10.31341/jios.47.2.5 UDC 004.056:355.405.2:547.96
Open Access Original Scientific Paper

Using the 3D Protein Structure as Key to Encrypt Images

Mohammed Abbas Fadhil Al-Husainy *dralhusainy@gmail.com*
Department of Cyber Security
Irbid National University, Irbid, Jordan

Hamza A. A. Al-Sewadi *alsewadi@hotmail.com*
Department of Computer Technology Engineering
Iraq University College, Basra, Iraq

Ahmed M. Sayed *ahmedpharma8530@gmail.com*
Department of Pharmacognosy
Nahda University, Beni-Suef, Egypt

Abstract

In the digital world, information is exposed anytime, anywhere, to everybody, hence its privacy is a crucial matter. No matter how complicated the encryption algorithm is, it requires a strong and hard-to-break encryption key. Since the three-dimensional protein sequence structures are usually highly conserved, even better than DNA sequences, this work presents an innovative scheme for implementing the protein sequence and structural to build protein data tables that are then used to generate extremely strong encryption-keys. An image-encryption scheme designed to implement such encryption keys is developed and produced compatible security strength with existing encryption schemes. Prototype experiments resulted in an average normalized mean absolute error of 66.84%, an average peak signal to noise ratio of 6.85 dB, and comparable entropy with other cryptosystems. The obtained results make this scheme a promising color image protection technique for various applications.

Keywords: Image-encryption, Data-security, Cryptography, Protein Structure

1. Introduction

The fast advancement of digital computing capabilities and communications networks has resulted in an ever-increasing volume of multimedia data (i.e., images, video, and audio) being stored or transmitted across insecure communication channels. Because user data typically contains confidential and private information, their security has become critical for protecting and defending users against various harmful assaults, such as preventing data loss and ensuring data integrity. Many technologies, such as steganography [1], [2], watermarking [3], and encryption are currently suitable for assuring a high level of security for medical images. Encryption techniques can turn an identifiable image into something unrecognizable, defending the security of personal information on public networks from malicious assaults [1].

Any cryptosystem consists of two main components; an encryption algorithm and encryption keys. The currently available cryptosystems are classified into two types: symmetric cryptosystems require the same secret key for both encryption and decryption. And, asymmetric cryptosystems require a pair of keys, one for encryption (public key) and one for decryption (private key). The symmetric cryptosystem is faster and achieves a satisfactory level of security but suffers from the key distribution problem, while the asymmetric cryptosystem is slower and solves the key distribution problem, sometimes attackers may exploit the existence of a relationship between the pair of keys to break the security of the cryptosystem. No matter how hard and sophisticated the encryption algorithm is, it is usually publicly available. Hence, the encryption key represents the most important factor, as it is the only secret component in the system for providing strong information protection. Therefore, a strong and hard-to-guess encryption key is sought, and researchers are continuously searching for new approaches to generate strong and hard-to-break keys. The main feature that determines the key strength is its key-space. Taking design cues from biological systems and processes has resulted in some beneficial problem-solving approaches [4], [5]. Previous studies of biology-inspired computing, such as the artificial immune system and ant colony optimization, have focused on extracting computing models from a high-level understanding of biological processes. Recently, many academics have begun to investigate biology-inspired computing at a much lower level, gaining insights into the evolutionary roots of biological information systems by studying how small cellular organisms' function [6]. Accordingly, a number of biology-inspired cryptographic approaches have also been introduced. For example, deoxyribonucleic acid (DNA) -based cryptography is an emerging cryptographic field as a result of research into DNA computing. DNA computing technology has been advanced and is now being employed in various applications, including cryptography [7].

The motivation of utilizing the three-dimensional (3D) proteins to generate secure encryption key since their structures are extremely conserved, even better than DNA sequence, from which data tables can be constructed using their sequence alignment and template structural similarity. Besides, it is anticipated that such implementation could be suitable for secure data processing, storage, and communication applications. Hence, the work in this paper presents a novel protein-based cryptographic method that was inspired by the complex and random structures of natural proteins. It is an image encryption technique adopting a symmetric cryptographic system based on the 3D-protein structure features, where the encryption keys are generated by utilizing these features.

Accordingly, this presented cryptographic method is more efficient than those previously described e.g., DNA-based ones.

After this brief definition in section 1, section 2 gives a brief explanation of the 3D protein structure, then Section 3 summarizes the related work review. Section 4 outlines the methodology of the proposed image cryptography algorithm. Section 5 describes the implementation of the algorithm, lists experimental results, and includes the discussion. Finally, the conclusion of the paper is presented in Section 6.

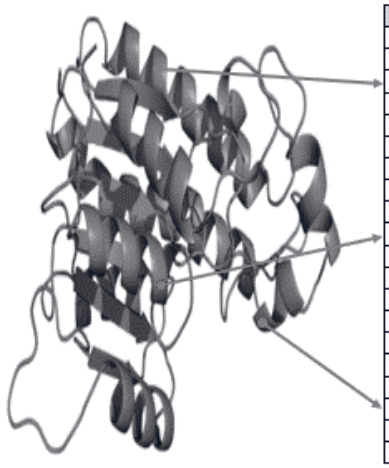
2. 3D Protein Structure

To clarify how to utilize the three-dimensional (3D) structure of a protein to generate and implement the secret key for the proposed image-encryption technique, a brief introduction is given below.

It has been shown that 3D protein structure is evolutionarily more conserved than expected based on sequence conservation alone [8]. The sequence alignment and template structure are used to produce a structural model of the target. Since the protein structures are usually more conserved than DNA sequences, detectable sequence similarity levels usually imply significant structural similarity [9].

Protein structure is a polymer of amino acids (i.e., polypeptide) with secondary, tertiary, and quaternary complex structures; an example of the 3D protein structure is shown in Fig 1-a. Structures of different proteins depend mainly on their amino acid sequence, where 20 different amino acids are employed as building blocks, see Fig 1-b [10].

Generally, proteins are expressions of DNA genetic codes, and hence, they have the same natural randomness as DNA. Additionally, they have other elements of randomness and complexity resulting from their secondary, tertiary, and quaternary structures. As a result, there are no two identical proteins in nature [10], [11].
Random Sequence Complexity (RSC), Ordered Sequence Complexity (OSC), and Functional Sequence Complexity (FSC) are three qualitative features of protein linear sequence complexity defined by Abel and Trevors [21]. Accordingly, proteins can be considered highly complex, random and chaotic. Using such complex and chaotic systems in cryptography will be of great interest, particularly if it is combined with other efficient cryptographic methods. Additionally, this protein-based encryption method is superior to that of the previously reported DNA-based one in utilizing the very complex 3D structure of proteins in the encryption algorithm.



