message undergoes reshaping into a matrix with two columns using the MATLAB MCM function.
- (5) Acquisition of the PK is conducted.

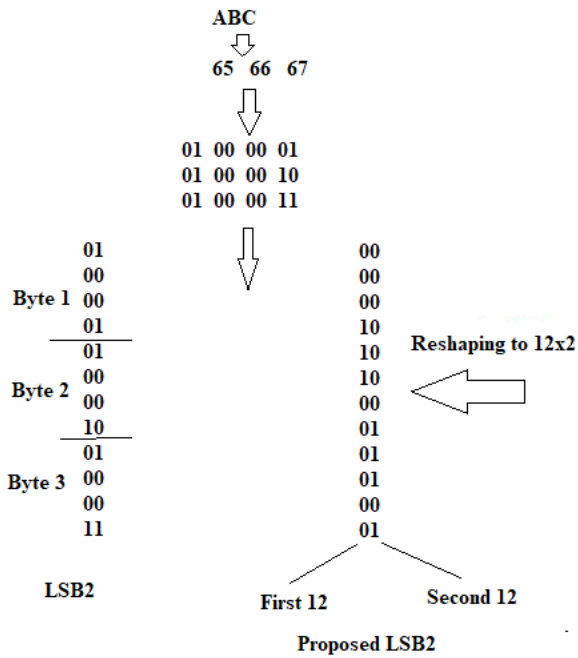


Figure 7. Comparison of data embedding using LSB2 and the proposed ULSB2 techniques

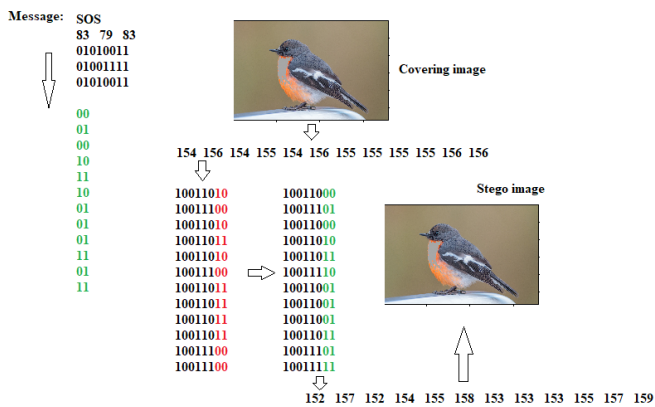


Figure 8. ULSB2 embedding procedure with 'SOS' as the secret message

(6) The cover image is sourced, and its dimensions are ascertained.

(7) The cover image undergoes reshaping into a single-row matrix.

(8) A segment from the row matrix, commencing from the position specified by the PK and equivalent in length to four times the message length, is extracted.

(9) The extracted segment is converted to binary.

(10) The two LSBs of this segment are set equal to the values from the MCM matrix.

(11) This segment undergoes conversion back to its decimal format.

(12) This decimal segment is then re-integrated into the original row matrix.

(13) The single-row matrix is reshaped into a 3D matrix, producing the steganographic image.

As depicted in Figure 9, the process of data extraction, facilitated by the proposed ULSB2 methodology, encompasses the subsequent steps:

(1) The steganographic image is acquired.

(2) The PK is sourced.

(3) Utilising the PK, the length of the embedded message is ascertained.

(4) The starting position for extraction, determined by the PK, is identified.

(5) The steganographic image is transformed into a single-row matrix.

(6) From the constructed row matrix, a segment, equivalent in size to four times the derived message length, is extracted commencing from the specified position.

(7) The isolated segment undergoes conversion to its binary form.

(8) The two LSBs from the binary segment are extracted, producing a two-column matrix.

(9) This two-column matrix is reshaped into an $8 \times n$ matrix, where n represents the number of rows.

(10) The resultant matrix is translated to its decimal equivalent.

(11) Finally, this decimal matrix is converted to its character representation, revealing the embedded message.

