Published Online First: September, 2024
https://doi.org/10.21123/bsj.2024.10371
P-ISSN: 2078-8665 - E-ISSN: 2411-7986

Baghdad Science Journal

Enhancing Internet Data Security: A Fusion of Cryptography, Steganography, and Machine Learning Techniques

Omar Fitian Rashid *[1], Mohammed Ahmed Subhi[2,3], Marwa Khudhur Hussein[4], Mohammed Najah Mahdi[5]

[1]Department of Geology, College of Science, University of Baghdad, Baghdad, Iraq.
[2]Department of Planning, Directorate of Private University Education, Ministry of Higher Education and Scientific Research, Baghdad, Iraq.
[3]Balad Technical Institute, Medical Technical University, Baghdad, Iraq.
[4]Department Construction and Projects, University of Baghdad, Baghdad, Iraq.
[5]ADAPT Centre, School of Computing, Dublin City University, Dublin, Ireland.
*Corresponding Author.

Received 04/12/2023, Revised 30/04/2024, Accepted 02/05/2024, Published Online First 20/09/2024

Abstract

Cryptography and steganography play critical roles in ensuring the security of network communications. Combining these methods holds great potential for safeguarding information transmitted over the internet. DNA Cryptography, a modern and robust technique, leverages the unique properties of DNA for secure data handling. The increasing of cyber threats has made it necessary to propose a new secure communication method to block unauthorized access to the sending information. This paper introduces an innovative system that integrates cryptography, steganography, and machine learning techniques to enhance data transfer security over the Internet. The system unfolds in six steps to encrypt and hide text. The initial step involves using the Caesar cipher to encode the original text. This is followed by the conversion of the text into DNA bases. Subsequent steps include the conversion of DNA bases to ASCII format and further transformation into binary numbers. The fifth step introduces a dynamic element by shifting binary numbers using a random key. The final step involves covertly embedding these binary numbers (ciphertext) within an image. Beyond traditional metrics, machine learning elements have been incorporated to elevate the system's performance. Performance evaluation was conducted across three standard images with varying data sizes, demonstrating the system's effectiveness. The proposed system showcased rapid cryptography times, with encryption and decryption times of 2.802 ms and 3.388 ms, respectively. The integration of machine learning techniques enriches the system's capabilities, presenting a compelling solution for secure and efficient data transfer over the Internet.

Keywords: DNA Cryptography, Decryption, Machine Learning, Steganography, Security, Encryption.

Introduction

The internet these days is widely and daily used, at the same time, this has led to increase network security threats due to the large data transfer number over the network. The attackers are trying to steal important information by interrupting the transfer process. So, security has become one of the most important things in our life. Banks and governmental buildings are the most important area that applies security because of their data importance. Where the objective of this study aims to address the enhance data transfer security over the Internet. Many methods have been invented to apply security and ensure sending data; the most used methods are cryptography and steganography[1]. One of the modern types of cryptography is DNA cryptography, which encodes information using DNA computing techniques due to the DNA properties like parallel molecular computing, storing, transmitting the data, and computing capabilities[2]. DNA encoding is used in other areas such as Cryptography [3-6], Intrusion detection [7-8], and Steganography Techniques as in references [9-12.]

Cryptography is applied in several research works by using DNA cryptography to encrypt information. Vikram et al.[3] proposed a DNA symmetric cryptography method that is used to boost data security, and their results showed that the text encryption process has high security. Beggas & Lounici [4] proposed a novel method for One Time Pad cryptosystems by generate random and unpredictable key, this done by using DNA sequences as a source for inherently random. The proposed method is done based on two steps: used DNA techniques and chaotic function. A new cryptosystem method was developed by the authors in reference [5] that used DNA cryptography and finite automata. The proposed method depends on receiver attributes for key generation and is used in encryption. The authors in reference [6] developed Dynamic DNA, a key-based method that can use many information forms like audio, images, and text. This method depends on generating a random key every time the sender wants to send a message, which makes this method strong versus attacks.

On the other hand, the steganography technique is used to hide information from unauthorized people to read or access this information, and a lot of research developed in this field. The authors in reference [9] developed a new steganography method depending on the Neural Network technique and Least Significant Bit (LSB) for DNA codon, where data hides inside DNA bases. The results showed that this method has a low execution time and a high security level. A haze image steganography method is proposed by reference [10], this method depends on hiding information within the weather effects of the picture used. This method consisted of three phases the estimating the model parameter, determining haze impacts, and embedding the text. The authors in reference [11] proposed an image steganography technique to enhance image transformation. This technique uses a hybrid chaotic map and mixed pixel positions of the cover image to make the system hard to break by the attackers. A new steganography method was proposed by reference [12], this method starts with converting data into a binary number, after that converting it to DNA sequences as nitrogen bases, then converting these sequences to amino acids, and finally, converting these amino acids to binary numbers and hide it in an image, and this image is sent to the receiver. A new method was proposed by Zha et al. [13] for adversarial steganography, where this method concentrated on optimizing the distribution of stego that affects security, this is done by using novel protocol to enhance cover. The authors in reference [14] proposed a new steganographic method without using a present cover image. This method creates an image containing different colors by mentioning a reference picture using the weighted color transfer method. The proposed method starts with encoding the sending message to boost the security level, and then derives an optimal weight value based on a specific measure method. After that, get a temporary image by representing each pixel in a floating-point way. Finally, hide the secret message in image pixels using the embedding vector. Performance evaluation is done using 5,000 images. The obtained results showed that this technique is strong against salt-and-pepper noise attacks, and the chance of breaking the proposed method is weak. Naser et al. [15] suggested a new message transferring method that secure the transferring process by used RC4 cipher algorithm and in the sometime added another layer for least significant bit embedding algorithm, the achieved results showed that this system provided high performance. A novel steganography method is proposed [16] based on insignificant DCT coefficients structure, where the information is superimposed on the polynomial-modeled coefficients. Al-Hassani [17] created a novel method to secure data cryptosystem based on chaotic logistic map by using