(a)

Amino acid	Atom	X	Y	Z
MET	N	27.34	24.43	2.614
MET	CA	26.266	25.413	2.842
MET	C	26.913	26.639	3.531
MET	O	27.886	26.463	4.263
:	:	:	:	:
MET	CE	24.447	23.984	7.62
GLN	N	26.335	27.77	3.258
GLN	CA	26.85	29.021	3.898
GLN	C	26.1	29.253	5.202
GLN	O	24.865	29.024	5.33
GLN	CB	26.733	30.148	2.905
GLN	CG1	26.882	31.546	3.409
:	:	:	:	:
ILE	N	26.849	29.656	6.217
ILE	CA	26.235	30.058	7.497
ILE	C	26.882	31.428	7.862
ILE	O	27.906	31.711	7.264
ILE	CB	26.344	29.05	8.645
ILE	CG1	27.81	28.748	8.999
:	:	:	:	:

(b)

Figure 1. (a) An example of the protein structure, (b) the detailed representation of the protein in the PDB structure file. (MET: methionine, GLN: Glutamine, ILE: Isoleucine).

3. Related Works

DNA was utilized in the field of cryptography and steganography. For example, Gehani et al. in 2000 [12] introduced this concept with XOR operation to produce strong indexed, random key strings. Then several schemes were reported for incorporating the DNA concept for cryptography, as summarized Al-Husainy et al. [13]. These schemes were secret-key cryptosystems such as the so-called molecular sticker developed by Chen J. [14] and Bibhash Roy et al. [15], while others were public-key cryptosystems, such as the one developed by Tanaka K. [16] and Cui G. et al. [17]. Moreover, many DNA-based image-encryption schemes were developed, see Al-Husainy et al. [13], Zhang et al. [18], and Khan et al. [19].

Image encryption through ribonucleic acid (RNA) sequence approach assisted with neural key sequences is reported by Davi et al. [20]. It transcripts messenger ribonucleic acid (mRNA) sequence to train a binary associative memory neural network (BAM NN), producing transfer ribonucleic acid (tRNA) sequences. Then these sequences provide the processes of change and diffusion for the pixels in colour digital imaging and communications in medical images. The authors claim considerable robustness through various decryption quality analyses, statistics and differential attack analyses.

Herein, we introduced a novel protein-based image encryption scheme inspired by the natural protein due to its random and complex properties. It is based on the three-dimensional protein structure features. Considerably strong encryption keys were generated and tested for the designed encryption scheme. This design benefits from the protein databases currently available to the public, these databases contain thousands to millions of different protein structures, such as the protein data bank ([21]; https://www.rcsb.org/) and Alpha Fold protein structure database ([22]; https://alphafold.ebi.ac.uk/). Hence, users can easily access these protein structures.

Accordingly, the combination of genetically acquired and structural-based codes of proteins makes them an exceptional biology-inspired coding and encryption model. The protein-based code that the end-user can use as a key or generate a secret key for a cryptographic system can be extracted from the protein structure file in the Protein Data Bank format (PDB). Such structure files can be easily downloaded from any protein database and represented in a table as listed in Fig 1-b. The most important five columns of data will be used to construct a protein-based key consisting of five different parameters. The first column represents the amino acid sequence in the chosen protein. Each one is referred to by three letters and represents the corresponding genetic code in the DNA.

The second column is the amino acids elements, which represents the elemental composition of each amino acid in the protein sequence. The alphabet, composed of 38 different characters, represents the type and position of each element in each amino acid in the protein sequence. The last three columns are the X, Y, and Z coordinates of each character (i.e., element) in the previous column within the 3D protein structure in Fig 1-a.

In the proposed study, different proteins will be used as keys; ubiquitin, Pim kinase, and DNA gyrase. These three proteins represent three different structures of different sizes and are hosted in Protein Data Bank.
Taken together, this protein structure-based five-as columns key can be considered a very strong and readily available key model to start our encryption algorithm with. Additionally, each obtained key is unique, as are no two identical keys in all currently available databases. Even similar proteins, of the same sequences, differ slightly in their structures (i.e., The coordinates of each amino acid).

4. Proposed Encryption Technique

In the proposed technique, the table of Fig 1-b will be used as the primary key and shall be filled by the user. The first two parameters; the amino acid sequence in the chosen protein (i.e., MET, GLN, and ILE) and the elemental composition of each amino acid in the protein sequence (i.e., N, CA, C, O, CB, GB1) will be used to perform the substitution operation. The last three parameters (i.e., The coordinates X, Y, and Z) will be used to perform the transposition operation.

The user of the proposed encryption technique enters two digital files: the source color image S to be encrypted and the protein data table shown in Fig. 1-b, which is used as the primary key K to encrypt the source image.
The source color image S is treated as a 3D matrix of red, green, and blue colors, and each color is treated as a byte. To describe the technique, consider the simple example of the 3D representation of the color image illustrated Fig. 2, where (a) an example of a source image and (b) the image pixel diagram showing a pixel value in the red, green, and blue (RGB) matrix.

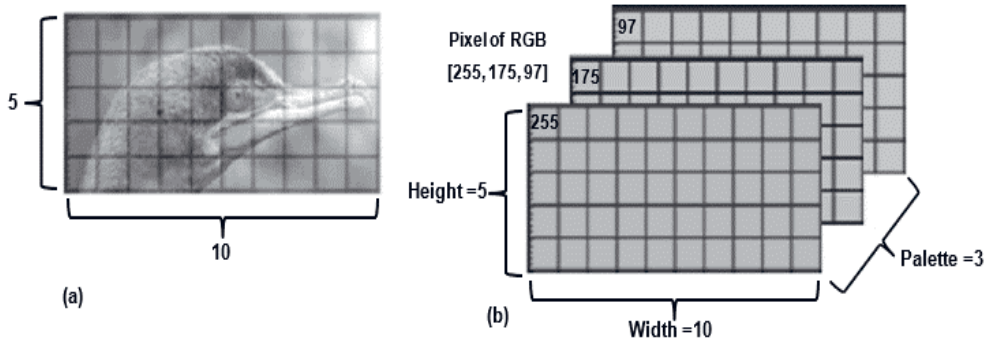


Figure 2. (a) Source image, (b) Illustration of a pixel and its value in the RGB matrix.

