Journal of Theoretical and Applied Information Technology
15th February 2023. Vol.101. No 3
© 2023 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENHANCING STANDARD ENCRYPTION ALGORITHMS USING MULTILAYERS ENCRYPTION TECHNIQUE

SHAIMA ALFADHLI[1] , KHOLOUD SAK[2] , MOHAMMED ALWAKEEL[3]

[1,2]University of Tabuk, Department of Information Technology, Tabuk, Saudi Arabia

[3]University of Tabuk, Department of Computer Engineering, Tabuk, Saudi Arabia

E-mail: [1] 431010489@stu.ut.edu.sa, [2] 431010522@stu.ut.edu.sa, [3] alwakeel@ut.edu.sa

ABSTRACT

There are several standard encryption algorithms used to encrypt text to protect it from unauthorized access, however due to advance in technology it has become easier for attackers to decrypt the encrypted text and gain an unauthorized access to the original text. Therefore, it has become a necessity to develop new schemes that increase the level of security of the standard encryption algorithms. In this research, we will introduce a new multilayers encryption/decryption technique that can be used to enhance any standard encryption algorithm. In particular, the proposed technique consists of four layers, in the first layer, the original text is anonymously randomized using a private randomization key, then in the second layer, a standard encryption algorithm is used to encrypt the randomized text resulting from the first layer, in the third layer, the features of a private image (image that is known only by the transmitter and the receiver) are extracted using artificial intelligent technique, and then these features are used as a private key to encrypt the text resulting from the previous layer, finally, in the fourth layer, the encrypted text resulting from the third layer will be hidden in a cover image using standard steganography technique and transmitted to the receiver. At the receiver side, the same steps will be executed in reverse order to get the original text. The proposed technique is expected to increase the security level of the standard encryptions algorithm, and makes it more complex for the attacker to decrypt the text.

Keywords: *Encryption Algorithm, Multilayers Encryption, Anonymous Randomization, Text Randomization, Image Steganography*

1. INTRODUCTION

Information and data are commodities that must be protected from unauthorized access; therefore, data security has become an issue of most importance due to the fact that an attacker may have an unauthorized access or can eavesdrop on essential and confidential information. Hence, encrypting the data is fundamental to secure it, and in order to protect the data, standard encryption algorithms such as AES or DES are utilized as cryptographic algorithms [1]. The cryptography algorithms in general are classified into three classes, symmetric key cryptography, asymmetric key cryptography and hash function. In symmetric encryption, one key is used for both the encryption and decryption processes, using a single private key is easy to implement but extra care must be taken to ensure that this key is kept from being accessed by an unauthorized entity. On the other hand, asymmetric encryption uses two keys, public key which is shared between all system's users and is used by the sender to encrypt the data, and a private key which is known and used by the receiver only to decrypt the received data [2-5]. It is obvious that the asymmetric key technique compared to the symmetric key technique requires more processing time as it needs more mathematical operations to some extent [6]. In addition to encryption, steganography which is a method to hide one document, text, image or video within another document, text, image or video is used to increase the level of data security. Steganography typically involves hiding sensitive information beneath an authentic cover image called a stego image such that the hidden information cannot be seen by human eyes, the stego image is then sent to the receiver, once it is received at the receiver side a decomposition or decryption process is executed to obtain the sensitive hidden information [8-11]. There are several steganography techniques in literature including Least Significant Bit (LSB) substitution, Random insertion LSB, Spatial LSB domain technique, …etc. [12-15].

In this research we present a new multilayers encryption/decryption technique that combine anonymous text randomization, standard encryption, artificial intelligent encryption, and steganography to enhance and increase the security level of any standard encryption algorithm.

2. LITERATURE REVIEW

Even though there are very few multilayers' techniques for encryption in literature, there exist extensive amount of literature studies of encryption algorithms and studies of steganography techniques. In these literatures all aspects of encryption algorithms and steganography techniques were investigated. In the following subsections some of these literatures related to our research are reviewed.

2.1 Encryption Algorithms Literatures

The authors in [2] implemented and analyzed several cryptographic algorithms including symmetric and asymmetric algorithms such as DES, AES, Blowfish, RSA, and 3DES, the analysis of these algorithms include performance evaluation, weaknesses, and strengths. Based on that a comparison between the presented algorithms is carried out. Several evaluation metrics were used in this research including encryption time, decryption time, memory used, number of bits required for encoding optimally, and avalanche effect. The results show that each algorithm has its strength and weakness points, and the used algorithm in any system should be selected based on the demands of the application that will be used, for example the results show that Blowfish algorithm is the best algorithm that can be used in terms of time and memory where it requires the shortest time among the studied algorithms and requires less memory, on the other hand AES gives the best performance in term of confidentiality and integrity, while DES is the best choice if we consider demand on network's bandwidth.

In [3] Panda also studied the performance of several encryption algorithms namely Blowfish, DES, RSA, and AES using several files' types such as text, image, and Binary files. In his research several evaluation's parameters such as decryption time, encryption time, and throughput were used to compare between the algorithms, and a simulation were developed. The results show that the performance of AES overcome other algorithms' performance in term of the evaluation's parameters.

Performance evaluation of symmetric encryption algorithms were presented in [4], where the performance of AES, DES, Triple DES, and Blowfish were investigated in terms of encryption and decryption time, the results indicate that AES outperform other algorithms, and in the second place there was Blowfish algorithm while the Triple DES algorithm has the worst performance.

In [5] a survey of several encryption algorithms such as DES, Triple DES, and AES was presented, the study concluded that based on the survey AES has the best performance in term of time, speed, key length, and throughput.

Elminaam, D., et al. in [6] also presented performance evaluation of six encryption algorithms namely: AES, RC2, DES, 3DES, Blowfish, and RC6, and comparisons between these algorithms using simulation were carried out based on several metrics including battery power consumption and encryption/decryption speeds. The simulation results show that whether the results were displayed in base 64 encoding or hexadecimal base encoding there is no significant difference in the performance, however in term of changing packet size the Blowfish outperform other algorithms, the simulation also shows that Blowfish, RC2, and RC6 have disadvantages compared to other algorithms in term of changing data type, for example text instead of image.