2-Dimensional key matrix, this method has got high efficiency results when tested it hundreds of image samples. A new technique is needed to developed in order to overcoming previous techniques' limitations such as high computation time, small key size, and not suitable for large messages. Finally, machine learning algorithms are applied to speed up training times [18], various previous techniques are proposed by using machine learning algorithms [19-21].

Materials and Methods

DNA cryptography is one of the modern promised cryptography techniques with high-security efficiency; this field was proposed and used after the invention of DNA computing. The new suggested method has two phases: encryption and hiding phase and extraction phase. Where DNA has high storage capacity, fast processing, and high computation capacity, and is more secure than other cryptography algorithms. One of the advantages of using DNA cryptography is can be applied for both symmetric and asymmetric cryptography. The encryption and hiding phase exist in the sender part, where this phase includes both the cryptography method and the steganography method. This phase has six steps, which are illustrated in Fig 1.

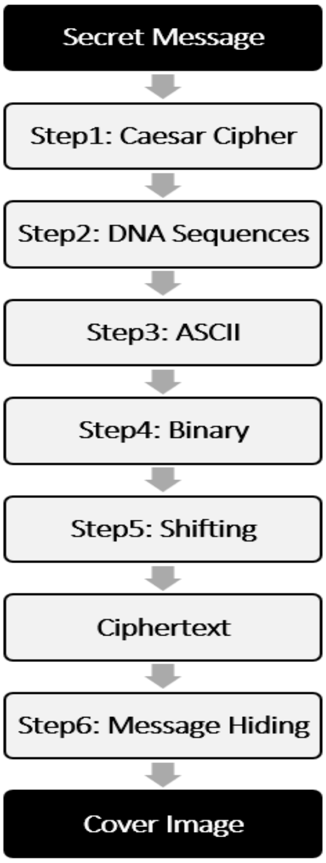

Figure 1. Encryption and hiding scheme image.

Firstly, using Caesar cipher method to encrypt the message; this method is a simple substitution cipher encryption technique; this method is applied by substituting each character with a different letter using a specific position number. An example of Caesar's cipher is shown as follows:

If the plaintext is equal to "My code is C36", and the shifting value is equal to 5, then the Ciphertext will equal to "R^%htij%nx%H8;"

Secondly, a new DNA encoding idea is proposed, where a new DNA encoding table is built based on all possible text cryptography values that include Characters, Numbers, and special characters. The characters consist of 52 possible values (including capital letters and small letters), the numbers contain 10 values (from 0 to 9), and special characters include 17 values. Therefore, the total values are equal to 79 values, where four DNA characters used can handle and represent all possible values. These values and their equivalent DNA encoding sequences are shown in Table 1.

Published Online First: September, 2024
https://doi.org/10.21123/bsj.2024.10371
P-ISSN: 2078-8665 - E-ISSN: 2411-7986
Baghdad Science Journal

Table 1. DNA encoding

Symbol	DNA	Symbol	DNA	Symbol	DNA	Symbol	DNA
A	ACGG	U	CGAT	p	CTCC	9	ACTG
B	TTCA	V	GAAC	q	GGTA	!	GTGT
C	CTGG	W	CGAC	r	GAGA	@	CGTA
D	ATAT	X	TGTC	s	CCTG	#	AAGG
E	GCAG	Y	CTAC	t	CTGA	$	CATT
F	GCGT	Z	TATA	u	AGAT	%	TCTG
G	GTAC	a	AATG	v	TGAG	&	AGCA
H	TCAT	b	TAAC	w	TGCC	+	ATTA
I	AATT	c	ACTA	x	TACA	-	AACT
J	GCAA	d	CAAT	y	AAGA	*	TTTG
K	CCCG	e	ATCG	z	AGAA	/	TCGT
L	CATG	f	AATA	0	ATCC	=	AAGT
M	CTTT	g	GACT	1	GGCC	?	CTTA
N	TGTT	h	TACG	2	TTGT	)	GCAT
O	GACC	i	TCTC	3	CATC	(	TGCG
P	TTTT	j	TGAA	4	AGAG	.	AAAA
Q	AAGC	k	GTTG	5	CCGT	,	CCCC
R	CTCT	l	CCTA	6	TTAT		GGGG
S	CTTG	m	GGAT	7	CGCC		
T	CTAA	n	GTTT	8	AGCC		

The suggested DNA encoding table is used to convert text that was extracted from step one to DNA sequences. Where DNA sequences for the extracted text are:

CTCTGCATTCTGTACGCTGATCTCTGAATCTGGTTTTACATCTGTCATAGCCTGCG

Algorithm 1 shows the steps to create a DNA encoding table.

```
Algorithm 1: Creating DNA Encoding Table
Input: Characters, Digits, and Special characters
Output: DNA encoding table
1. Generate DNA sequences for all values
DNA_Sequence (1) ← Generate a random sequence with a length of four characters
Total_Sequences ← 1
for i ← 2 to Total number of values (equal to 79) do
    New_Sequence ← Generate a random sequence with a length of four characters
    for j ← 1 to Total_Sequences do
        if New_Sequence = Sequence (j)
            Generate new sequence
        Else
            Total_Sequences ← Total_Sequences + 1
            Sequence (Total_Sequences) ← New_Sequence
        end if
    end for
end for
```

Thirdly, convert the DNA sequence to ASCII. DNA sequence values can be either A, C, G, or T, and the ASCII values for these characters are 65, 67, 71, and 84 respectively.

Fourthly, convert ASCII values to binary numbers, which are equal to 01000001, 01000011, 01000111, and 01010100 for all four numbers in the previous step respectively. The achieved binary numbers in the last example are equal to:

010000110101010001000011010101000100011101
000011010000010101010001010100010000110101
010001000111010101000100000101000011010001
110100001101010100010001110100000101010100
010000110101010001000011010101000100011101
000001010000010101010001000011010101000100
011101000111010101000101010001010100010101
000100000101000011010000010101010001000011
010101000100011101010100010000110100000101
010100010000010100011101000011010000110101
0100010001110100001101000111

The last cryptography step used a key (ex: the key is equal to 14) to switch binary numbers. The achieved binary numbers are equal to a ciphertext that will send, these numbers after applying the shifting key as follow:

000011010001110100001101010100010000110101
010001000111010000110100000101010100010101
000100001101010100010001110101010001000001
010000110100011101000011010101000100011101
000001010101000100001101010100010000110101
010001000111010000010100000101010100010000
110101010001000111010001110101010001010100
010101000101010001000001010000110100000101
010100010000110101010001000111010101000100
001101000001010101000100000101000111010000
11010000110101010001000111 01

Finally, hiding the ciphertext (binary numbers) within the cover image; this is done by storing binary numbers length at the first picture location equal to (0,0). After that, two random values were used as x and y positions to hide all binary numbers (ciphertext). The steps of message encryption and hiding are shown in Algorithm 2.

Algorithm 2: Encryption and hiding

```
Input: Message, shifting value, DNA encoding table, Cipher key.
Output: Cover image.
1. Applied Caesar cipher to encrypt the message.
for i ← 1 to Length (Message)
T= Substring (Message, i, 1)
Shift_Text = Char (ASCII (T) + Shifting_Value)
Cipher = Cipher & Shift_Text
end for
2. Converted text to DNA sequences based on DNA encoding table
for i ← 1 to Length (Cipher)
for j ← 1 to DNA_Encoding_Table_Values
if Substring(Cipher, i, 1) = Substring (DNA_Encoding_Table, j, 1)
DNA_Text= DNA_Text & DNA_Encoding_Table (j,2)
end if
end for
end for
3. Converted DNA sequence to ASCII
for i ← 1 to Length (DNA_Text)
A= Substring (DNA_Text, i, 2)
ASCII_Values = ASCII_Values & ASCII (A)
end for
4. Converted ASCII values to binary numbers
for i ← 1 to Length (ASCII_Values)
B= Substring (ASCII_Values, i, 1)
```

```
    j = 1
    res = 0
    while B > 0
        rem = B % 2
        res = res + (j * rem)
        B =B /2
        j = j*10
    end while
    Binary_Numbers = Binary_Numbers & res
end for
5. Shifted binary numbers based on the Cipher key
for i ← Cipher_key +1 to Length (Binary_Numbers)
    Final_Ciphertext = Final_Ciphertext & Binary_Numbers (i)
end for
for i ← 1 to Cipher_key
    Final_Ciphertext = Final_Ciphertext & Binary_Numbers (i)
end for
6. Hide the ciphertext (binary numbers) within the cover image
Picture (0,0) = Length (Final_Ciphertext)
for i ← 1 to Length (Final_Ciphertext)
    Randomize 0
    X= random()
    Y=random()
    V= Substring (Final_Ciphertext, i, 1)
    Picture (x,y) = Picture (x,y) & V
end for
```

Extracting the hidden message from the received image and decode it in the receiver part; this phase has six steps, as illustrated in Fig. 2.

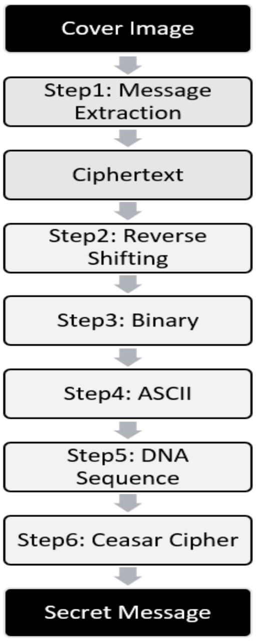

Figure 2. Extract and decode hidden message scheme.

The steps for extracting and decoding hidden messages are shown in Algorithm 3.