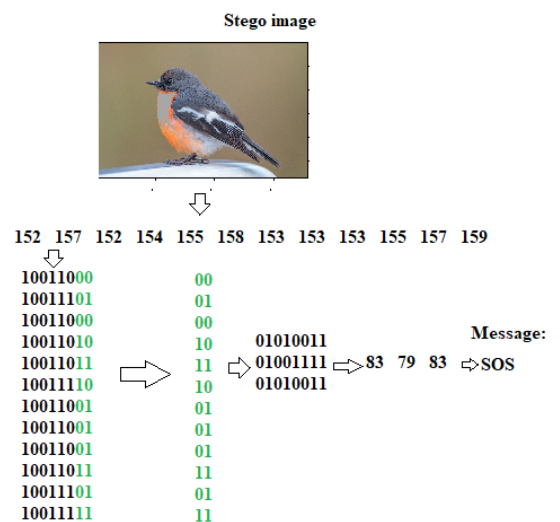


Figure 9. Illustration of the ULSB2 data extraction technique

4. IMPLEMENTATION AND RESULTS ANALYSIS

In evaluating image quality, two crucial metrics are commonly employed: the PSNR and the MSE. The PSNR metric is described as the ratio between the maximum conceivable power of a signal and the corrupting noise's power which impinges upon the integrity of its representation. Essentially, this metric facilitates a qualitative assessment between a reference image and a cover image embedded with concealed data. Conversely, the MSE metric quantifies the mean of the squared discrepancies between the distorted image and its reference counterpart. Notably, a high MSE coupled with a reduced PSNR indicates compromised image quality, whereas an elevated PSNR and a diminished MSE signify optimal quality. It is imperative for an adept data steganography technique to yield a stego image of superior quality, as evidenced by a minimal MSE and a heightened PSNR. The computation of the MSE value is accomplished by aggregating the squared variances of all pixels and subsequently dividing by the total pixel count, as elucidated in Eq. (1). Subsequent to this, PSNR can be derived through Eq. (2):

MSE between x and y, n: message length

$$MSE_{SR} = \frac{1}{N} \sum_{j=0}^{n-1} [x(j) - y(j)]^2, N = n \quad (1)$$

$$PSNR_{SR} = 10 \times \log_{10} \frac{(MAX_j)^2}{MSE_{SR}} \quad (2)$$

In the formulae, 'MAX' signifies the pinnacle pixel value, while 'N' denotes the aggregate sample count. The symbols x_j and y_j correspond to the sample values of the original and modified images, respectively.

Another pivotal statistic for gauging steganographic algorithm quality is the correlation coefficient (CC) between a pair of images. This metric provides insights into the interdependence between the grey values of the respective images. Evaluating the CC furnishes an understanding of the correlation magnitude between two images. Notably, coefficients within the $|1-0.7|$ range indicate robust correlation, suggesting significant similarity between source and encrypted file samples. A medium correlation is inferred from values within the $|0.7-0.3|$ span, whereas coefficients within the $|0.3-0|$ bracket signify weak correlation. The calculation of the CC is delineated in Eq. (3):

$$CC_{xy} = \frac{cov(x, y)}{\sqrt{D(x)} \sqrt{D(y)}} \quad (3)$$

where,

$$D(x) = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2,$$

$$D(y) = \frac{1}{N} \sum_{i=1}^N (y_i - \bar{y})^2,$$

$$cov(x, y) = \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}),$$

In the above equations, 'N' represents the cumulative sample number. Symbols x_j and y_j indicate the sample values of the cover and stego entities, respectively. The symbols \bar{x} and \bar{y} represent the mean sample values. Lastly, 'cov(x, y)' embodies the covariance between the two datasets.

The robustness of data steganography algorithms is often gauged through the Number of Sample Change Rate (NSCR),

a test crafted to assess algorithmic quality. Primarily, this test is devised to juxtapose the sample values of the covering and stego images, subsequently delineating the percentage variance. NSCR is mathematically derived using the equation provided as Eq. (4):

