NATURENGS
MTU Journal of Engineering and Natural Sciences
https://dergipark.org.tr/tr/pub/naturengs
DOI: 10.46572/naturengs.1009013

Random Number Generator Based on Discrete Cosine Transform Based Lossy Picture Compression

Selman YAKUT[1]*

[1]Department of Computer Engineering, Faculty of Engineering and Natural Sciences, Malatya Turgut Özal University, Malatya, Turkey.

(**Received:** 13.10.2021; **Accepted:** 30.11.2021)

ABSTRACT: The widespread digitalization make the security of digital systems more important. Various cryptographic systems are used to ensure the security of the systems. An important part of the system is random numbers. In the article, a random number generator based on the discrete cosine transform, which is the basis of image compression algorithms, is proposed. In this generator, the difference between the original image and the compressed image produced using the discrete cosine transform is used. The original picture is transferred to the frequency domain by using the discrete cosine transform. Then, it is converted back to the space domain by using the inverse discrete cosine transform. These transformations cause some losses as certain coefficients are taken into account. Raw random numbers were generated using the differences between the original image and the compressed image. Then, the possible weaknesses in the generated random numbers were removed by passing these raw data through the hash function. The SHA-512 algorithm was used as the hash function. An important advantage of the developed system is that it can be easily implemented. The safety of generated random numbers is demonstrated by the successful result of NIST and correlation tests. In addition, the security parameters of the proposed method are shown in the discussion section.

Keywords: Discrete cosine transform, Image compression, Random numbers, Hash functions.

1. INTRODUCTION

Today, digital devices are widely used day by day. The use of the devices such as smartphones, tablets, and computers is indispensable in many areas [1]. Cyberattacks, on the other hand, abuse the weak sides of these digital devices [2]. Many cryptographic systems and protocols are used to avoid this weakness [3-5]. These systems ensure the security of both data sources and data. The basis of these systems is usually based on secret and secure parameters such as key and seed value [3]. For the security of the systems, it is important to produce these values from reliable sources [4]. Random number generators produce secure random numbers, which is the most important parameter that affects the security of these systems and data [3].

Random numbers are the most important part of many cryptographic applications [6]. Private-public key pair, secret key, seed value are among them. Since this value has critical importance on the whole system, it is important to produce them. In addition, there are security parameters that random numbers must provide to be used in cryptographic systems. These parameters are

*Corresponding Author: selman.yakut@ozal.edu.tr
ORCID number of authors: [1] 0000-0002-0649-1993

expressed as R1-R4 and guarantee the safety of the numbers used. R1-R4 parameters are briefly described in Table 1.

Table 1. The security requirements for secure random number generators

R1:	Random numbers should not contain any statistical weaknesses
R2:	Knowing the subsequences of random numbers should not allow the calculation or prediction of the preceding and succeeding random numbers
R3:	It should not be possible to compute the previous random numbers with the possibility of prediction when the internal state value is known, or even if the internal state value is unknown
R4:	It should not be possible to compute the next random numbers with the possibility of prediction when the internal state value is known, or even if the internal state value is unknown

Various methods and sources are used for random number generation. However, random number generators are divided into three sub-classes: true random number generators (TRNG), pseudo-random number generators (SRSU), and hybrid random number generators (HRNG). TRNG generators use physical and non-repeatable sources such as electronic noise and radioactive decay [7]. PRNG are generators based on some calculations which use a specific algorithm and seed value [8-10]. HRNG, on the other hand, is an approach based on the use of these two approaches together [11].

Many post-processing algorithms are used to eliminate possible weaknesses in random numbers generators [6]. Post-processing algorithms are widely used to make the generator more secure [12-16]. There are many post-processing algorithms in the literature [13-15]. Cryptographic hash algorithms are widely used as post-processing algorithms in random number generation [15]. The important reason for this is that these algorithms are resistant to collisions and are one-way functions. In addition, these algorithms produce data with a uniform distribution and strong statistical properties [16].

Data generated from digital data sources are subjected to compression processes to reduce the data load in transmission and storage processes. Compression is divided into two classes such as lossy and lossless compression [17-18]. Lossless compression is that the original image is compressed without any loss of the image. On the other hand, Lossy compression causes some loss on the original image and the original image does not reproduce. The most widely used method of lossy compression is to compress the original image by converting it to the frequency domain. Discrete cosine transform, which is used in many algorithms such as JPEG, is an important transform method.

Discrete cosine transform is a widely used method in data compression [19-22]. In the method, data is first converted into the frequency domain. Then, in the frequency domain, some coefficients that take into account the human visual sense are kept, while some coefficients are not taken into account [23]. Thus, significant compression of the data is done by using a series of processes that are not noticed or barely noticed by the human eye [24]. When converted to this compressed image, the original image does not reproduce because of some loss [19-22].

In the study, a random number generator based on the discrete cosine transform, which is the basis of image compression algorithms, is proposed. Random numbers are generated by using the difference values between the original image and the compressed image using the discrete cosine transform. These numbers are passed through cryptographic hash algorithms, ensuring that they have a uniform distribution and become safe. It can be used as a random number generator in many digital applications with the proposed method. In addition, it is shown that generated random numbers are safe by statistical tests.