2.2 Steganography Techniques Literatures

Abd El-Latif. A. et al. in [10] design a novel Quantum walk-based (QW) image steganography mechanism for cloud systems that is based on quantum concepts and has three steganography algorithms, basically, the main idea is to hide the data into host media without the need of pre-encryption. The presented mechanism was analyzed and the results show that the mechanism has high embedding capacity and high security based on QWs in addition to good visual quality. To secure the data (hide the data) the algorithms relies on quantum behavior of quantum walks and the keystream that is generated from running quantum walks, and to extract the data stego-image and initial key used for running quantum walks on a circle is needed.

The author in [8] highlighted the fundamental components of a convolutional neural network and discussed memory and time complexity in neural networks, in addition to efficiency's issues. The Author then reviews using deep learning in steganalysis with respect to current methods of steganography, by presenting different neural networks from the period (2015-2018) that have been evaluated using methodology designed specifically for steganalysis. The author concluded that convolutional neural network can and is used widely in steganalysis community.

Dalal. M., & Juneja. M. in [12] presented a new video steganography scheme that may be used for Standard Definition (SD) and High Definition (HD) videos, that was aimed to balance robustness and imperceptibility. The data is encrypted then discrete wavelet transforms (DWTs) is used to hide message in video frames using only the luminance component. Hiding the message in the video frames is done by using the middle frequency subbands, after applying second level discrete wavelet transformations to the video frames to break them down into 16 sub-bands. The effectiveness of the suggested technique is evaluated on several videos using bit error rate (BER), peak signal to noise ratio (PSNR), and structural similarity index. The approach is additionally tested for various levels of compression on stego-video. The results show that the proposed system can achieve great imperceptibility for both HD and SD videos, additionally, it offers robustness to various noise attacks and compression levels, which makes the presented scheme a promising technique to secure data transmission.

The proposed work in [9], with the help of machine learning algorithms, aims to improve edge-based image steganography and offers a greater payload capacity and imperceptibility. The method applies an adaptive embedding procedure over the subband Dual-Tree Complex Wavelet Transform (DT-CWT) coefficients. Machine learning-based optimization approaches are used to embed the data with the lowest possible retrieval error. A secret key that is generated during the embedding process is required for data retrieval and must be sent to the receiver over a secure connection. Standard benchmark parameters including Peak signal-to-noise ratio (PSNR), Correlation factor (CF), Retrieval error, Structural Similarity Index Measure (SSIM), Histogram, and Bits Per Pixel (BPP) are used to assess the algorithm's performance. Even with embedding of up to 7.87 BPP, the stego-image PSNR remains 50dB which demonstrates unequivocally that the proposed technique greatly outperforms the most advanced image steganographic systems.

A secure halftone image steganographic method based on pair swapping to minimize embedding distortion was presented by Liu, W. et al. in [11]. In contrast to the majority of other studies, the suggested technique does not include a master-slave relationship, and the choice of pixel pairings rather than slave pixels determines the steganographic performance. The superiority of pair switching was proved based on a human visual system (HVS) model of halftone images, and vertical swapping is shown to be the best pair swapping strategy for enhancing visual quality. After developing a statistical model to forecast the vertical pair pattern by taking into account its surrounding area, a distortion assessment is then proposed to assess the embedding distortions on both statistics and vision. To minimize the embedding distortions and take advantage of the distortion measurement, syndrome-trellis code (STC) is used. The suggested steganography technique delivers strong statistical security with large embedding capacity without reducing the visual quality, according to the obtained results.

Kaur. H. & Rani. J. in [13] described and examined several steganography methods and data retrieval techniques. the research concludes that some algorithms have law embedding capacity with high temporal complexity, therefore, it is necessary to create effective and precise steganography algorithms, either by fusing already existing approaches or by creating brand-new ones.

In [14] the authors compared three different steganographic techniques: Least Significant Bit (LSB), Discrete Cosine Transform (DCT) steganography, and Discrete Wavelet Transform (DWT) steganography.

Aggarwal in [15] explored image steganography and demonstrated it using the LSB approach, in which the least significant bit of each byte is changed to create the bit-string representing the embedded file. Since changing the LSB only results in slight color changes, it is typically not perceptible to the naked eye, however, steganography has not been as effective when using an 8-bit color image file as it has been when using a 24-bit color image file due to the restrictions in color variations and the need of a color-map.

3. PROBLEM STATEMENT AND RESEARCH QUESTIONS

From section 2, one can conclude that the standard RSA, DES, AES, etc are the most commonly used encryption techniques, and in their standard mode the generated encrypted text for the same original text and encryption key is always the same. Even when modes are used in these standard techniques such as Cipher Block Chaining mode or Cipher FeedBack mode to randomize the text, the randomization is performed using an initialization vector which prevents a sequence of plaintext that's identical to a previous plaintext sequence from producing the same encrypted text, however the initialization vector in these modes is not kept secret and does not consider as a private randomization key. The literatures also show that almost all of the presented methods were based on a single layer technique to encrypt text, and the use of multilayers techniques to enhance the performance of the standard encryption technique is very limited. Even when multilayers technique was used, any additional layer cannot be used solely to encrypt a text as a standalone technique, which does not provide the flexibility to achieve the balance between the desired level of system's security and overhead or complexity. Hence, this research aims to answer the following questions:

- Can we generate variant random encrypted text for an original text then retrieve the original plain text?
- What are the additional layers that can be developed or added to the standard encryption algorithm to enhance the security level.
- Can any layer of the additional layers that were developed (or added) be used solely to encrypt a text as a standalone technique.
- Can we mix any number of the suggested layers in this study to achieve the balance between the desired level of security and system's overhead or complexity.