4.1. Preparation Stage

Initially, the size (in byte) of the source image S is calculated using Equation (1), where Palette equals 3 for RGB color image.

$$n = \text{Width} \times \text{Height} \times \text{Palette} \quad (1)$$

Change the size of the primary key (protein) *K* so that it has a number of rows equal to the *n*. Two cases have to be dealt with:

- If the size of $K < n$, add more rows by repeating the rows starting with the first row in the protein table.
- If the size of $K > n$, delete the unnecessary rows at the end of the protein table.

Use *n* as a seed for a pseudorandom number generator used in the proposed encryption technique to:

- Assign a unique non-repeating random value between (0...255) for each different amino acid as shown in Table 1.
- Assign a unique non-repeating random value between (0...255) for each different atom as shown in Table 2.

Generate a random sequence of row numbers in the key (protein) data table *K*. This will produce a new protein table containing a random sequence of rows as shown in Table 3.

Change the range of values for the *X*, *Y*, and *Z* coordinates in the protein data table *K* to equal the Width, Height, and Palette ranges, respectively for the source image *S*. Then use equations (2), (3), and (4) to determine new values for each *X, Y*, and *Z*.

$$X_{new} = Round\left(\frac{X_{old} - Min_X}{Max_X - Min_X} \times Width\right) \quad (2)$$

$$Y_{new} = Round\left(\frac{Y_{old} - Min_Y}{Max_Y - Min_Y} \times Height\right) \quad (3)$$

$$Z_{new} = Round\left(\frac{Z_{old} - Min_Z}{Max_Z - Min_Z} \times Palette\right) \quad (4)$$

4.2. Encryption Stage

Two main operations (Substitution and Transposition) are performed to encrypt the source image *S* and create the encrypted image *E*. The proposed encryption technique treats the source image *S* as a single 3D block and implements the substitution and transposition operations on every byte of the source image *S*. The key (protein) data table *K* produced from the preparation stage is used in these two operations.

1. Substitution Operation

Two different implementations of the substitution operation are performed on each byte of the source image S. These implementations are explained below:

A. XOR Operation with Amino Acid Code

In the 3D block of the source image *S*, perform the XOR logical operation between each byte in *S* with the amino acid code in the key (protein) data table *K*. The following programming code shows the substitution operation.

Amino acid		Amino acid Code between (0…255)
Name	Abbreviation	
Alanine	ALA	237
Arginine	ARG	82
Asparagine	ASN	242
Aspartic acid	ASP	171
Cysteine	CYS	189
Glutamine	GLN	24
Glutamic acid	GLU	162
Glycine	GLY	112
Histidine	HIS	60
Isoleucine	ILE	15
Leucine	LEU	137
Lysine	LYS	179
Methionine	MET	160
Phenylalanine	PHE	208
Proline	PRO	55
Serine	SER	249
Threonine	THR	4
Tryptophan	TRP	119
Tyrosine	TYR	17
Valine	VAL	38
Asparagine	ASX	78
Glutamine	GLX	183
Any amino acid	XAA	81
Isoleucine	XLE	184

Table 1. Amino acid abbreviations and their random code.

Amino acid	Atom	X	Y	Z
15	89	26.235	30.058	7.497
162	82	24.865	29.024	5.33
4	222	26.882	31.546	3.409
208	19	27.886	26.463	4.263
189	219	26.85	29.021	3.898
4	82	27.906	31.711	7.264
:	:	:	:	:

Table 3. The key (protein) data table *K* contains a random sequence of rows.

Atom Abbreviation	Atom Code between (0…255)
N	81
C	167
O	82
CB	165
CG	19
CD	32
NE	127
CZ	163
NH1	216
NH2	126
OD1	114
ND2	75
OD2	41
SG	209
OE1	130
OE2	159
NE2	92
ND1	235
CE1	148
CD2	46
OD	103
CG1	55
CD1	207
CG2	195
CE	222
NZ	2
SD	0
CE2	26
CZ	245
CA	89
OG	219
OG1	244
NE1	60
CZ2	96
CH2	56
CZ3	35
CE3	239
OH	49

Table 2. Amino acid abbreviations and their random code.

```
K_index = 0
For (X: 0 ... Width_S -1)
  For (Y: 0 ... Height_S -1)
    For (Z: 0 ... Palette_S -1)
S (X, Y, Z) = S (X, Y, Z)  XOR  K_Amino acid[K_index]
K_index = K_index + 1
```

B. XOR Operation with Atom Code

In the 3D block of the source image *S*, perform the XOR logical operation between each byte in *S* with the one of the atom code in the key (protein) data table *K*. The following programming code shows the substitution operation.

```
K_index = 0
For (X: 0 ... Width_S -1)
  For (Y: 0 ... Height_S -1)
    For (Z: 0 ... Palette_S -1)
      S (X, Y, Z) = S (X, Y, Z)  XOR  K_Atom[K_index]
      K_index = K_index + 1
```

The substitution operation is used to satisfy as much as possible of the confusion effect in the data of the source image *S*. Choosing the XOR logical operation aims to implement a fast operation where using a mathematical formula would slow the encryption process. Also, the random values of the amino acid code are used to certainly add more difficulties for attackers to know the key and break the encrypted data.

2. Transposition Operation

For more protection to the encrypted data produced from the substitution operation and to fulfil Shanons' criteria, transposition operation is implemented to cause the diffusion effect on the contents of the source image *S*. This is conducted by performing swap operations between the bytes of *S*. The following programming code is implemented for the transposition operation.

```
K_index = 0
For (X: 0 ... Width_S -1)
  For (Y: 0 ... Height_S -1)
    For (Z: 0 ... Palette_S -1)
S (X, Y, Z) ↔ S (K_X[K_index], K_Y[K_index], K_Z[K_index])
K_index = K_index + 1
```

The nature of the 3D structure of the protein that has random *X, Y*, and *Z* coordinates helps to achieve a high level of distribution of the bytes over the whole content of the image *S*. Besides, the swapping operations cause more difficulty for the attackers to predict the key.

4.3. Construct Encrypted Image E

After completing the substitution and transposition operations, the resulting byte values are combined to produce an encrypted image *E*.