$$NSCR = \frac{\sum_{i=1}^N D_i}{N} \times 100\%, \quad (4)$$

where,

$$D_i \begin{cases} 1, x_i \neq y_i \\ 0, \text{Otherwise} \end{cases}$$

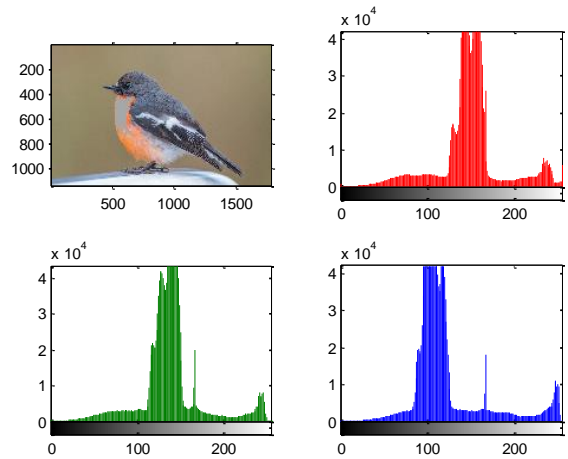


Figure 10. Histogram of covering image

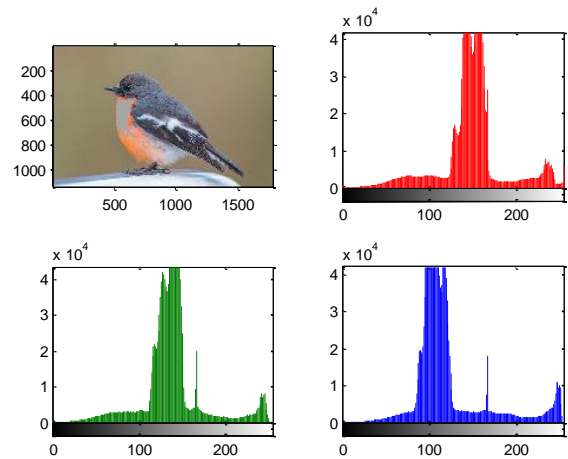


Figure 11. Histogram of LSB stego image

Table 6. Quality parameters - A comparison between covering and stego images (LSB method)

Message Size (byte)	MSE	PSNR	CCr	CCg	CCb	NSCR (%)
100	0.000063897	207.4077	1	1	1	0.0064
250	0.00016227	198.0875	1	1	1	0.0162
500	0.00031883	191.3338	1	1	1	0.0319
750	0.00048715	187.0946	1	1	1	0.0487
1000	0.00065171	184.1843	1	1	1	0.0652
2000	0.0013	177.1754	1	1	1	0.1314
3000	0.0020	173.1374	1	1	1	0.1967
4000	0.0026	170.2216	1	1	1	0.2633
5000	0.0033	168.0235	1	1	1	0.3280
7500	0.0049	163.9468	1	1	1	0.4931
10000	0.0065	161.1349	1	1	1	0.6532
Remark	Low	High	Equal 1	Equal 1	Equal 1	Low

Table 7. Speed parameters associated with the LSB method

Message Length (byte)	HT (Second)	ET (Second)	HTP (K bytes per Second)	ETP (K bytes per Second)
100	0.0560	0.0070	1.7440	13.8766
250	0.0982	0.0105	2.4852	23.3174
500	0.1027	0.0145	4.7536	33.6080
750	0.1298	0.0225	5.6438	32.6208
1000	0.1555	0.0305	6.2811	32.0069
2000	0.3131	0.1005	6.2384	19.4420
3000	0.3617	0.2108	8.0998	13.8992
4000	0.4688	0.3731	8.3318	10.4710
5000	0.6134	0.9514	7.9597	5.1320
7500	0.8574	3.4192	8.5427	2.1421
10000	1.1214	6.6394	8.7086	1.4709
Average	0.3889	1.0709	6.2535	17.0897

A series of messages was examined using the LSB method of data steganography. The histogram of the covering image is depicted in Figure 10, while Figure 11 showcases the stego image histograms, encapsulating a message of 10,000 characters.

Subsequent measurements and calculations focused on assessing quality and speed parameters. The derived results are illustrated in Tables 6 and 7.

Based on the data extracted from Table 6, it is discerned that the LSB method conformed to stipulated quality benchmarks. Irrespective of the embedded message's length, the stego image consistently approximated the covering image. Moreover, the LSB method exhibited commendable speed parameters, with an average concealment duration of 0.3889 seconds and an average extraction duration amounting to 1.0709 seconds.

In a parallel assessment, messages were evaluated using the conventional LSB2 data steganography method. The ensuing quality and speed metrics for this method are presented in Tables 8 and 9. Furthermore, Figure 12 displays the stego image, containing a 10,000 character message, accompanied by its RGB histograms.

From the insights gleaned from Table 8, the LSB2 method was identified as meeting the prescribed quality specifications. Analogous to the LSB method, the stego image mirrored the covering image, independent of the message length (be it short or extensive). As depicted in Table 9, the LSB2 steganography method advanced its speed metrics, marking a decline in both the concealment (0.0675 seconds on average) and extraction (0.0176 seconds on average) intervals.

Upon subjecting the same messages to the classical ULSB2 data steganography method, Tables 10 and 11 were formulated, illustrating the quality and speed parameters pertinent to this

technique. Concurrently, Figure 13 unveils the stego image encapsulating a 10,000-character message, complemented by its RGB histograms.