Algorithm 3: Extract and decode the hidden message

```
Input: Cover image, shifting value, DNA encoding table, Cipher key.
Output: Original message

1. Extract the ciphertext (binary numbers)
Final_Ciphertext_Length = Picture (0,0)
for i ← 1 to Final_Ciphertext_Length
    Randomize 0
    X= random()
    Y= random()
    V = Picture (x,y)
    Final_Ciphertext = Final_Ciphertext & V
end for
2. Shifted binary numbers (reverse shifting) based on the Cipher key
for i ← Length (Final_Ciphertext) - Cipher_key +1 to Length (Final_Ciphertext)
    Binary_Num = Binary_Num & Final_Ciphertext (i)
end for
for i ← 1 to Length (Final_Ciphertext) - Cipher_key
    Binary_Num = Binary_Num & Final_Ciphertext (i)
end for
3. Converted binary numbers to ASCII values
for i ← 1 to Length (Binary_Num step 8
    A= Substring (Binary_Num, i, 8)
    res = 0
    for j ← 1 to 8
        res = res + (Substring (A, j, 1) * (2^j))
    end for
    ASCII_Values = ASCII_Values & res
end for
4. Converted ASCII to DNA sequence
for i ← 1 to Length (ASCII_Values)
    D= Substring (ASCII_Values, i, 2)
    DNA_Text = DNA_Text & Char (D)
end for
5. Converted DNA sequences to text based on DNA encoding table
for i ← 1 to Length (DNA_Text)
    for j ← 1 to DNA_Encoding_Table_Values
        if Substring (DNA_Text, i, 4) = Substring (DNA_Encoding_Table, j, 2)
            Shift_Text = Shift_Text & DNA_Encoding_Table (j,1)
        end if
    end for
end for
6. Applied Caesar cipher (reverse) to decrypt the message.
for i ← 1 to Length (Shift_Text)
    T= Substring (Shift_Text, i, 1)
    Plaintext = Char (ASCII (T) - Shifting_Value)
    Message = Message & Plaintext
end for
```

The steps to extract the hidden message:

- Reading and finding binary numbers hidden within the cover image.
- Inverting the shifting process using the same key.
- Converting binary groups to ASCII format, which are 01000001, 01000011, 01000111, and 01010100, where the ASCII numbers for these groups are equal to 65, 67, 71, and 84 respectively.
- Then converting ASCII numbers to characters, which are A, C, G, and T (all possible DNA values).
- Converting DNA sequences to text using Fig. 2 steps mentioned previously.
- The last step is to decrypt the text by applying the Caesar cipher.

Results and Discussion

Four experiments are used to evaluate our suggested system for this purpose: cryptography time, Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and the consumed time for the steganography process. Where the cryptography and steganography time is calculated to showed that the proposed method is overcoming the time problem that previous methods had. While MSE and PSNR is calculated to demonstration the robust the proposed steganography method. Firstly, the calculated cryptography time for files with various volumes in terms of milliseconds (ms) is illustrated in Table 2 and Fig. 3.

Table 2. The time needed for the cryptography process (in terms of ms)

Size	Numbers of Characters	Encryption Time (ms)	Decryption Time (ms)
1K	999	2.802	3.388
2K	2049	11.981	9.823
3K	3079	23.514	11.007

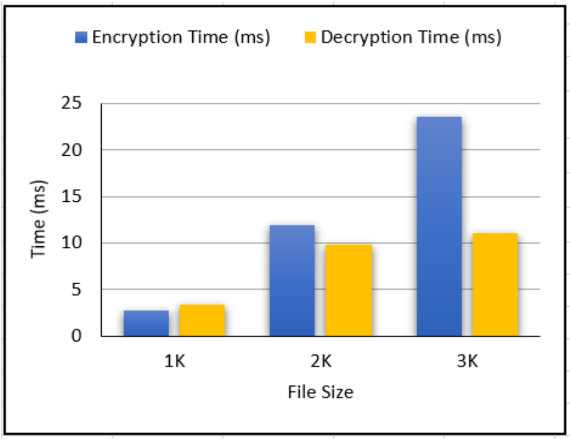

Figure 3. Exhibited the time needed for the cryptography process for different files size.

Table 2 and Fig. 3 show that the encryption time and decryption time are fast, where the encryption time for three different files, which have a size of 1K, 2K, and 3K is equal to 2.802 ms, 11.981 ms, and 23.514 ms respectively, while the decryption time for the same files is equal to 3.388 ms, 9.823 ms, and 11.007 ms respectively.

Our experiment is done using three standard images with a size equal to 512x512 pixels, and these images are exhibited in Fig. 4. An example of steganography process, Fig. 5 displays the application of the proposed steganography method on the first image (Lena) and show the visual image before and after hide the bits inside it.

Figure 4. Extract and decode hidden message schemes.