4.4. Decryption Stage

The same primary protein key *K* and the encrypted image *E* are used in this stage to recover the source image *S*. The perpetration stage is performed here in the same procedure as described earlier. Then the substitution and transposition operations are performed but in reverse order.

5. Implementation and Discussion

To evaluate the performance of the proposed encryption technique, a set of measurements is used in the experiments to test the proposed encryption technique, such as statistical and visual tests, information entropy, avalanche effect, key space, encryption execution time, and correlation analysis. In addition, a comparison is made between the proposed encryption technique with commonly used cryptosystems (such as DES, AES, and DNA-based methods) based on some criteria. These criteria are an encryption execution time, normalized mean absolute error (NMAE), peak signal-to-noise ratio (PSNR), Information Entropy, correlation, number of changing pixel rates (NPCR), the unified averaged changed intensity (UACI), and key space (Brute force attack) [23].

In the experiments, various images of different contents and sizes were selected and used to evaluate the performance of the proposed technique. The strengths and weaknesses of the proposed technique are determined by comparing the results of the proposed technique with the existing cryptosystems. Some of these images are shown in Fig. 3, and their results are listed afterward.

5.1. Statistical and Visual Tests

Attackers can use an image histogram to get information about the distribution of the color/byte intensity in the image. The efficiency of any image encryption technique depends on achieving a high degree of flatness in the color/byte histogram of the encrypted image. This factor plays an important role in achieving a high level of protection for encrypted images against statistical analysis attacks. This will add more difficulties for the attackers. Figure 4 shows the histograms of the source and the encrypted image of two selected images as examples, namely Bridge and the Monument images of Fig. 3.

Figure 4 indicates that the proposed image encryption technique succeeded in producing a high level of flatness in the encrypted image histogram compared to the source image histogram.

The peaks shown at the same locations in both, the source and the encrypted images may leak some information which presents a drawback of the algorithm.

However, the overall histogram presented no serious threat as was manifested by the results of the information entropy, avalanche effect, NMAE, and the PSNR tests.

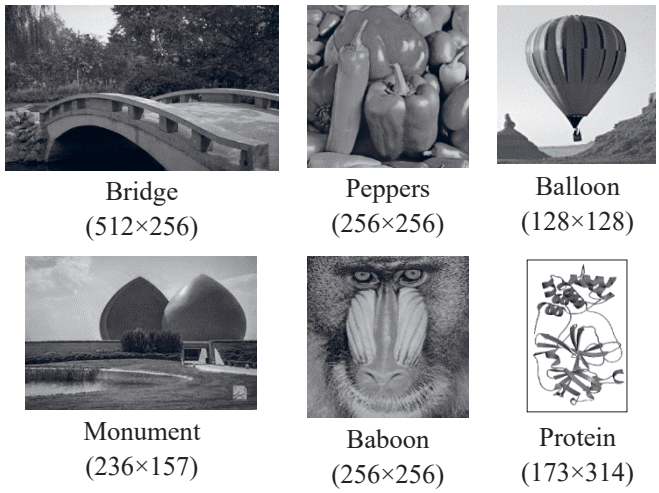


Figure 3. Source images used in the experiments.

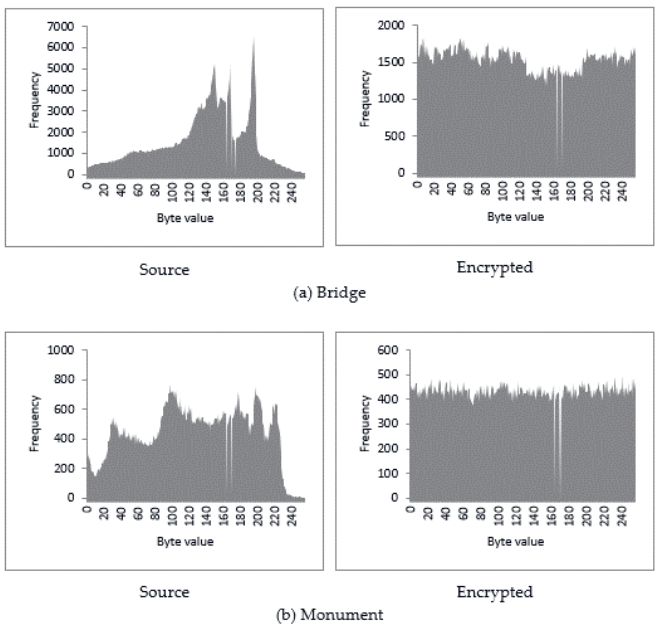


Figure 4. Histograms of the source and encrypted images, (a) Bridge (b) Monument.

Achieving a high level of diffusion and confusion effects in the encrypted image is the major objective of any proposed image encryption technique. The performance efficiency of the image encryption technique depends on the distortion rate generated in the encryption image, which will prevent any attacker from guessing the nature of

the source image. The encrypted images that resulted in the testing experiments proved that the proposed image encryption technique produces a high level of distortion in the encrypted images. Figure 5 shows the encrypted images generated using the proposed image encryption technique for the images in Fig. 3.

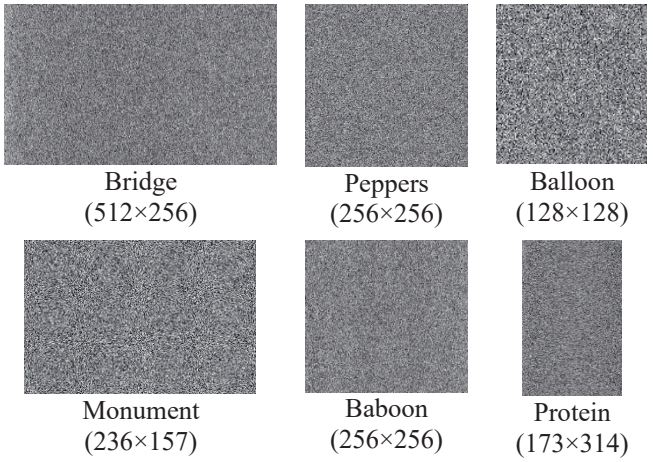


Figure 5. The encrypted images generated from the source images in Fig. 3.

5.2. Information Entropy

The property of randomness of any image is called Information entropy. Entropy is the average (expected) amount of information from the data. When the information entropy of an image is high, it becomes difficult to predict the content of that image. Equation (5) is used to calculate the information entropy [24], [25].

$$Entropy = -\sum_{i=1}^{n} P_i . \log_2(P_i) \quad (5)$$

Where n is the number of different data values and P_i is the probability of the data value occurring.