As per the insights drawn from Table 10, it was determined that the ULSB2 method met the designated quality criteria. For every message, irrespective of its length, the stego image exhibited a striking resemblance to the corresponding covering image.

In terms of speed metrics, the ULSB2 method showcased commendable efficiency, registering an average concealment duration of 0.0655 seconds and an average extraction time (ET) of 0.0161 seconds.

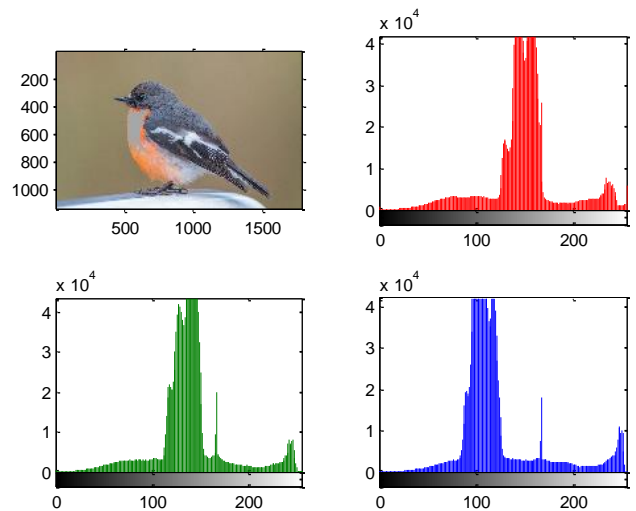


Figure 12. Histogram of LSB2 stego image

Table 8. Quality parameters - A comparison using the LSB2 method

Message Size(byte)	MSE	PSNR	CCr	CCg	CCb	NSCR (%)
100	0.00015688	198.4254	1	1	1	0.0047
250	0.00042685	188.4161	1	1	1	0.0126
500	0.00082445	181.8333	1	1	1	0.0247
750	0.0013	177.6636	1	1	1	0.0369
1000	0.0017	174.7973	1	1	1	0.0488
2000	0.0033	167.9772	1	1	1	0.0980
3000	0.0049	164.1031	1	1	1	0.1455
4000	0.0065	161.1374	1	1	1	0.1949
5000	0.0082	158.8670	1	1	1	0.2446
7500	0.0124	154.6928	1	1	1	0.3685
10000	0.0164	151.9457	1	1	1	0.4895
Remark	Low	High	Equal 1	Equal 1	Equal 1	Low

Table 9. Speed parameters associated with the LSB2 method

Message Length (byte)	HT (Second)	ET (Second)	HTP (K bytes per Second)	ETP (K bytes per Second)
100	0.0139	0.0036	7.0154	27.4023
250	0.0161	0.0042	15.1583	57.5912
500	0.0384	0.0113	12.7064	43.3754
750	0.0239	0.0071	30.6696	103.3866
1000	0.0272	0.0078	35.9522	125.0881
2000	0.0849	0.0123	22.9969	158.5753
3000	0.0600	0.0165	48.8089	177.9970
4000	0.0766	0.0206	50.9835	189.6864
5000	0.0975	0.0263	50.0984	185.9659
7500	0.1330	0.0355	55.0528	206.1772
10000	0.1705	0.0486	57.2690	201.0704
Average	0.0675	0.0176	35.1556	134.2105

Table 10. Quality metrics - contrasting covering and stego images (ULSB2 method)

Message Size (byte)	MSE	PSNR	CCr	CCg	CCb	NSCR (%)
100	0.00016211	198.0976	1	1	1	0.0048
250	0.00040789	188.8703	1	1	1	0.0121
500	0.00082772	181.7937	1	1	1	0.0246
750	0.0012	178.1467	1	1	1	0.0365
1000	0.0016	175.1476	1	1	1	0.0492
2000	0.0032	168.1488	1	1	1	0.0975
3000	0.0049	163.9229	1	1	1	0.1478
4000	0.0066	161.0171	1	1	1	0.1964
5000	0.0082	158.8858	1	1	1	0.2438
7500	0.0124	154.7252	1	1	1	0.3682
10000	0.0164	151.9435	1	1	1	0.4900
Remark	Low	High	Equal 1	Equal 1	Equal 1	Low