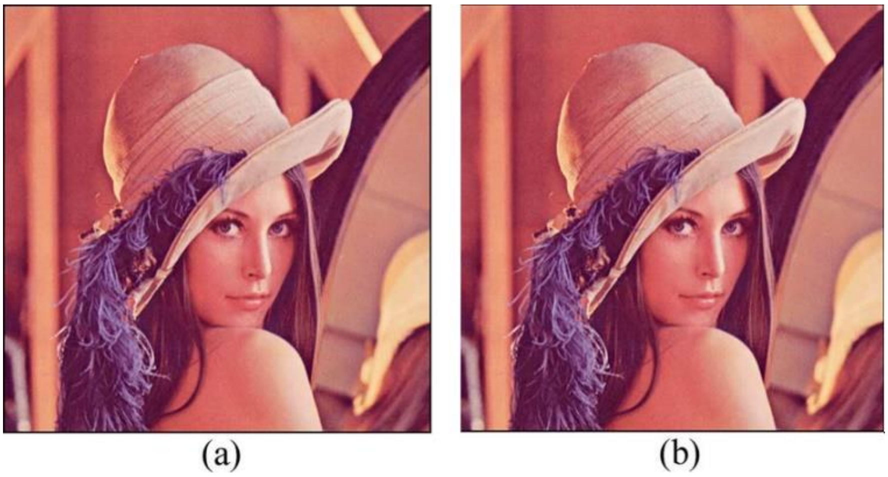

Figure 5. Steganography process (a) original image (b) after hide bits inside it

Secondly, MSE and PSNR are utilized to compare image compression quality [22], and these measurements can be calculated based on Eq. 1 and Eq. 2.

$$MSE = \frac{\sum_{M.N}[I_1(m.n) - I_2(m.n)]^2}{M * N} \qquad 1$$

The numbers of image rows and columns are presented by M and N respectively.

$$PSNR = 10 * log_{10} \frac{(R^2)}{MSE} \qquad 2$$

Where image fluctuation is presented by R.

MSE for the proposed method with various file sizes is illustrated in Table 3 and Fig. 6. The achieved PSNR results are clarified in Table 4 and Fig. 7.

Table 3. MSE results for the proposed method.

Image	File size		
	1k	**2k**	**3k**
(a) Lena	30.977	62.809	72.995
(b) Parrots	33.541	66.730	76.631
(c) Pepper	32.299	68.001	77.409

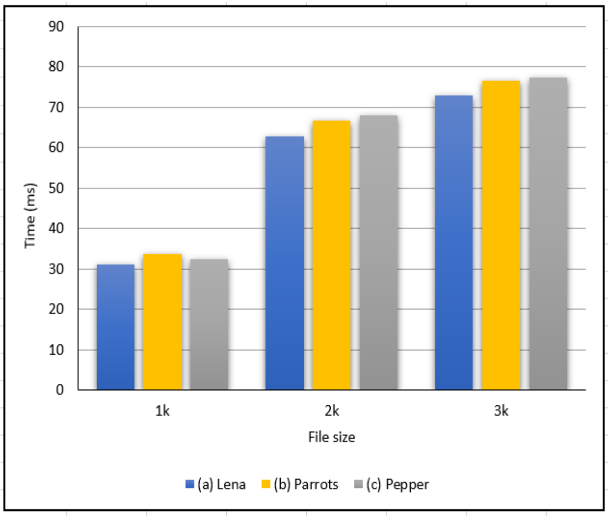

Figure 6. MSE results for the proposed method.

From Table 3 and Fig. 6, it is clear that the calculated MSE value for three standard images (Lena, Parrots, and Pepper) using three different file sizes is low, that mean error rate is low. The MSE results for the first standard image (Lena) for different files with sizes equal to 1K, 2K, and 3K equal 30.977, 62.809, and 72.995 respectively. The MSE results for the second standard image (Parrots) for the same files are equal to 62.809, 66.730, and 68.001 respectively, and finally, the MSE results for the last standard image (Pepper) for the same files are equal to 72.995, 76.631, and 77.409 respectively.

Table 4. PSNR results for the proposed method

Image	File size		
	1k	**2k**	**3k**
(a) Lena	32.446	29.008	28.871
(b) Parrots	33.081	30.340	28.884
(c) Pepper	33.113	30.910	29.907

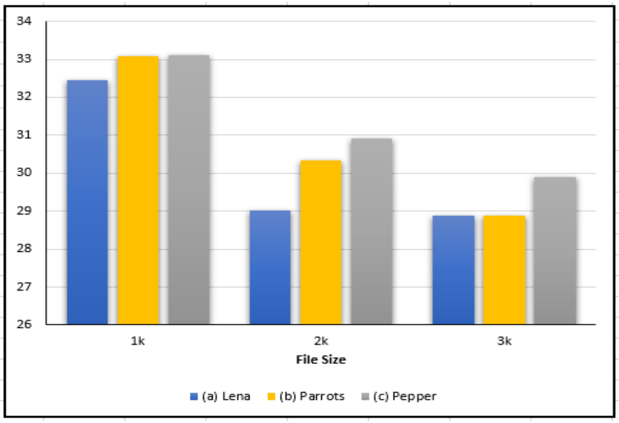

Figure 7. PSNR results for the proposed method.

As shown in Table 4 and Fig. 7, the achieved PSNR is high which means the compression quality is good. The PSNR results for the first standard image for various files with sizes equal to 1K, 2K, and 3K equals 32.446, 29.008, and 28.871 respectively. The PSNR results for the second image for the same files are equal to 33.081, 30.340, and 28.884 respectively, and the PSNR results for the third image for the same files are equal to 33.113, 30.910, and 29.907 respectively.

Finally, the steganography time needed is calculated for three various file sizes in terms of milliseconds (ms), which is shown in Table 5.