Table 4 lists and compares the information entropy of the source and encrypted images with that well-known encryption techniques; DES and AES, and DNA-based encryption algorithm [25].

From Table 4, it is obvious that the proposed encryption method produces an acceptable level of randomness in the encrypted image compared with the other well-known encryption techniques.

The average entropy value of the three encryption techniques is about 8, which indicates that the proposed image encryption technique is comparable to the other techniques and indicates that it is difficult to implement a successful attack.

From Table 4, it is obvious that the proposed encryption method produces an acceptable level of randomness in the encrypted image compared with the other well-known encryption techniques.

The average entropy value of the three encryption techniques is about 8, which indicates that the proposed image encryption technique is comparable to the other techniques and indicates that it is difficult to implement a successful attack.

Image	Entropy				
	Source	Proposed	DES	AES	DNA
Bridge	7.5422	7.9945	7.9995	7.9995	7.9769
Peppers	7.6772	7.9967	7.9990	7.9991	7.9636
Balloon	7.1668	7.9815	7.9967	7.9963	7.9653
Monument	7.7973	7.9981	7.9982	7.9982	7.9950
Baboon	7.7788	7.9978	7.9990	7.9990	7.9941
Protein	2.8616	7.8934	7.9988	7.9991	6.4184
Average Entropy value	6.8040	7.9770	7.9990	7.9985	7.7189

Table 4. Entropy values of the source and encrypted images.

5.3. Avalanche Effect

The avalanche effect is a numerical and a visual measure used to test the sensitivity of the proposed encryption technique to any small changes in its parameters. The main goal is to develop an encryption technique with high sensitivity characteristics, i.e., any small change in the key or the source data causes a considerable change in the encrypted data.

The test is conducted for different number of bits variations in the encryption key, and the effects of these changes are calculated in the generated encrypted image and the recovered source image.

5.3.1. Avalanche effect on the encrypted image

During the experiments, equation (6) [26] was used to calculate the avalanche effect (AE), the percentage of the number of bits in the encrypted image that changes when a few bits in the used encryption key are changed.

$$Avalanche\ Effect = \frac{Number\ of\ changed\ bits\ in\ key\ used}{Total\ number\ of\ bits\ in\ encrypted\ image} \times 100 \quad (6)$$

The avalanche test was performed by changing different number of bits in the encryption key between 1 bit to 7 bits. Table 5 lists the avalanche test values recorded in the experiments of the example images in Fig 3.

5.3.2. Avalanche effect on the recovered source image

Another use of the avalanche effect is in the source image recovery, where any changes in the encryption key bits will affect the recovered source image. An effective image encryption technique is that when some bits in the encryption key are changed and then use the key to decrypt the encrypted image, it will create an image completely different from the source image. Figure 6 shows the recovered

images after changing a number of bits in the encryption key for some images in Fig. 5. The recovered images in Fig. 6 are completely different from the source images of Fig. 3. This means that the key used by the proposed image encryption technique is very sensitive to the avalanche effect.

Image	Number of changed bits in the encryption key (%)		
	1 bit	4 bits	7 bits
Bridge	50.015	50.017	50.029
Peppers	50.008	50.009	50.010
Balloon	50.043	50.044	50.045
Monument	50.053	50.054	50.081
Baboon	49.989	49.991	50.014
Protein	50.043	50.044	50.051

Table 5. Avalanche test results during experiments.

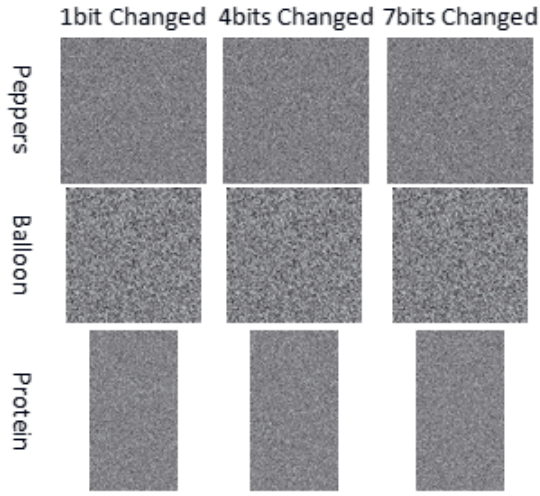


Figure 6. The effect of changing bits in the encryption key on the recovered images.

5.4. Key Size and its Complexity

The success and failure of a brute force attack mainly depend on the length and randomness of the encryption key. Using a larger key length means the attackers need more time to breach the security. Moreover, a high degree of randomness in the key increases the difficulty of guessing the key. The length of the key used is normally measured in bits.

For the proposed image encryption technique, its key-length depends on the number of bits required for the protein structure representation as well as the number of bytes (n) in the source image, and can be clarified as follows; 8 bits to represent the random value between (0...255) for each one of the 24 amino acids and 8 bits to represent the random value between (0...255) for each one of the 38 atom types. Then,

if an image with width (W), height (H), and color palette (P) is to be encrypted, the total number of bytes in the image shall be n = (W×H×P).

Therefore, the key length in bits (K_{bit}) for this image encryption technique is calculated by equation (7).

$$K_{bit} = \left((8+8) + \left(\frac{\log W + \log H + \log P}{\log 2} \right) \right) \times n \quad (7)$$

In the proposed image encryption technique, the user can select any digital protein file of any size to be used for the encryption key. The primary key (protein file) used in this technique is usually large, as illustrated by the following example.
The encryption key in the proposed algorithm results in a huge key space compared with other encryption algorithms, such as 3DES, AES, Twofish, Blowfish, and RC4, as listed in Table 6 [27], [28], which indicates strong security.

Algorithm	Encryption key	
	Key Size (bits)	Key space
3DES	168	2^{168}
AES	256	2^{256}
Twofish	256	2^{256}
Blowfish	448	2^{448}
RC4	2048	2^{2048}
Proposed Technique	K_{bit}	$> 2^{K_{bit}}$

Table 6. Key size and key space comparison.