Table 11. Speed Metrics concerning the ULSB2 method

Message Length(byte)	HT(Second)	ET(Second)	HTP (K bytes per Second)	ETP (K bytes per Second)
100	0.0138	0.0020	7.0856	49.9930
250	0.0159	0.0027	15.3499	89.4616
500	0.0198	0.0040	24.7051	123.1292
750	0.0236	0.0049	30.9800	150.7351
1000	0.0274	0.0065	35.6684	150.3560
2000	0.0428	0.0118	45.6653	165.0449
3000	0.0997	0.0164	29.3855	179.1496
4000	0.0765	0.0188	51.0303	208.2123
5000	0.0961	0.0260	50.8231	188.1623
7500	0.1342	0.0351	54.5886	208.4938
10000	0.1711	0.0493	57.0842	197.9476
Average	0.0655	0.0161	36.5787	155.5169

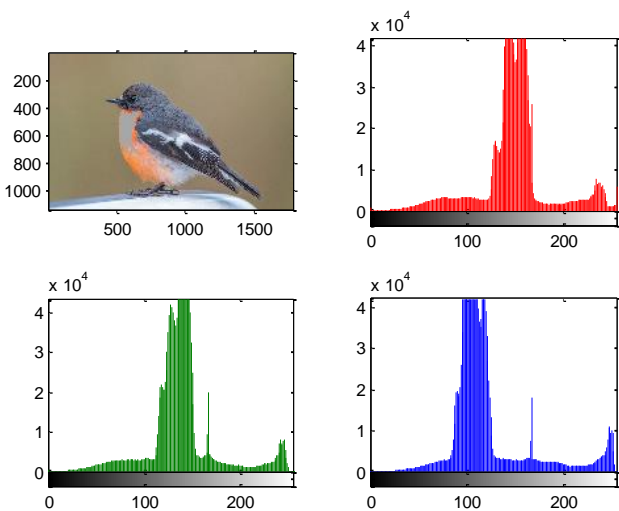


Figure 13. Histogram of ULSB2 stego image ULSB2

Upon examination of the data in Table 10, it was ascertained that the ULSB2 method adhered to stringent quality standards. In every instance, regardless of the hidden message's length, the stego image bore remarkable similarity to its corresponding covering image. The ULSB2 method also manifested superior speed parameters, registering an average concealment duration of 0.0655 seconds and an average extraction duration of 0.0161 seconds.

The speed metrics obtained underscored the proficiency of the proposed ULSB2 method, notably in both hiding times (HT) and ET, as illuminated in Figures 14 and 15. A notable augmentation in data concealment and extraction speeds for the proposed method is articulated in Tables 12 and 13.

From the collated results, an inference was drawn underscoring that methods LSB, LSB2, and ULSB2 aligned with the designated quality standards, delivering robust quality metric values. The derived PSNR values for these triadic data steganography methods, as depicted in Figure 16, met the acceptable threshold.

The speed enhancement attributable to the proposed method, particularly in concealment duration (HT), is chronicled in Table 12. This enhancement, or speedup, was computed by contrasting the average concealment time across methods against the average time exhibited by ULSB2. Similarly, the escalation in extraction speed is elaborated upon in Table 13, where the metric is derived by equating the average ET across the spectrum of methods against the ULSB2 benchmark.

Table 12. Speedup assessment vis-a-vis ULSB2 (HT)

Method	HT (Second)	Speedup of ULBS
LSB	0.3889	5.9374
LSB2	0.0675	1.0305
ULSB2	0.0655	1.0000

Table 13. Speedup assessment vis-a-vis ULSB2 (ET)

Method	ET (Second)	Speedup of ULBS
LSB	1.0709	66.5155
LSB2	0.0176	1.0932
ULSB2	0.0161	1.0000