Table 5. The time needed for the steganography process (in terms of ms)

Size	Time (ms)	(a) Lena	(b) Parrots	(c) Pepper
1K	Hide message	1.998	2.003	2.119
	Extract message	0.930	1.003	1.249
2K	Hide message	4.477	4.732	3.904
	Extract message	2.499	2.001	2.033
3K	Hide message	4.671	4.980	5.623
	Extract message	2.314	3.015	2.467

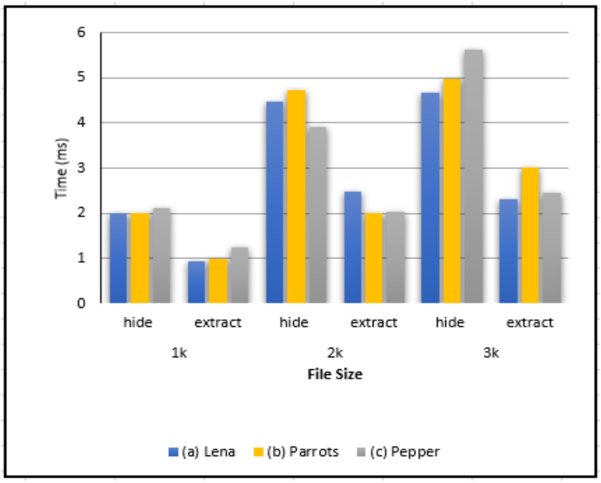

Figure 8. The time needed for the steganography process.

Table 5 and Fig. 8 show that both message hiding time and message extraction time are very fast. The time needed to hide the message for a file with a size of 1K for three standard images is equal to 1.998 ms, 2.003 ms, and 2.119 ms respectively, while the time needed to extract the message for the same file and images are equal to 0.930 ms, 1.003 ms, and 1.249 ms respectively. Also, the time needed to hide the message for a file with a size of 2K for the same images is equal to 4.477 ms, 4.732 ms, and 3.904 ms respectively, while the time needed to extract the message for the same file and images are equal to 2.499 ms, 2.001 ms, and 2.033 ms respectively. Finally, the hiding time for a file with a size of 3K for the same images is equal to 4.671 ms, 4.980 ms, and 5.623 ms respectively, and the extraction time for the same file and images is equal to 2.314 ms, 3.015 ms, and 2.467 ms respectively.

The results of the currently proposed method are compared with the results of other cryptography methods to evaluate its performance and highlight its ability to encode and hide the message. Table 6 and Fig. 9 exhibit a comparison in terms of encryption time and decryption time between the results of the proposed method and with three cryptography methods mentioned in references [23-25].

Table 6. Comparison between the results obtained from the present proposed method with results achieved by other cryptography methods

	File Size	Encryption Time (ms)	Decryption Time (ms)
Kataria et al. [23]	600 Byte	20289	-
Proposed method	600 Byte	1.868	-
Akiwate & Parthiban [24]	10 KB	5529.8784	4.54851
Proposed method	10 KB	59.924	49.115
Pradhan et al. [25]	262144 Byte	4574	4753
Proposed method	262144 Byte	155.753	143.091

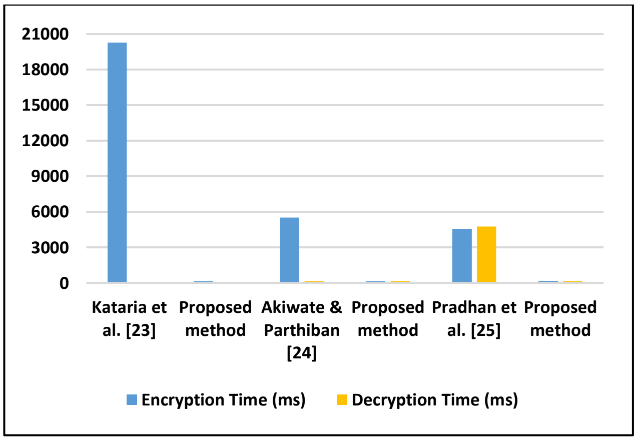

Figure 9. Comparison between the encryption time and decryption time obtained from the current proposed system with other methods.

From Table 6 and Fig. 9, it is noted that the encryption time and decryption time obtained by the current proposed system is quite good. The proposed method gives a faster encryption time result compared with Kataria et al. [25] when using a file with 600 bytes, and the achieved encryption time is equal to 1.868 ms, also when comparing our achieved time results with reference [26] when using a file with 10 KB, the proposed method obtained faster encryption time result and this result is equal to 59.924 ms, but the decryption time obtained by reference [26] is faster and equal to 4.54 ms. Finally, the encryption time and decryption time achieved by the proposed method achieved faster results compared with results achieved by reference [27] when using a file with 262144 bytes, and these results are equal to 155.753 ms and 143.091 ms.

Conclusion

A new method to secure text transmission over the internet is proposed by combining both cryptography and steganography techniques. This method consists of six steps: applying the Caesar cipher algorithm to encrypt text, then converting the text obtained from the previous step to DNA bases, then converting these bases to ASCII format, then converting ASCII to binary numbers, after that using the shifting method to shift these binary numbers depending on the random key, and finally hiding the final binary numbers that obtained from the previous step inside an image. The performance evaluation was done based on cryptography time, Mean Square Error, Peak signal-to-noise ratio, and consumed time for the steganography process.