Where: K_{bit} is defined in equation (7)

Example: Consider the (236×157 pixels) monument color image shown in Fig 3. The width, W = 236, and the height, H = 157, P = 3, hence n = (W×H×P) then, the encryption key length is:

$$K_{bit} = \left((8+8) + \left(\frac{\log 236 + \log 157 + \log 3}{\log 2} \right) \right) \times 111156$$

$\therefore K_{bit} > 6924493.102$ bits

5.5. Encryption Execution Time

The speed of the encryption process is one of the important metrics for evaluating the performance of an encryption method. The same images have been encrypted using the proposed technique and other known encryption methods (DES, AES, DNA), the recorded results are listed in Table 7.

It can be seen from Table 7 that the encryption process time, including the time required for the preparation stage, of the proposed technique is relatively close to the

known encryption methods. This means that the proposed encryption technique can be used effectively to protect the images.

Image	Encryption Time ET (sec)			
	Proposed	***DES***	***AES***	***DNA***
Bridge	0.762	0.629	0.592	0.632
Peppers	0.402	0.328	0.307	0.301
Balloon	0.117	0.157	0.140	0.150
Monument	0.245	0.208	0.134	0.172
Baboon	0.400	0.347	0.258	0.306
Protein	0.365	0.379	0.361	0.375
Average ET value	0.382	0.341	0.299	0.323

Table 7. Encryption process time for encryption methods.

5.6. NMAE and PSNR Metrics

Two numerically determined metrics can be determined to reflect the amount of distortion produced in the encrypted images to thwart the attackers, namely the normalized mean absolute error NMAE and peak signal to noise ratio PSNR. These metrics are calculated by Equations (8) and (9), respectively [29] as follows:

$$NMAE = \frac{\sum_{k=0}^{n-1}|S(k)-E(k)|}{n} \times 100 \tag{8}$$

$$PSNR_{db} = 10.\log_{10}\left(\frac{Max_S^2}{NMAE}\right) \tag{9}$$

Where S is the source image and E is the encrypted image, and Max_S is the maximum possible pixel value of S.

Tables 8 and 9 show the resulting NMAE and the PSNR values for the proposed technique and for DES, AES, and DNA encryption methods. The NMAE and PSNR values of the proposed technique are competitive values with well-known DES, AES, and DNA encryption methods. Moreover, the NMAE and PSNR values in Tables 8 and 9 are represented graphically in Fig. 7 and Fig. 8, respectively.

Image	NMAE (%)			
	Proposed	***DES***	***AES***	***DNA***
Bridge	58.57	57.60	57.69	58.15
Peppers	83.53	83.40	83.37	82.56
Balloon	66.33	66.19	65.99	65.72
Monument	65.47	65.19	65.11	65.45
Baboon	67.19	66.78	66.93	66.34
Protein	53.35	55.41	55.34	46.67
Average NMAE value	65.74	65.76	65.74	64.15

Table 8. NMAE values for encryption methods.

Image	PSNR (dB)			
	Proposed	***DES***	***AES***	***DNA***
Bridge	7.86	7.99	7.99	7.89
Peppers	6.77	6.78	6.79	6.85
Balloon	6.77	6.84	6.84	6.89
Monument	7.49	7.54	7.55	7.51
Baboon	7.53	7.58	7.56	7.63
Protein	5.62	5.30	5.32	5.46
Average PSNR value	7.01	7.01	7.01	7.04

Table 9. PSNR values for encryption methods.

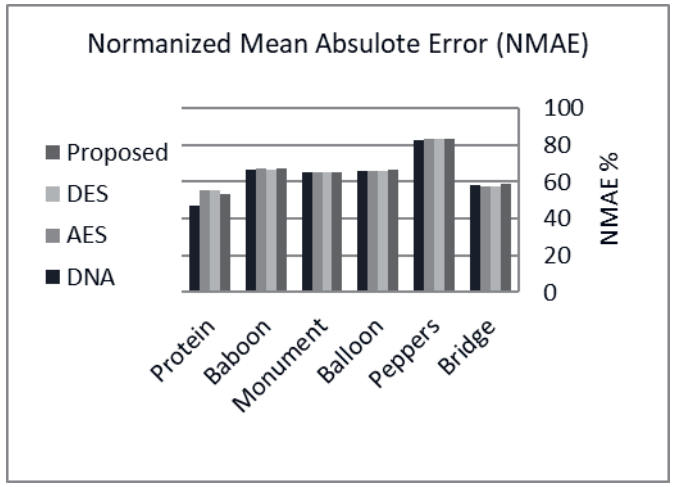


Figure 7. The graphical representation of NAME values of Table 8.

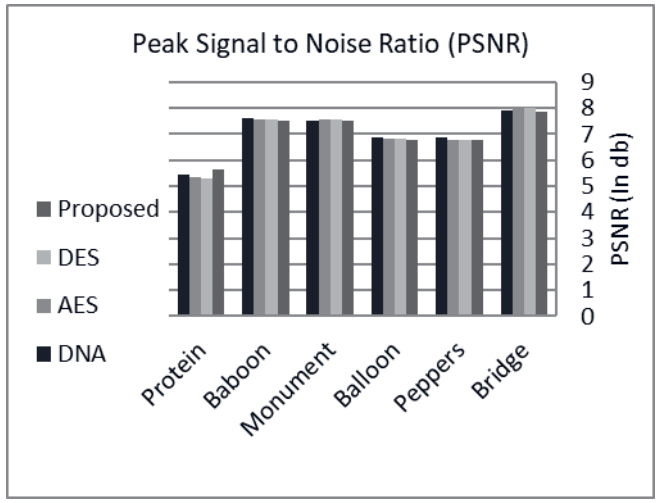


Figure 8. The graphical representation of PSNR values of Table 9.

5.7. Correlation Metric

One of the distortion effects that the encryption method aims to achieve in the encrypted image leads to minimizing the relationship between the values of the adjacent pixels in the encrypted image. Equation (10) is used to calculate the correlation between two adjacent pixels in the diagonal orientations.

$$C = \frac{N\sum_{j=1}^{N}(X_j \times Y_j) - \sum_{j=1}^{N} X_j \times \sum_{j=1}^{N} Y_j}{\sqrt{\left(N\sum_{j=1}^{N} X_j^2 - \left(\sum_{j=1}^{N} X_j\right)^2\right) \times \left(N\sum_{j=1}^{N} Y_j^2 - \left(\sum_{j=1}^{N} Y_j\right)^2\right)}} \quad (10)$$

Where X and Y are the values of the colors of two adjacent pixels in the encrypted image and N is the number of adjacency pixels selected in the image that are used to calculate the correlation.