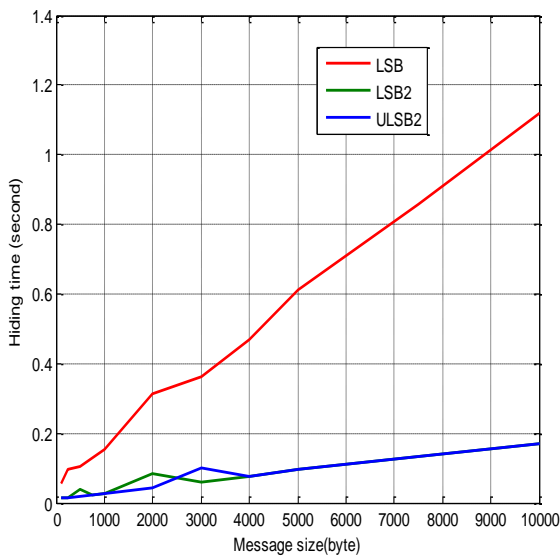


Figure 14. Comparative analysis of HTs

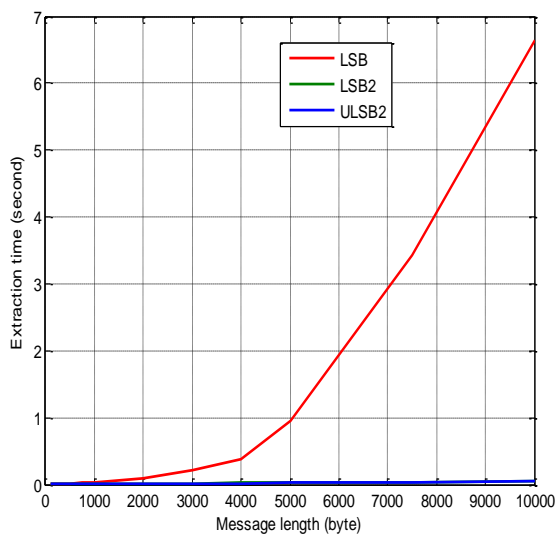


Figure 15. Comparative analysis of ETs

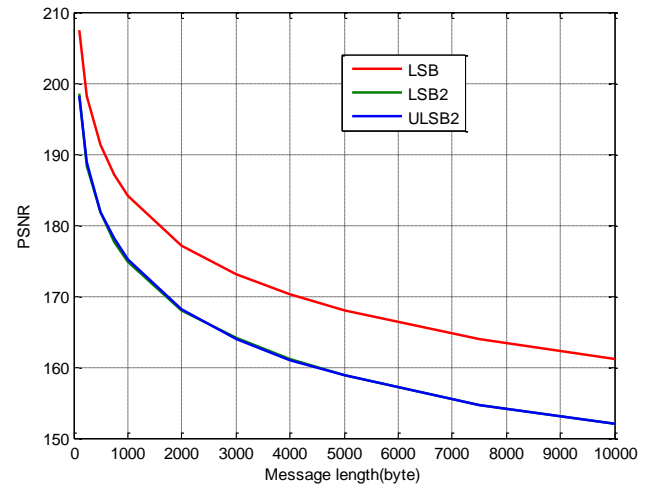


Figure 16. PSNR comparative analysis

5. CONCLUSION

Experimental outcomes indicated the inadequacy in security associated with both the LSB and LSB2 methods, with both necessitating extended durations for hiding and extraction processes. A simplification of the LSB and LSB2 methods was subsequently undertaken by substituting the logical operations present in both hiding and extraction functions with straightforward assignment operations. With the advent of this novel ULSB2 data steganography technique, messages were concealed and unveiled in a burst-like manner.

In enhancing the steganographic layer of secret protection, the introduction of a PK played a pivotal role. This not only extended an ample key space but also fortified the concealed message against potential hacking attempts.

When the aforementioned method was applied to a range of message lengths, whether short or long, favourable results were consistently observed for parameters such as MSE, PSNR, CC, and NSCR, all measured against the covering and stego images, thereby fulfilling the stipulated quality prerequisites.

Upon evaluating the efficacy of the ULSB2 method, it was discerned that its speed metrics surpassed those of its LSB and LSB2 counterparts, particularly in facets of data concealment and extraction. A noteworthy implication of the ULSB2 method lies in its capability to supplant the traditional LSB and LSB2 methods entirely, coupled with its innate aptitude to manage messages of varying lengths. This method holds promise in safeguarding covert communications containing sensitive data, thereby bolstering the data's privacy and integrity during transmission.

Looking ahead, it is posited that further exploration into novel embedding techniques might unveil potential alterations to concealed data during its transmission, paving the way for enhanced steganographic methodologies.