4. THE CONTRIBUTIONS OF THE PROPOSED TECHNIQUE

The main contributions of this research are as follows:

- A Novel multilayers data encryption /decryption technique is presented to enhance the standard encryption algorithms.
- Several novel anonymous text's randomization techniques that can be used to encrypt text were introduced, unlike conventional randomization used in some modes of standard encryption, the proposed randomization techniques are anonymous and each character in the original text is randomized individually based on Private Random Key (PRK). These randomization techniques can be used as a standalone technique to encrypt text, and can also be combined with any standard encryption algorithm or with other layers presented in this work to enhance the security level of the encryption.
- Each layer in the proposed technique may be used as a standalone technique to encrypt/decrypt text.
- The user of the proposed technique may mix any number of the proposed layers to encrypt/decrypt text based on the desired level of security and complexity.

5. SUGGESTED SOLUTION

The proposed multilayers technique that can be used to enhance the standard encryption algorithms combines anonymous text randomization, standard encryption, artificial intelligent encryption, and steganography, hence the proposed technique consists of four layers namely:

1) Anonymous randomization layer, where the original text is anonymously randomized using a private randomization key.
2) Standard encryption layer, where a standard encryption algorithm such as DES is used to encrypt the randomized text resulting from the first layer.
3) Artificial intelligent (A.I.) encryption layer, where the features of a private image (image that is known only by the sender and the receiver) are extracted using standard artificial intelligent technique, and then these features are used as a private key to encrypt the text resulting from the previous layer.
4) Image steganography layer, where the encrypted text resulting from the third layer will be hidden in an image using a standard image steganography technique.

In the presented technique standard encryption and standard steganography are part of it, hence the overhead of the proposed technique will be higher than the overhead of standard encryption or steganography in term of encryption/decryption time, speed, complexity, etc. However, the proposed technique is expected to provide higher security level than standard encryption and standard steganography, since the unauthorized entity need to decrypt text from each layer including standard encryption layer, standard steganography layer, and the additional layers before getting the original text. In addition to that, each layer in the proposed technique can be used as a standalone encryption method, and the user has the liberty to use and mix any number of the proposed layers to reach the desire level of security and complexity, where sometimes higher security level is required for sensitive data or information, even though achieving this will increase the overhead, hence the user my trade off the security level with system's complexity and the overhead by adding or removing some of the layers.

We assume for the proposed technique that there are specifically one hundred characters that can be used to write the original text regardless of its length, and each character has its unique index, as shown in Table 1. In the following subsections the functions of each layer are illustrated.

Table 1: Characters Set Used to Write Original Text

Index	Char.	Index	Char.	Index	Char.	Index	Char.	Index	Char.
1		21	4	41	H	61	\\	81	p
2	!	22	5	42	I	62	]	82	q
3	"	23	6	43	J	63	^	83	r
4	#	24	7	44	K	64	_	84	s
5	$	25	8	45	L	65	`	85	t
6	%	26	9	46	M	66	a	86	u
7	&	27	:	47	N	67	b	87	v
8	'	28	;	48	O	68	c	88	w
9	(	29	<	49	P	69	d	89	x
10	)	30	=	50	Q	70	e	90	y
11	*	31	>	51	R	71	f	91	z
12	+	32	?	52	S	72	g	92	{
13	,	33	@	53	T	73	h	93	|
14	-	34	A	54	U	74	i	94	}
15	.	35	B	55	V	75	j	95	~
16	/	36	C	56	W	76	k	96	¢
17	0	37	D	57	X	77	l	97	£
18	1	38	E	58	Y	78	m	98	¥
19	2	39	F	59	Z	79	n	99	±
20	3	40	G	60	[	80	o		

5.1 Encryption Technique

The original text will be written using characters from the characters set shown in Table 1, then it is processed in each encryption layer mentioned above as follow:

5.1.1 Anonymous randomization layer

In this layer the original text will be randomized anonymously using one of the four randomization algorithms illustrated below, in each algorithm a Private Randomization Key (PRK) is used and each character from the original text will be converted into four random characters. Unlike conventional randomization used in some modes of standard encryption such as Cipher Block Chaining (CBC) mode or Cipher FeedBack (CFB) mode, which relies on initialization vector that is attached and sent without encryption with the message, our randomization technique is anonymous and each character in the original text is randomized individually based on PRK which is not sent with the message.

In order to show the effectiveness of the randomization techniques, let us define the Randomization Effectiveness Factor (REF) as the total number of trials that an unauthorized entity (the attacker) needs to try to remove randomization (derandomization) from the text and get the original text using brute force attack (i.e. the number of all possible phrases until the correct phrase is found), note that as REF approaches infinity (∞) it is impossible for the attacker to derandomize the text.

To use brute force attack to derandomize the text, the attacker needs to know three elements, namely:

1) The exact algorithm that was used to perform the randomization (including the equations and the exact steps that were used in the algorithm).
2) The characters set that was used to generate the original text (i.e. all the characters in the set and the index of each character).
3) The PRK that was used in the algorithm.

We presented in the following subsections four different randomization algorithms, each depends on reversable linear equations, and since the number of reversable linear equations that may be used in the algorithms are infinite then the number of algorithms that may be used are also infinite, hence; ($REF \approx \infty$) and it is impossible for the attacker to derandomize the text without knowing the exact algorithm that was used to randomize the original text. However, even if the attacker knows the exact algorithm that was used for randomization without knowing the characters set and PRK, the REF is still a very huge number. To show that, let us assume that the attacker knows exactly the randomization algorithm and the length of the characters set (α) that were used, so he needs to find the exact characters set (i.e. all the characters in the set and the index of each character; one sample of characters set is shown in Table 1), and the RPK using brute force to derandomize the text. The number of possible exact characters sets with length (α) is defined in (1),

$$\alpha(\alpha-1)(\alpha-1)\ldots.(2)(1)=\alpha! \qquad (1)$$

and the possible number PRKs is at least equals to (α-*1*), which depends on the reversable randomization linear equations that was used in the randomization algorithm. Hence REF can be calculated as shown in (2),

$$REF_{(\alpha)} = (\alpha!)(\ \alpha\text{-}1). \qquad (2)$$

Hence, for a characters set similar to the one shown in Table 1 where (α = *100*), the Randomization Effectiveness Factor is $REF_{(100)}$ = (*100!*)(*100-1*) ≈ *9.24e*(*+159*), which is a very huge number, this implies that it is almost impossible for the attacker to derandomize the text using brute force in an acceptable period of time even; though the attacker knows the exact randomization algorithm that was used.