The achieved encryption time is equal to 2.802 ms, while the decryption time is equal to 3.388 ms. Also, the achieved MSE and PSNR results are equal to 30.977 and 32.446 respectively. Finally, the consumed time for the steganography process: text hiding time is equal to 1.998 ms, while the text extracting time from the image is equal to 0.930 ms. The obtained results are considered fast and good. The proposed method can be applied for any type of public application that used to transfer private information which provide a data protection process against intruders. For future work, the proposed method can be improved by replacing the Caesar cipher method by DES or AES algorithm, and using video frames in the steganography method instead of image this will make the steganography more secure.

Authors' Declaration

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Furthermore, any Figures and images, that are not ours, have been included with the necessary permission for re-publication, which is attached to the manuscript.
- No animal studies are present in the manuscript.
- No human studies are present in the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee at University of Baghdad.

Authors' Contribution Statement

O.F.R., M.A.S., M.K.H., and M.N.M. collectively conceived and designed the study. O.F.R. and M.A.S. implemented the data security techniques, specifically cryptography and steganography, in the methodology. M.K.H. conducted experiments and data curation. M.N.M. oversaw the project, provided supervision, and secured funding. O.F.R., M.A.S., M.K.H., and M.N.M. collectively analyzed the results. O.F.R., M.A.S., M.K.H., and M.N.M. contributed to the writing of the paper, incorporating insights from all authors.

References

1. Kaundal AK, Verma AK. DNA Based Cryptography: A Review. IJICT. 2014; 4(7): 693-698.
2. Bhimani P. A Review on Cryptography Techniques using DNA Computing. IJCERT. 2018; 5(6): 187-191. https://doi.org/10.22362/ijcert/2018/v5/i6/v5i604
3. Vikram A, Kalaivani S, Gopinath G. A Novel Encryption Algorithm based on DNA Cryptography. International Conference on Communication and Electronics Systems (ICCES). 2019. https://doi.org/10.1109/ICCES45898.2019.9002399
4. Beggas F, Lounici A. Generation of random sequences using DNA cryptography for OTP encryption. Biosyst. 2023; 234. https://doi.org/10.1016/j.biosystems.2023.105064
5. Pavithran P, Mathew S, Namasudra S, Lorenz P. A novel cryptosystem based on DNA cryptography and randomly generated mealy machine. Comput Secur. 2021; 104. https://doi.org/10.1016/j.cose.2020.102160
6. Akiwate B, Parthiban L. A Dynamic DNA for Key-based Cryptography. International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS). 2018; 223-227. https://doi.org/10.1109/CTEMS.2018.8769267
7. Rashid OF, Othman ZA, Zainudin S, Samsudin NA. DNA Encoding and STR Extraction for Anomaly Intrusion Detection Systems. IEEE Access. 2021; 31892 – 31907. https://doi.org/10.1109/ACCESS.2021.3055431
8. Rashid OF. DNA Encoding for Misuse Intrusion Detection System based on UNSW-NB15 Data Set. Iraqi J Sci. 2020; 61(12): 3408-3416. https://doi.org/10.24996/ijs.2020.61.12.29
9. Mohammed MH, Taloba AI, Ali BH. DNA-Based Steganography Using Neural Networks. 2018 International Japan-Africa Electronics, Communications, and Computations (JAC-ECC). 2018; 79-82. https://doi.org/10.1109/JEC-ECC.2018.8679564
10. Qi B, Yang C, Tan L, Luo X, Liu F. A Novel Haze Image Steganography Method via Cover-Source Switching. J Vis Commun. Image Represent. 2020; 70: 1-11. https://doi.org/10.1016/j.jvcir.2020.102814
11. Sharafi L, Khedmati Y, Shabani MM. Image steganography based on a new hybrid chaos map and discrete transforms. Optik. 2021; 226. https://doi.org/10.1016/j.ijleo.2020.165492
12. Sushma RB, Namitha MV, Manjula GR, Johar S, Hiriyanna GS. DNA based Steganography Using 2-3-3 Technique. Data Science and Communication. 2019; 1-6. https://doi.org/10.1109/IconDSC.2019.8816945
13. Zha H, Zhang W, Yu N, Fan Z. Enhancing image steganography via adversarial optimization of the stego distribution. Signal Process. 2023; 212. https://doi.org/10.1016/j.sigpro.2023.109155
14. Hsieh K, Wang C. Constructive image steganography using example-based weighted color transfer. J Inf Secur Appl . 2022; 65. https://doi.org/10.1016/j.jisa.2022.103126
15. Naser MA, Al-alak SMK, Hussein AM, Jawad MJ. Steganography and Cryptography Techniques Based

Secure Data Transferring Through Public Network Channel. Baghdad Sci J. 2022; 19(6): 1362-1368. https://dx.doi.org/10.21123/bsj.2022.6142