REFERENCES

- [1] Abu-Ein, A., Alqadi, Z.A., Nader, J. (2016). A technique of hiding secrete text in wave file. *International Journal of Computer Applications*, 9(2): 96-103.
- [2] Shayeb, J.N.I., Alqadi, Z., Nader, J. (2019). Analysis of digital voice features extraction methods. *International*

- Journal of Educational Research and Development, 1(4): 49-55.
- [3] Sharadq, A., Al-Qadi, Z., Zahran, B., Nader, J. (2016). Experimental investigation of wave file compression-decompression. *International Journal of Computer Science and Information Security*, 14(10): 774.
- [4] Hindi, Z.A.A.A.Y., Majed, O.D. (2020). Procedures for speech recognition using LPC and ANN. *International Journal of Engineering Technology Research & Management*, 4(2): 48-55.
- [5] Al-Dwairi, M.O., Hendi, A., AlQadi, Z. (2019). An efficient and highly secure technique to encrypt-decrypt color images. *Engineering, Technology & Applied Science Research*, 9(3): 4165-4168.
- [6] Hendi, A.Y., Dwairi, M.O., Al-Qadi, Z.A., Soliman, M.S. (2019). A novel simple and highly secure method for data encryption-decryption. *International Journal of Communication Networks and Information Security*, 11(1): 232-238.
- [7] Rasras, R.J., Zahran, B., Sara, M.R.A., AlQadi, Z. (2021). Developing digital signal clustering method using local binary pattern histogram. *International Journal of Electrical and Computer Engineering (IJECE)*, 11(1): 872-878. <https://doi.org/10.11591/ijece.v11i1.pp872-878>
- [8] Aqel, M.J., ALQadi, Z., Abdullah, A.A. (2018). RGB color image encryption-decryption using image segmentation and matrix multiplication. *International Journal of Engineering & Technology*, 7(3.13): 104-107.
- [9] Nadir, J., Ein, A.A., Alqadi, Z. (2016). A technique to encrypt-decrypt stereo wave file. *International Journal of Computer and Information Technology*, 5(5): 465-470.
- [10] Khawatreh, S., Ayyoub, B., Abu-Ein, A., Alqadi, Z. (2018). A novel methodology to extract voice signal features. *International Journal of Computer Applications*, 975: 8887.
- [11] Al-Dwairi, M.O., Hendi, A.Y., Soliman, M.S., Alqadi, Z.A. (2019). A new method for voice signal features creation. *International Journal of Electrical and Computer Engineering*, 9(5): 4077. <https://doi.org/10.11591/ijece.v9i5.pp4092-4098>
- [12] Al-Qaisi, A., Khawatreh, S.A., Sharadqah, A.A., Alqadi, Z.A. (2018). Wave file features extraction using reduced LBP. *International Journal of Electrical and Computer Engineering*, 8(5): 2780. <https://doi.org/10.11591/ijece.v8i5.pp.2780-2787>
- [13] Abu-Faraj, M., Al-Hyari, A., Alqadi, Z. (2022). A Dual Approach for Audio Cryptography. *Journal of Southwest Jiaotong University*, 57(1): 24-33.
- [14] Abu-Faraj, M.A., Al-Hyari, A., Alqadi, Z. (2022). A complex matrix private key to enhance the security level of image cryptography. *Symmetry*, 14(4): 664. <https://doi.org/10.3390/sym14040664>
- [15] Kuyoro, A., Nzenwata, U.J., Awodele, O., Idowu, S. (2022). GAN-based encoding model for reversible image steganography. *Revue d'Intelligence Artificielle*, 36(4): 561-567. <https://doi.org/10.18280/ria.360407>
- [16] Mua'ad Abu-Faraj, K.A., Alqadi, Z. (2021). Deep machine learning to enhance ANN performance: Fingerprint classifier case study. *Journal of Southwest Jiaotong University*, 56(6). <https://doi.org/10.35741/issn.0258-2724.56.6.61>
- [17] Rasras, B.Z.R.J., Alqadi, Z., Sara, M.R.A., Zahran, B. (2019). Developing new Multilevel security algorithm for data encryption-decryption (MLS_ED). *International Journal of Advanced Trends in Computer Science and Engineering*, 8(6): 3228-3235.
- [18] Abu-Faraj, M., Alqadi, Z., Aldebei, K. (2021). Comparative analysis of fingerprint features extraction methods. *Journal of Hunan University Natural Sciences*, 48(12): 177-182.
- [19] Rasras, R.J., Sara, M.R.A., Alqadi, Z. (2023). Efficient method to message-image cryptography using reordered image-key. *Traitement du Signal*, 40(1): 235. <https://doi.org/10.18280/ts.400122>
- [20] Al-Dwairi, M.O., Alqadi, Z.A., Abujazar, A.A., Zneit, R.A. (2010). Optimized true-color image processing. *World Applied Sciences Journal*, 8(10): 1175-1182.
- [21] Ulbeh, W.A., Moustafa, A., Alqadi, Z.A. (2009). Gray image reconstruction. *European Journal of Scientific Research*, 27(2): 167-173.
- [22] Moustafa, A.A., Alqadi, Z.A. (2009). Color image reconstruction using a new R'G'I model. *Journal of Computer Science*, 5(4): 250. <https://doi.org/10.3844/jcs.2009.250.254>.
- [23] Aqel, M.J., ALQadi, Z., Abdullah, A.A. (2018). RGB color image encryption-decryption using image segmentation and matrix multiplication. *International Journal of Engineering & Technology*, 7(3.13): 104-107. <https://doi.org/10.14419/ijet.v7i3.13.16334>.
- [24] Zahran, B., Alqadi, Z., Nader, J., Ein, A.A. (2016). A comparison between parallel and segmentation methods used for image encryption-decryption. *International Journal of Computer Science & Information Technology (IJCSIT) Volume*, 8(5): 127-133. [10.5121/ijcsit.2016.8509](https://doi.org/10.5121/ijcsit.2016.8509)
- [25] Matrouk, K., Al-Hasanat, A., Alasha'ary, H., Al-Qadi, Z., Al-Shalabi, H. (2014). Analysis of matrix multiplication computational methods. *European Journal of Scientific Research*, 121.
- [26] Alqadi, Z.A., Aqel, M., El Emary, I.M. (2008). Performance analysis and evaluation of parallel matrix multiplication algorithms. *World Applied Sciences Journal*, 5(2): 211-214.
- [27] Alqadi, Z.A., Abu-Jazzar, A.M.J.A.D. (2005). Analysis of program methods used for optimizing matrix multiplication. *Journal of Engineering*, 15(1): 73-78.
- [28] Das, R., Das, I. (2016). Secure data transfer in IoT environment: Adopting both cryptography and steganography techniques. In *2016 Second International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, IEEE, pp. 296-301. <https://doi.org/10.1109/ICRCICN.2016.7813674>.
- [29] Rasras, R.Z., Sara, M.R.A., AlQadi, Z.A., Zneit, R.A. (2019). Comparative Analysis of LSB, LSB2, PVD Methods of Data Steganography. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(3): 748-754. <https://doi.org/10.30534/ijatcse/2019/64832019>
- [30] Emam, M.M., Aly, A.A., Omara, F.A. (2016). An improved image steganography method based on LSB technique with random pixel selection. *International Journal of Advanced Computer Science and Applications*, 7(3). <https://doi.org/10.14569/IJACSA.2016.070350>
- [31] Abuzalata, J.A.A.M., Alqadi, Z., Al-Azzeh, J., Jaber, Q. (2019). Modified inverse LSB method for highly secure message hiding. *International Journal of Computer*