5.1.1.1 Randomization algorithm 1

In this algorithm, for each character in the original text a random integer (R) between 1 and 99 is generated, and used with the index of the original character and the PRK as parameters in four equations to calculate four random numbers (X,Y,Z, and W), which are used as indices of four characters from the characters set in Table 1, hence each original character will be represented by four random characters set, namely the characters set that has the indices [X, Y, Z, and W]. Table 2 below shows the detailed steps of this algorithm.

Table 2: Anonymous Randomization Algorithm No. 1

Randomization algorithm 1:	
Input: Original plaintext, characters set table (shown in Table 1.), PRK	
Output: Anonymous randomize text (each original character is represented by four random characters)	
1	for i=0 ➔ length (Original Plaintext)
2	get character from Original Plaintext [i] ➔ S1
3	get the index of S1 from the characters set table ➔ n
4	generated the random number from 1 to 99 ➔ R
5	X = Remainder of (n / R)
6	Y = Quotient of (n / R)
7	Z = Remainder of (R / PRK)
8	W = Quotient of (R / PRK)
9	Result = get character from the characters set table [X] + get character from the characters set table [Y] + get character from the characters set table [Z] + get character from the characters set table [W]
10	return Result
11	end for

5.1.1.2 Randomization algorithm 2

In this algorithm, for each character in the original text two random integers (R1 and R2) between 1 and 99 are generated, and used with the index of the original character and the PRK as parameters in two equations to calculate two random numbers (X and Y). Based on the value of (R1), the randomize characters set that represent the original character is formed, where if (R1<50) the randomize characters set will be the characters set that have indices [R1, X, R2, and Y] otherwise the randomize characters set will be the characters set that has indices [R1, X, Y, and R2]. Table 3. below shows the detailed steps of this algorithm.

Table 3: Anonymous Randomization Algorithm No. 2

Randomization algorithm 2:	
Input: Original plaintext, characters set table (shown in Table 1.), PRK	
Output: Anonymous randomize text (each original character is represented by four random characters)	
1	for i=0 ➔ length (Original Plaintext)
2	get character from Original Plaintext [i] ➔ S1
3	get the index of S1 from the characters set table ➔ N
4	generated a random number from 1 to 99 ➔ R1
5	generated a random number from 1 to 99 ➔ R2
6	X = Remainder of (N / PRK)
7	Y = Quotient of (N / PRK)
8	IF R1>0 & R1<50 THEN : Result = get character from the characters set table [R1] +get character from the characters set table [X] +get character from the characters set table [R2] +get character from the characters set table [Y] ELSE : Result = get character from the characters set table [R1] +get character from the characters set table [X] +get character from the characters set table [Y] +get character from the characters set table [R2]
9	return Result
10	end for

5.1.1.3 Randomization algorithm 3

In this algorithm, for each character in the original text a random integer (R) between 1 and 99 is generated, and used with the index of the original character and the PRK as parameters to calculate two random numbers (X and Z), and two variables Y and W are given specific predetermined values. The next step is to check; if (X) is greater than (99) (i.e. outside the range of the characters set table's indices) then Y will be alternated from its predetermined value and a new value of (X) is calculated such that it will be within the range of the characters set table's indices. Also we will check; if (Z) is greater than (99) (i.e. outside the range of the characters set table's indices) then W will be alternated from its predetermined value and a new value of (Z) is calculated such that it will be within the range of the characters set table's indices. Finally the numbers (X, Y, Z, and W) are used as indices of four characters from the characters set in Table 1, hence each original character will be represented by four random characters set, namely the characters set that has the indices [X, Y, Z, and W]. Table 4. below shows the detailed steps of this algorithm.

Table 4: Anonymous Randomization Algorithm No. 3

Randomization algorithm 3:	
Input: Original plaintext, characters set table (shown in Table 1.), PRK	
Output: Anonymous randomize text (each original character is represented by four random characters)	
1	for i ➔ length (Original Plaintext)
2	get character from Original Plaintext[i] ➔ S1
3	get the index of S1 from the characters set table ➔ N
4	generated the random number from 1 to 99 ➔ R
5	X = N+ R
6	Y = 25
7	IF (X>99) THEN : X= X-99 , Y=15
8	Z = R + PRK
9	W = 65
10	IF (Z>99) THEN : Z = Z – 99 , W = 45
11	Result = get character from the characters set table [X] +get a character the characters set table [Y] +get a character from the characters set table [Z] +get a character from the characters set table [W]
12	return Result
13	end for

5.1.1.4 Randomization algorithm 4

In this algorithm, for each character in the original text a random integer (R) between 1 and 99 is generated, and used with the index of the original character as parameters in four equations to calculate four random numbers (X,Y,Z, and W), which are used as parameters in another two equations to calculate two random numbers (N1 and N2), which in turn are used with PRK in four equations to calculate four random numbers (X2, Y2, Z2, and W2), these late four random numbers are used as indices of four characters from the characters set in Table 1, hence each original character will be represented by four random characters set, namely the characters set that has the indices [X2, Y2, Z2, and W2]. Table 5. below shows the detailed steps of this algorithm.