16. Rabie T, Baziyad M, Kamel I. Secure high payload steganography: A model-based approach. J Inf Secur Appl . 2021; 63. https://doi.org/10.1016/j.jisa.2021.103043
17. Al-Hassani MD. A Novel Technique for Secure Data Cryptosystem Based on Chaotic Key Image Generation. Baghdad Sci. J. 2022; 19(4): 905-913. http://dx.doi.org/10.21123/bsj.2022.19.4.0905
18. Xian G. Parallel machine learning algorithm using fine-grained-mode spark on a mesos big data cloud computing software framework for mobile robotic intelligent fault recognition. IEEE Access. 2020; 8: 131885-131900. https://doi.org/10.1109/ACCESS.2020.3007499
19. Salman SA, Dheyab SA, Salih QM, Hammood WA. Parallel Machine Learning Algorithms. Mesopotamian J Big Data. 2023; 12–15. https://doi.org/10.58496/MJBD/2023/002
20. Almomani A, Nahar K, Alauthman M, Al-Betar MA, Yaseen Q, Gupta BB. Image cyberbullying detection and recognition using transfer deep machine learning. Int J Cogn Comput. Eng. 2024; 5: 14-26. https://doi.org/10.1016/j.ijcce.2023.11.002
21. Kaduwela NA, Horner S, Dadar P, Manworren RCB. Application of a human-centered design for embedded machine learning model to develop data labeling software with nurses: Human-to-Artificial Intelligence (H2AI). Int J Med Inform. 2024; 183. https://doi.org/10.1016/j.ijmedinf
22. Subramanian N, Elharrouss O, Al-Maadeed S, Bouridane A. Image Steganography: A Review of the Recent Advances. IEEE Access. 2021; 9: 23409-23423. https://doi.org/10.1109/ACCESS.2021.3053998
23. Kataria S, Singh K, Kumar T, Nehra MS. ECR (Encryption with Cover Text and Reordering) based Text Steganography. IEEE Second International Conference on Image Information Processing (ICIIP-2013). 2013; 612-616. https://doi.org/10.1109/ICIIP.2013.6707666
24. Akiwate B, Parthiban L. A Dynamic DNA for Key-based Cryptography. International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS). 2018; 223-227. https://doi.org/10.1109/CTEMS.2018.8769267
25. Pradhan D, Som S, Rana A. Cryptography Encryption Technique Using Circular Bit Rotation in Binary Field. 8th International Conference on Reliability, Infocom Technologies and Optimization Amity University. 2020. https://doi.org/10.1109/ICRITO48877.2020.9197845

Published Online First: September, 2024
https://doi.org/10.21123/bsj.2024.10371
P-ISSN: 2078-8665 - E-ISSN: 2411-7986

Baghdad Science Journal

تعزيز أمان بيانات الإنترنت: دمج تقنيات التشفير والتستر وتقنيات تعلم الآلة

عمر فتيان رشيد[1]، محمد احمد صبحي[2،3]، مروة خضر حسين[4]، محمد نجاح مهدي[5]

[1]قسم علم الارض، كلية العلوم، جامعة بغداد، بغداد، العراق.
[2]قسم التخطيط، دائرة التعليم الجامعي الاهلي، وزارة التعليم العالي والبحث العلمي، بغداد، العراق.
[3]المعد التقني/ بلد، الجامعة التقنية الوسطى، بغداد، العراق.
[4]قسم الاعمار والمشاريع، جامعة بغداد، بغداد، العراق.
[5]مركز بحوث تقنيات الذكاء الاصطناعي ADAPT، دبلن، ايرلندا.

الخلاصة

يلعب علم التشفير والتستر أدوارًا حاسمة في ضمان أمان الاتصالات الشبكية. يحمل تجميع هذه الطرق إمكانات كبيرة لحماية المعلومات المرسلة عبر الإنترنت. يقوم علم التشفير الجيني، وهو تقنية حديثة وقوية، بالاستفادة من الخصائص الفريدة لحمض الدي إن إيه لمعالجة البيانات بشكل آمن. يقدم هذا البحث نظامًا مبتكرًا يدمج تقنيات التشفير والتستر وتقنيات تعلم الآلة لتعزيز أمان نقل البيانات عبر الإنترنت. يتكون النظام من ست خطوات لتشفير وإخفاء النص. تتضمن الخطوة الأولى استخدام التشفير بطريقة سيزر لترميز النص الأصلي. يتبع ذلك تحويل النص إلى قواعد حمض الدي إن إيه. تتضمن الخطوات التالية تحويل قواعد حمض الدي إن إيه إلى تنسيق ASCII وتحويلها إلى أرقام ثنائية. تقدم الخطوة الخامسة عنصرًا ديناميكيًا من خلال تحويل الأرقام الثنائية باستخدام مفتاح عشوائي. تتضمن الخطوة النهائية تضمين هذه الأرقام الثنائية (النص المشفر) بشكل مخفي داخل صورة. وبالإضافة إلى المقاييس التقليدية، تم دمج عناصر تعلم الآلة لرفع أداء النظام. تم إجراء تقييم الأداء عبر ثلاث صور قياسية بأحجام بيانات متفاوتة، مما أظهر فعالية النظام. عرض النظام المقترح أوقاتًا سريعة للتشفير، حيث كانت أوقات التشفير والفك تبلغ 2.802 ميلي ثانية و 3.388 ميلي ثانية على التوالي. يثري دمج تقنيات تعلم الآلة قدرات النظام، مقدمًا حلاً جاذبًا لنقل البيانات بشكل آمن وفعال عبر الإنترنت.

الكلمات المفتاحية: التشفير الجيني لحمض الدي إن إيه، الاخفاء ، تعلم الآلة، الحماية ، التشفير، فك التشفير.