- Science and Mobile Computing, 8(2): 93-103.
- [32] Rasras, R.J., AlQadi, Z.A., Sara, M.R.A. (2019). A methodology based on steganography and cryptography to protect highly secure messages. *Engineering, Technology & Applied Science Research*, 9(1): 3681-3684.
- [33] Zhou, X., Gong, W., Fu, W., Jin, L. (2016). An improved method for LSB based color image steganography combined with cryptography. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pp. 1-4. <https://doi.org/10.1109/ICIS.2016.7550955>.
- [34] Wu, D.C., Tsai, W.H. (2003). A steganographic method for images by pixel-value differencing. *Pattern Recognition Letters*, 24(9-10): 1613-1626. [https://doi.org/10.1016/S0167-8655\(02\)00402-6](https://doi.org/10.1016/S0167-8655(02)00402-6)
- [35] Aqel, M.J., Alqadi, Z.A., El Emary, I.M. (2007). Analysis of stream cipher security algorithm. *Journal of Information and Computing Science*, 2(4): 288-298.
- [36] Al-Azzeh, J., Zahran, B., Alqadi, Z., Ayyoub, B., Abu-Zaher, M. (2018). A novel zero-error method to create a secret tag for an image. *Journal of Theoretical & Applied Information Technology*, 96(13). <http://dx.doi.org/10.13140/RG.2.2.14533.35048>
- [37] Alqadi, Z.A.A., Abu Zalata, M.K., Qaryouti, G.M. (2016). Comparative analysis of color image steganography. *JCSMC*, 5(11): 37-43.
- [38] Jose, M. (2014). Hiding image in image using LSB insertion method with improved security and quality. *International Journal of Science and Research*, 3(9): 2281-2284.