Table 5: Anonymous Randomization Algorithm No. 4

Randomization algorithm 4:	
Input: Original Plaintext, characters set table (shown in Table 1.), PRK	
Output: Anonymous randomize text (each original character is represented by four random characters)	
1	for i ➔ length (Original Plaintext)
2	get character from original Plaintext[i] ➔ S1
3	get the index of S1 from the characters set table ➔ N
4	generated the random number from 1 to 99 ➔ R
5	X = Remainder of (N / 10)
6	Y = Quotient of (R / 10)
7	Z = Quotient of (N / 10)
8	W = Remainder of (R / 10)
9	N1 = X * 10 + Y
10	N2 = W * 10 + Z
11	X2 = Remainder of (N1 / PRK)
12	Y2 = Quotient of (N1 / PRK)
13	Z2 = Remainder of (N2 / PRK)
14	W2 = Quotient of (N2 / PRK)
15	Result = get character from the characters set table [X2] + get character from the characters set table [W2] + get character from the characters set table [Y2] + get character from the characters set table [Z2]
16	return Result
17	end for

5.1.2 Standard encryption layer

Any standard encryption algorithm such as DES, AES, RAS …etc. can be used in this layer, and for the propose of illustration we used DES encryption for this layer, where the anonymously randomize text generated from the previous layer is used as an input to this layer and encrypted using standard DES algorithm.

5.1.3 A.I. encryption layer

In this layer any A.I. standard image analysis algorithm can be used to extract the feature of a private image that is known only to the sender and the receiver, and the values of the extracted features from that image are used as a private key to encrypt the text generated from the previous layer using standard XOR logic operation, similar to the XOR operation used in standard encryption algorithm.

5.1.4 Image steganography layer

In this layer any image steganography technique such as LSB and PVD can be used to hide the encrypted text generated from the previous layer. For illustration propose in this paper we used standard LSB that was presented in [15], and the result generated from this layer is the final encrypted message that will be transmitted to the receiver.

5.2 Decryption Technique

The proposed decryption technique consists of four layers; namely steganography text extracting layer, A.I. decryption layer, standard decryption layer, and derandomization layer. Once the message is received it will be processed by the steganography text extracting layer then it will be passed to the next layers until the original text is obtained as the output of the final layer; namely derandomization layer. In the following subsections the functions of each layer are illustrated.

5.2.1 Steganography text extracting layer

The received message which is the stego image contains the hidden encrypted text is used as an input to this layer, and the hidden encrypted text is extracted from the image using the standard LSB extraction algorithm presented in [15], once the hidden text is extracted it is passed to the next layer (A.I. decryption layer).

5.2.2 A.I. decryption layer

In this layer the same A.I. standard image analysis algorithm that was used in the A.I. encryption layer is used to extract the feature of the private image that is known only to the sender and the receiver, and the values of the extracted features from that image are used as a private key to decrypt the text generated from the previous layer using standard XOR logic operation, similar to the XOR operation used in standard decryption algorithm, and the result is sent to the next layer.

5.2.3 Standard decryption layer

Standard decryption algorithm (that is related to the standard encryption algorithm used during the encryption process at the sender side) is used to decrypt the text received from the previous decryption layer. In this research standard DES decryption is used, since we used standard DES for encryption to encrypt the text in the standard encryption layer. The decrypted text from this layer is used as an input to the final layer namely derandomization layer to get the original text.

5.2.4 Derandomization layer

In section 5.1.1 we presented four different algorithms to anonymously randomize the original text, and in the following subsections a derandomization algorithm for each algorithm presented in section 5.1.1 is illustrated. Recall that all randomization process presented in section 5.1.1 substitute each character from the original text with four random characters, hence, in the derandomization algorithm the randomized text that was received from the previous layer is divided into segments each consists of four characters, and they are used as inputs to the derandomization algorithm where each segment is used to recover one original text's character.

5.2.4.1 Derandomization algorithm 1

This algorithm is used to derandomize the text that was randomized using the algorithm presented in section 5.1.1.1, the steps of this algorithm are illustrated in Table 6.

Table 6: Derandomization Algorithm No. 1

	Derandomization algorithm 1:
	Input: Randomize text (from previous layer), characters set table (shown in Table 1.), PRK
	Output: Original plaintext
1	for i ➔ length Randomize text (from previous layer): step 4 (i.e. i=i+4)
2	get character from Randomize text [i] ➔ S1
3	get character from Randomize text [i+1] ➔ S2
4	get character from Randomize text [i+2] ➔ S3
5	get character from Randomize text [i+3] ➔ S4
6	get the index of S1 from the characters set table ➔ X
7	get the index of S2 from the characters set table ➔ Y
8	get the index of S3 from the characters set table ➔ Z
9	get the index of S4 from the characters set table ➔ W
10	R = W * PRK + Z
11	N = Y* R + X
12	get the original character from the characters set table [N]➔ Original plaintext
13	return Original plaintext
14	end for

5.2.4.2 Derandomization algorithm 2

This algorithm is used to derandomize the text that was randomized using the algorithm presented in section 5.1.1.2, the steps of this algorithm are illustrated in Table 7.

Table 7: Derandomization Algorithm No. 2

Derandomization algorithm 2:	
Input: Randomize text (from previous layer), characters set table (shown in Table 1.), PRK	
Output: Original plaintext	
1	for i ➔ length Randomize text (from previous layer) : step 4 (i.e. i=i+4)
2	get character from Randomize text [i] ➔ S1
3	get character from Randomize text [i+1] ➔ S2
4	get character from Randomize text [i+2] ➔ S3
5	get character from Randomize text [i+3] ➔ S4
6	get the index of S2 from the characters set table ➔ X
7	get the index of S1 from the characters set table ➔ R1
8	IF (R1>0) & (R1<50) THEN : Get the index of S4 from the characters set table ➔ Y ELSE : get the index of S3 from the characters set table ➔ Y
9	N = Y * PRK + X
10	get the original character from characters set table [N]➔ Original Plaintext
11	return Original Plaintext
13	end for

5.2.4.3 Derandomization algorithm 3

This algorithm is used to derandomize the text that was randomized using the algorithm presented in section 5.1.1.3, the steps of this algorithm are illustrated in Table 8.

Table 8: Derandomization Algorithm No. 3.