During the experiments, 3000 pairs of adjacent pixels were randomly selected. Equation (10) is used to test the correlation between two adjacent pixels in the vertical, horizontal, diagonal, and anti-diagonal orientations. Tables 10-13 show the correlation values of the proposed technique and the known DES, AES, and DNA encryption methods in the vertical, horizontal, diagonal, and anti-diagonal orientations, respectively.

The recorded correlation values in Tables 10-13 indicate that the proposed techniques achieved a competitive average correlation value with other methods.

Image	**Correlation Values – Vertical Orientation**			
	Proposed	***DES***	***AES***	***DNA***
Bridge	0.0104	0.0031	0.0052	0.0108
Peppers	0.0344	0.0141	0.0194	0.0485
Balloon	0.0221	0.0571	0.0195	0.0278
Monument	0.0041	0.0649	0.0023	0.0476
Baboon	0.0535	0.0205	0.0145	0.0394
Protein	0.0123	0.0251	0.028	0.0134
Average correlation value	0.0228	0.0308	0.0148	0.0313

Table 10. Correlation values for encryption methods in vertical orientation.

Image	**Correlation Values – Horizontal Orientation**			
	Proposed	***DES***	***AES***	***DNA***
Bridge	0.0547	0.0317	0.0014	0.0645
Peppers	0.0286	0.0221	0.0281	0.0481
Balloon	0.0022	0.0531	0.0076	0.0201
Monument	0.0467	0.0349	0.0565	0.0709
Baboon	0.0816	0.0023	0.0223	0.0711
Protein	0.0114	0.0066	0.0223	0.0122
Average correlation value	0.0375	0.0251	0.023	0.0478

Table 11. Correlation values for encryption methods in horizontal orientation.

Image	Correlation Values – Diagonal Orientation			
	Proposed	***DES***	***AES***	***DNA***
Bridge	0.0093	0.0372	0.0056	0.0139
Peppers	0.0384	0.0166	0.0026	0.0305
Balloon	0.047	0.0083	0.0422	0.0768
Monument	0.0321	0.008	0.0418	0.0411
Baboon	0.0666	0.0033	0.0242	0.0089
Protein	0.0385	0.0235	0.0412	0.0102
Average correlation value	0.0387	0.0162	0.0263	0.0302

Table 12. Correlation values for encryption methods in diagonal orientation.

Image	Correlation Values – Anti-diagonal Orientation			
	Proposed	***DES***	***AES***	***DNA***
Bridge	0.0246	0.0181	0.0455	0.0167
Peppers	0.0109	0.0112	0.0243	0.0109
Balloon	0.014	0.0674	0.0287	0.0835
Monument	0.0348	0.0065	0.0019	0.0018
Baboon	0.0078	0.025	0.0507	0.0422
Protein	0.0229	0.0094	0.0048	0.0442
Average correlation value	0.0192	0.0229	0.026	0.0332

Table 13. Correlation values for encryption methods in anti-diagonal orientation.

5.8. NPCR and UACI Randomness Tests

The strength of image encryption algorithms for differential attacks can be assessed using NPCR and UACI metrics. The NPCR and UACI are calculated using equations (12) and (13), respectively.

$$D(i,j) = \begin{cases} 1, C_1(i,j) \neq C_2(i,j) \\ 0, C_1(i,j) = C_2(i,j) \end{cases} \tag{11}$$

$$NPCR = \frac{\sum_{i=1}^{Width} \sum_{j=1}^{Height} D(i,j)}{Width \times Height} \times 100\% \tag{12}$$

$$UACI = \frac{\sum_{i=1}^{Width} \sum_{j=1}^{Height} |C_1(i,j) - C_2(i,j)|}{Width \times Height \times 255} \times 100\% \tag{13}$$

Where C_1 and C_2 represent the encrypted images before and after one pixel change in a source image, $Width \times Height$ represents the image size. The calculated values of NPCR and UACI between the two encrypted images C_1 and C_2 are listed in Table 14 and 15, respectively.

It is clear from Tables 14 and 15 that the proposed image encryption algorithm succeeded in creating two different encrypted images when only one pixel in the source image changed, which means that the algorithm has a high sensitivity to any small changes in the source image.

Image	NPCR (%)			
	Proposed	***DES***	***AES***	***DNA***
Bridge	99.33	99.60	99.61	99.21
Peppers	99.40	99.58	99.61	99.30
Balloon	99.00	99.60	99.61	99.01
Monument	99.53	99.60	99.60	99.44
Baboon	99.50	99.64	99.62	99.50
Protein	99.00	99.62	99.59	99.10
Average correlation value	99.29	99.61	99.61	99.26

Table 14. NPCR values between the two encrypted images C_1 and C_2.

Image	UACI (%)			
	Proposed	***DES***	***AES***	***DNA***
Bridge	33.03	33.44	33.54	33.12
Peppers	33.26	33.39	33.52	33.01
Balloon	33.17	33.66	33.50	33.24
Monument	33.33	33.40	33.48	33.07
Baboon	32.92	33.48	33.39	33.56
Protein	32.90	33.43	33.37	33.27
Average correlation value	33.1	33.47	33.47	33.21

Table 15. NPCR values between the two encrypted images C_1 and C_2.

6. Conclusions

A new symmetric image encryption scheme based on 3D protein structure is developed, producing a considerably strong encryption key with huge key - spaces. Utilizing the randomness of protein sequences along with their chaotic and very complex 3D structures led to the construction of random data tables from which extremely strong encryption keys were generated. Various security tests that included statistical and visual tests, avalanche tests, entropy, normalized mean absolute error, peak signal to noise ratio, and values were satisfactory compared to other cryptosystems, such as DES, AES, and DNA methods. The proposed cryptographic scheme can also be linked to any other cryptosystem to provide a more efficient hybrid security technique suitable for various applications.

There is no limit to the type of images used as the proposed algorithm can encrypt any images (e.g., RGB, RGBA, etc.). Different images of different sizes were used in the experiment to evaluate the proposed algorithm. The proposed algorithm treats the source image as a single block. This makes it difficult to encrypt very large images. Therefore, in future work, the source image will be divided into a set of small segments and encrypt each one separately.

Acknowledgments

- Mohammed Abbas Fadhil Al-Husainy is grateful to the Irbid National University, Irbid, Jordan for the financial support granted to cover the publication fee of this research article.
- Hamza A. A. Al-Sewadi is grateful to the Iraq University College, Basra, Iraq for the financial support granted to cover the publication fee of this research article.
- Ahmed M. Sayed is grateful to the Nahda University, Beni-Suef, Egypt for the financial support granted to cover the publication fee of this research article.