Derandomization algorithm 3:	
Input: Randomize text (from previous layer), characters set table (shown in Table 1.), PRK	
Output: Original plaintext	
1	for i ➔ length Randomize text (from previous layer) : step 4 (i.e. i=i+4)
2	get character from Randomize text [i] ➔ S1
3	get character from Randomize text [i+1] ➔ S2
4	get character from Randomize text [i+2] ➔ S3
5	get character from Randomize text [i+3] ➔ S4
6	get the index of S1 from the characters set table ➔ X
7	get the index of S2 from the characters set table ➔ Y
8	get the index of S3 from the characters set table ➔ Z
9	get the index of S4 from the characters set table ➔ W
10	IF (W= 45) THEN : Z = Z + 99
11	R = Z - key
12	IF (Y = 15) THEN : X = X +99
13	N = X- R
14	get the original character from characters set table [N]➔ Original Plaintext
15	return Original Plaintext
16	end for

5.2.4.4 Derandomization algorithm 4

This algorithm is used to derandomize the text that was randomized using the algorithm presented in section 5.1.1.4, the steps of this algorithm are illustrated in Table 9.

Table 9: Derandomization Algorithm No. 4

Derandomization algorithm 4:	
Input: Randomize text (from previous layer), characters set table (shown in Table 1.), PRK	
Output: Original plaintext	
1	for i ➔ length Randomize text (from previous layer) : step 4 (i.e. i=i+4)
2	get character from Randomize text [i] ➔ S1
3	get character from Randomize text [i+1] ➔ S2
4	get character from Randomize text [i+2] ➔ S3
5	get character from Randomize text [i+3] ➔ S4
6	get index S1 from the characters set table ➔ X
7	get index S2 from the characters set table ➔ W
8	get index S3 from the characters set table ➔ Y
9	get index S4 from the characters set table ➔ Z
10	N1 = Y * key +X
11	N2 = W * key + Z
12	X = N1 // 10
13	Z = N2 % 10
14	N = Z *10 + X
15	get the original character from characters set table [N]➔ Original Plaintext
16	return Original Plaintext
17	end for

6. IMPLEMENTATION

An Encryption/Decryption software was developed using Python to test the proposed technique, the result of executing the software shows that the security level of the proposed technique outperforms the security level of standard techniques and standard steganography, and the visual inspection of the resulted stego-image does not provide any hint regarding the presence of an encrypted text hidden in it, even though the size of the encrypted text is quadruple the original text size. The next subsections show examples of using the proposed technique to encrypt and decrypt text.

6.1 Encryption Examples

The steps of the encryption using the proposed technique is as follow:

1) Replace each character by four random characters using one of the anonymous randomization algorithms presented in section 5.1.1 (in our examples we used the anonymous

randomization algorithm presented in section 5.1.1.1).

2) Encrypt the randomize text using standard DES algorithm.
3) Encrypt the result from step 2 using features of a private image as a private key, where the image features are extracted using standard A.I. image analysis technique.
4) Use LSB steganography presented in [15] to hide the text resulted from step 3.

6.1.1 Example 1:

Encrypting the word (Cat)

- Get the index of each character in the original text [C ➔ 36, a ➔ 66, t ➔ 85] to perform anonymous randomization.
- Replace each character by four random characters using one of the anonymous randomization algorithms presented in section 5.1.1. (in this example we used the anonymous randomization algorithm presented in section 5.1.1.1) and the result of the anonymous randomization of the word (Cat) is: [C□!$*!+"□J"□]
- Encrypt the randomize text using standard DES algorithm.
- Encrypt the result of the DES encryption using private image features as explain in step 3 above.
- Hide the final encrypted text using LSB steganography.

Figure 1. shows a screen shot of the developed software when executed to encrypt the word (Cat) using the proposed technique.

6.1.2 Example 2:

Encrypting the sentence (Night Sky)

- Get the index of each character in the original text [N,i,g,h,t,□, ➔ 47, 74, 75,73,85, 0], [S,k,y, ➔ 52, 76, 90] to perform anonymous randomization.
- Replace each character by four random characters using one of the anonymous randomization algorithms presented in section 5.1.1. (in this example we used the anonymous randomization presented in section 5.1.1.1) and the result of the anonymous randomization of the sentence (Night Sky) is: [!%)□i□)#0!+"0"&!$!.#□□□#S□-“k□4#"#'!]
- Encrypt the randomize text using standard DES algorithm.
- Encrypt the result of the DES encryption using private image features as explain in step 3 above.
- Hide the final encrypted text using LSB steganography.

Figure 2. shows a screen shot of the developed software when executed to encrypt the sentence (Night Sky) using the proposed technique.

6.2 Decryption Examples

Messages are received as images that contain hidden text, the steps of decryption using the proposed technique is as follow:

1) Use LSB steganography presented in [15] to extract the hidden encrypted text from the stego-image.
2) Decrypt the resulted text from step 1 using features of a private image as a private key, where the image features are extracted using standard A.I. image analysis technique.
3) Decrypt the text resulted from step 2 using standard DES algorithm.
4) Derandomize the decrypted text resulted from step 3 using one of the derandomization algorithms presented in section 5.2.4 (in our examples we used the derandomization algorithm presented in section 5.2.4.1).

Figure 3 and figure 4 show the decryption of the text generated in example 1 presented in section 6.1.1 and example 2 presented in section 6.1.2 respectively. From figure 3 we can see that we successfully retrieved the word (Cat) that was encrypted in example 1, and figure 4 shows that we successfully retrieved the sentence (Night Sky) that was encrypted in example 2.

7. CONCLUSION

A new multilayers encryption technique was proposed in this research, the proposed technique consists of four layers and combined anonymous text randomization, standard encryption, artificial intelligent encryption, and steganography to enhance and increase the security of any standard encryption algorithm. The overhead of the proposed technique will be higher than the overhead of standard encryption or steganography, however, security level will be higher in the proposed technique, in addition to that, since each layer can be used as a standalone encryption method, the user has the liberty to trade off the security level with system complexity and the overhead by adding or removing some of the layers to reach the desire level of security and complexity. Unlike conventional randomization used in some modes of standard encryption, we presented several randomization algorithms that can be used with the proposed technique where the randomization is anonymous and each character in the original text is randomized individually based on a private randomization key which is not sent with the message. In addition, the implementation of the proposed technique is very simple and can be implemented easily using any programming language.

Future research directions of this work may include performance analysis of each layer as a standalone encryption method, based on several performance metrics such as processing time, security level, throughput, encrypted text size.. etc., and compare the results with the performance of standard encryption and steganography methods and other relevant works. The proposed work also can be expanded to incorporate encryption of image, audio, and video as well as other digital data types, in addition to that, a new encryption technique using A.I. may be developed to replace the standard A.I method that was used in A.I. encryption layer to extract the features of a private image and use it as a private key.

AUTHORS' CONTRIBUTIONS:

Conceptualization, M. Alwakeel; Methodology, M. Alwakeel, S. Alfadhli, and K. Sak; Software, S. Alfadhli, and K. Sak; Validation, M. Alwakeel, and S. Alfadhli; Analysis, M. Alwakeel, S. Alfadhli, and K. Sak; Investigation, M. Alwakeel, S. Alfadhli, and K. Sak; Writing and editing, M. Alwakeel, S. Alfadhli, and K. Sak; Supervision, M. Alwakeel; All authors have read and agreed to this version of the manuscript.

REFERENCES:

[1] W. Stalling., Cryptography and network security, Principles and practices. Harlow, England, *Pearson*, 7th edition, 2013.

[2] M. Wahid, A. Ali, B. Esparham, and M. Marwan, "A comparison of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish for guessing attacks prevention," *Journal of Computer Science Applications and Information Technology*, vol. 3, no. 2, pp. 218–230, pp. 1–7, 2018.

[3] M. Panda, "Performance analysis of encryption algorithms for security," in *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES),* pp. 278–284, 2016.

[4] M. Panda, "Performance evaluation of symmetric encryption algorithms for information security," *International Journal of Advanced Research Trends in Engineering and Technology*, Vol. 4, No. 11, pp. 37-41, 2017.

[5] R. Sivakumar, B. Balakumar, and V. Pandeeswaran "A study of encryption algorithms (RSA,DES, 3DES and AES) for information security," *International Research Journal of Engineering and Technology*, Vol. 5, No. 4, pp. 4133-4137, 2018.

[6] D. Elminaam, H. Abdulkader, and M. Hadhoud, "Performance evaluation of symmetric encryption algorithms," *International Journal of Computer Science and Network Security*, Vol. 8, No. 12 , pp.280-286, 2008.

[7] Vaudenay, and A. Serge, classical introduction to cryptography: Applications for communications security. *Springer Science & Business Media*, 2005.

[8] M. Chaumont., "Deep learning in steganography and steganalysis," *Chapter 1 in Digital Media Steganography: Principles, Algorithms, Advances*, Elsevier Book chapter, pp. 1-46, 2019.

[9] I. Kadhim, P. Premaratne, and P. Vial., "High capacity adaptive image steganography with cover region selection using dual-tree complex wavelet transform," *Cognitive Systems Research*, vol. 60, pp, 20-32, https://doi.org/10.1016/j.cogsys.2019.11.002, 2020.

[10] A. A. Abd El-Latif., B. Abd-El-Atty., S. Elseuof., H. Khalifa., A. Alghamdi., K. Polat., and M. Amin, "Secret images transfer in cloud system based on investigating quantum walks in steganography approaches," *Physica A: Statistical Mechanics and its Applications*, vol. 541, pp. 123687, 2020.

[11] W. Liu., X. Yin., W. Lu., J. Zhang., J. Zeng., S. Shi., and M. Mao., "Secure halftone image steganography with minimizing the distortion on pair swapping," *Signal Processing*, vol. 167, https://doi.org/10.1016/j.sigpro.2019.107287 , 2020.

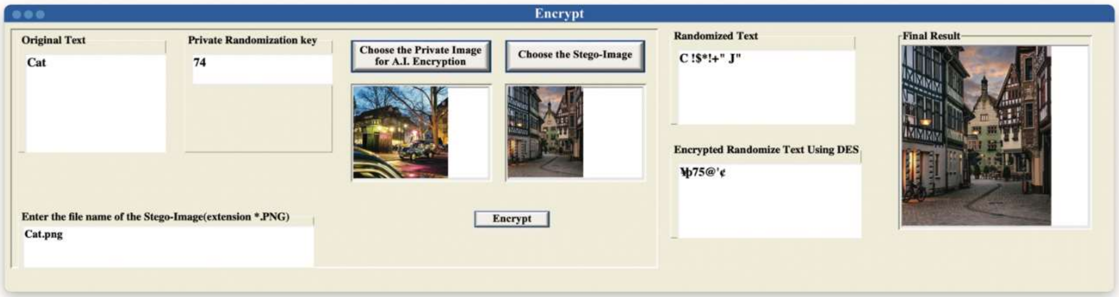
[12] M. Dalal and M. Juneja, "A robust and imperceptible steganography technique for SD and HD videos," *Multimedia Tools and Applications*, vol. 78, no. 5, pp. 5769–5789, 2019.

[13] H. Kaur and J. Rani, "A survey on different techniques of steganography," in *4th International Conference on Advancements in*

Engineering & Technology (ICAET-2016) , Vol. 57, pp. 1–6, 2016 , https://doi.org/10.1051/matecconf/20165702003 .

[14] N. Kaur and S. Behal, “A survey on various types of steganography and analysis of hiding techniques”, *International Journal of Engineering Trends and Technology (IJETT)*, vol. 11, No. 8, pp. 388-392, 2014.

[15] A. Aggarwal, A. Sangal, and A. Varshney, “Image steganography using LSB algorithm”, *International Journal of Information Sciences and Application*, vol. 11, No. 1, (Special Issue), pp. 85-89, 2019.