References

[1] Saračević, M. H., Adamović, S. Z., Miškovic, V. A., Elhoseny, M., Maček, N. D., Selim, M. M., & Shankar, K. Data encryption for Internet of Things applications based on catalan objects and two combinatorial structures. IEEE Transactions on Reliability, 2020, 70(2), 819-830.

[2] Ahani, S., & Ghaemmaghami, S. (2015). Color image steganography method based on sparse representation. IET Image Processing, 9(6), 496-505.

[3] Zong, T., Xiang, Y., Guo, S., & Rong, Y. (2016). Rank-based image watermarking method with high embedding capacity and robustness. IEEE Access, 4, 1689-1699.

[4] Afek, Y., Alon, N., Barad, O., Hornstein, E., Barkai, N., & Bar-Joseph, Z. (2011). A biological solution to a fundamental distributed computing problem. science, 331(6014), 183-185.

[5] Kroeker, K. L. (2011). Biology-inspired networking. Communications of the ACM, 54(6), 11-13.

[6] Tero, A., Takagi, S., Saigusa, T., Ito, K., Bebber, D. P., Fricker, M. D., & Nakagaki, T. (2010). Rules for biologically inspired adaptive network design. Science, 327(5964), 439-442.

[7] Xiao, G., Lu, M., Qin, L., & Lai, X. (2006). New field of cryptography: DNA cryptography. Chinese Science Bulletin, 51(12), 1413-1420.

[8] Debora S. Marks, Lucy J. Colwell, Robert Sheridan, Thomas A. Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. (2011). Protein 3D Structure Computed from Evolutionary Sequence Variation. Plos One.

[9] Pearson W. R. (2013). An introduction to sequence similarity ("homology") searching. Current protocols in bioinformatics, Chapter 3, Unit3.1.

[10] Whitford, D. (2013). Proteins: structure and function. John Wiley & Sons.

[11] Agrawal, A., Sarkar, C., Dwivedi, S. K., Dhasmana, N., & Jalan, S. (2014). Quantifying randomness in protein–protein interaction networks of different species: a random matrix approach. Physica A: Statistical Mechanics and its Applications, 404, 359-367.

[12] Hassan, R., Pepic, S., Saracevic, M., Ahmad, K., Tasic, M. A Novel Approach to Data Encryption Based on Matrix Computations. Comput. Mater. Contin, 2020, 66, 1139-1153.

[13] Al-Husainy M. A., Al-Sewadi H. A, and Masadeh S. R. (2022) Using DNA Tape as a Key for Image Encryption. International Journal of Electronic Security and Digital Forensics, Vol 14, No. 4.

[14] Chen J. (2003). A DNA-based, biomolecular cryptography design. Proceeding of the IEEE International Symposium on Circuits and Systems (ISCAS), Vol. 3, PP 822–825

[15] Roy B., Singha P. (2011). An improved symmetric key cryptography with DNA based strong cipher". Devices and Communications (ICDeCom), IEEE, International Conference on 24-25 Feb. 2011.

[16] Tanaka K., Okamoto A., and Saito I. (2005). Public-key system using DNA as a one-way function for key distribution. Biosystems, Vol. 81, N0 1, PP 25–29. https://doi.org/10.1016/j.biosystems.2005.01.004

[17] Cui G.Z., Qin L. M., and Wang Y. F. (2008). An Encryption Scheme using DNA Technology. Computer Engineering and Applications, PP 37-42.

[18] Zhang Y., Fu B., and Zhang X. (2012). DNA cryptography based on DNA Fragment assembly. IEEE International Conference on Information Science and Digital Content Technology (ICIDT), Vol. 1, PP 179-182.

[19] Khana J. S., Ahmadb J., Ahmeda S. S., Siddiqac H. A., Abbasid S. F., and Kayhana S. K. (2019). DNA key based visual chaotic image Encryption. Journal of Intelligent & Fuzzy Systems, IOS press, DOI:10.3233/JIFS-18277

[20] Devi R. S., Aravind A.R. N., Vishal J. C, D. Amritha D., Thenmozhi K., Rayappan J. B. B., Rengarajan A., Padmapriya P. (2020). Image encryption through RNA approach assisted with neural key sequences. Multimedia Tools and Applications, Springer Science+Business Media, LLC, part of Springer Nature. https://doi.org/10.1007/s11042-019-08562-5

[21] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., ... & Bourne, P. E. (2000). The protein data bank. Nucleic acids research, 28(1), 235-242.

[22] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., ... & Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. Nature, 596(7873), 583-589.

[23] Atiewi, S., Al-Rahayfeh, A., Almiani, M., Yussof, S., Alfandi, O., Abugabah, A., & Jararweh, Y. (2020). Scalable and secure big data IoT system based on multifactor authentication and lightweight cryptography. IEEE Access, 8, 113498-113511.

[24] H. Zhang, J. E. Fritts, and S. A. Goldman (2003). Entropy-based objective evaluation method for image segmentation. in Storage and Retrieval Methods and Applications for Multimedia 2004, vol. 5307, pp. 38–49.

[25] Zhang W., Wang S., Han W., Yu H., and Zhu Z. (2020). An Image Encryption Algorithm Based on Random Hamiltonian Path," Entropy, vol. 22, no. 1, p. 73.

[26] Alabdullah B., Beloff N., and White M. (2021). E-ART: A New Encryption Algorithm Based on the Reflection of Binary Search Tree." Cryptography, Vol. 5, No. 4. https://doi.org/10.3390/cryptography5010004.

[27] Tagashira M, and Nakagawa T. (2020). "Biometric Authentication Based on Auscultated Heart Sounds in Healthcare," IAENG International Journal of Computer Science, vol. 47, no. 3.

[28] Kushwaha P. K., Singh M. P., and Kumar P. (2014). A survey on lightweight block ciphers. International Journal of Computer Applications, vol. 96, no. 17.

[29] Al-Husainy M. A. F. and Al-Sewadi H. A. A. (2018). Full Capacity Image Steganography Using Seven-Segment Display Pattern as Secret Key. Journal of Computer Science, Vol. 14, No. 6, pp 753-763. https://doi.org/10.3844/jcssp.2018.753.763.