Original Text
Cat

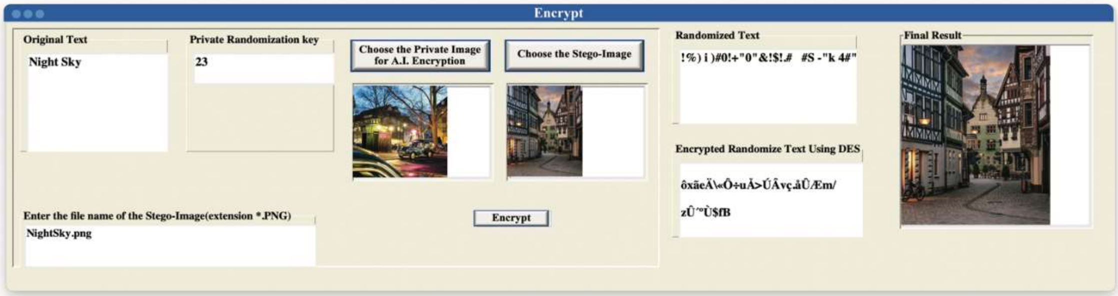
Private Randomization key
74

Enter the file name of the Stego-Image(extension *.PNG)
Cat.png

Randomized Text
C !$*!+" J"

Encrypted Randomize Text Using DES
¥þ75@'¢

Figure 1 Encryption of the Word (Cat)



Original Text
Night Sky

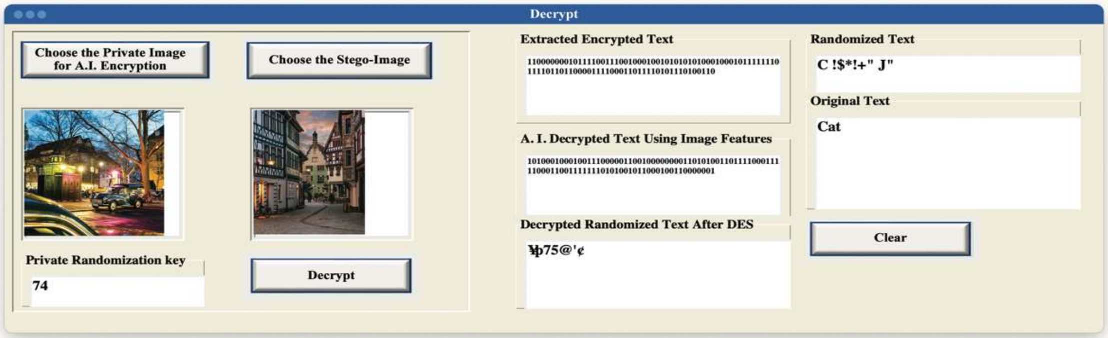
Private Randomization key
23

Enter the file name of the Stego-Image(extension *.PNG)
NightSky.png

Randomized Text
!%) i)#0!+"0"&!$!.# #S -"k 4#"

Encrypted Randomize Text Using DES
ôxãeÄ\«Ô÷uĂ>ÚÂvç.åÛÆm/
zÛ´ºÙ$fB

Figure 2 Encryption of the Sentence (Night Sky)



Private Randomization key
74

Extracted Encrypted Text
1100000001011110011100100010010101010100010001011111110
11110110110000111100011011110101110100110

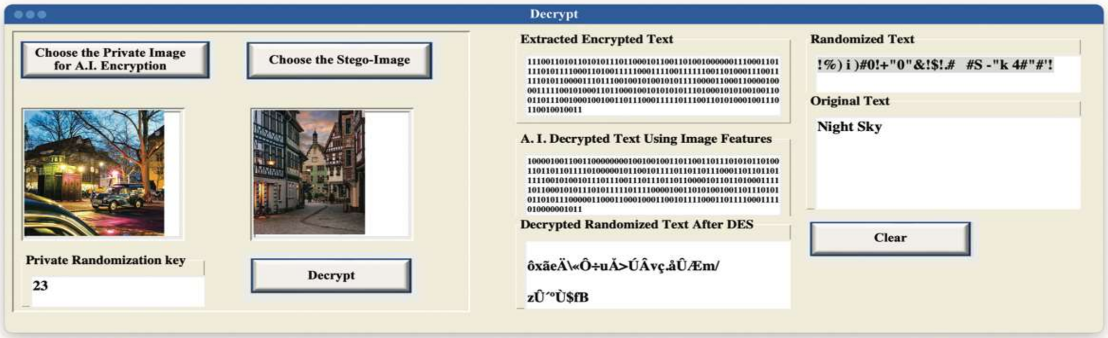
A. I. Decrypted Text Using Image Features
1010001000100111000001100100000000110101001101111000111
1100011001111111010100101100010011000000 1

Decrypted Randomized Text After DES
¥þ75@'¢

Randomized Text
C !$*!+" J"

Original Text
Cat

Figure 3 Decryption of the Word (Cat) that was Encrypted in Example 1 in Section 6.1.1



Private Randomization key
23

Extracted Encrypted Text
1110011010110101011101100010110011010010000000111000110 1
1110101111000110100111110001111001111110011010001110011
1110101100001110111001001010010101111000011000110000100
0011111001010001101100010010101010111010001010100100110
0110111001000100100110111000111110111001101010001001110
110010010011

A. I. Decrypted Text Using Image Features
1000010011001100000000010010010011011001101110101010100
1101101101111010000010110010111101101101110001101101101
1111001010010111011100111011101101100001011011010001111
1011000101011101011111011110000100110101001001101110101
0110101110000011000110001000110010111100011011110001111
010000001011

Decrypted Randomized Text After DES
ôxãeÄ\«Ô÷uĂ>ÚÂvç.åÛÆm/
zÛ´ºÙ$fB

Randomized Text
!%) i)#0!+"0"&!$!.# #S -"k 4#"#'!

Original Text
Night Sky

Figure 4 Decryption of the Sentence (Night Sky) that was Encrypted in Example 2 in Section 6.1.2.