The rest of this paper is organized as follows: In the second part, the structure of the proposed method, the entropy source used, and the operations performed were examined. In the next section, the safety evaluation of the numbers produced and the method used were made. The last section contains the results.

2. MATERIAL AND METHODS

In the study, secure random numbers, which are an important part of many applications, are generated. The proposed approach is given in Figure 1. Firstly, the original picture is transformed into the frequency domain by using the DCT. Then, the data in the frequency domain is transformed back to the space domain by going through IDCT and the compressed image obtained. The compressed image is subtracted from the original image and the difference matrix is obtained. Then, raw random numbers are produced by converting the data in each row and column in this matrix to binary number format. Then, these raw data are passed through the cryptographic hash algorithm and possible correlations and weaknesses are removed.

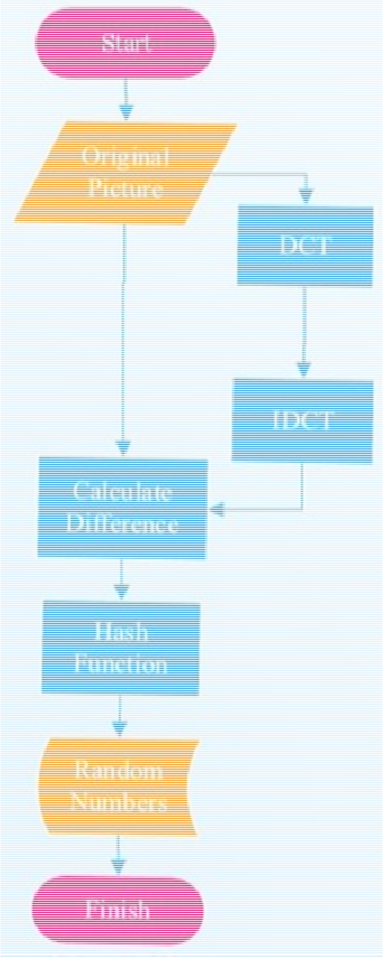


Figure 1. The general structure of the proposed method

DCT is a Fourier-based method that converts data from the space domain to the frequency domain. In the method, original data is represented with fewer coefficients without any significant difference in terms of human visual psychology. In the method, compressed data is created by giving some losses from the original data. First, the original data is transferred to the frequency domain by DCT transform. The data transferred to this domain is expressed by the coefficient matrix. Some values of this coefficient matrix are preserved by considering human visual psychology and some values are neglected. Then, the data in the frequency domain is transferred to the space domain by using the IDCT transform, and compressed data is obtained. The difference between the original and the compressed image is taken as an entropy source.

A random number generator is designed by using the difference between the original data and the compressed data as an entropy source. Firstly, the original image pixel value is subtracted from the compressed image pixel value and the difference matrix is calculated. Then these difference matrix values are taken as positive and converted to binary. A binary bit sequence is converted to binary and added in a row. Finally, these binary bit string values are given as input to the hash algorithm and random numbers are generated.

Transferring images from the space domain to the frequency domain is widely used in steganography techniques. Many transformations are used for this transfer process. DCT is one of the commonly used transformations based on mathematical operations. The formulas used for DCT and IDCT transformations are given in Eq. (1), Eq. (3) respectively. First of all, transformation matrices were produced using the equations. Using these matrices, transformations are made from the space domain to the frequency domain.

$$C(u,v) = \mathrm{a(u)a(v)} \sum_{x=0}^{N-1} \sum_{y=1}^{N_1} f(x,y) \left(a_n \cos\frac{(2x+1)\pi x}{2N} + b_n \sin\frac{(2y+1)\pi x}{2N} \right) \tag{1}$$

Here the values of *a(u)* and *a(v)* are prepared Eq. (2)

$$a(k) = \begin{cases} \sqrt{\frac{1}{N}} & k = 0 \; ise \\ \sqrt{\frac{2}{N}} & k \neq 0 \; ise \end{cases} \tag{2}$$

The inverse discrete cosine transform is performed as shown in Eq. (3). IDCT transfers input images or data from the frequency domain to the space domain. With this inverse transformation, the data is compressed. Compressing is done without significant visual differences between the compressed data and the original data.

$$C(u,v) = \mathrm{a(u)a(v)} \sum_{x=0}^{N-1} \sum_{y=1}^{N_1} f(x,y) \left(a_n \cos\frac{(2x+1)\pi x}{2N} + b_n \sin\frac{(2y+1)\pi x}{2N} \right) \tag{3}$$

Potential weaknesses in these numbers are removed by using hash functions in the designed generator. The SHA-512 function, which is used in many applications and has a secure structure, is used as a cryptographic hash function [25]. These functions extract a summary of a given message, which can be expressed as a fingerprint. During this process, both data are compressed and passed through some logical and algebraic operations. With the help of these processes, possible weaknesses in these data are removed and safe summary values are produced. The general structure of these functions is given in Figure 2.

By using the structure of hash functions as in Figure 3, random numbers larger than the hash value are generated. The hash value taken at each step is used to generate random numbers and to calculate new data.

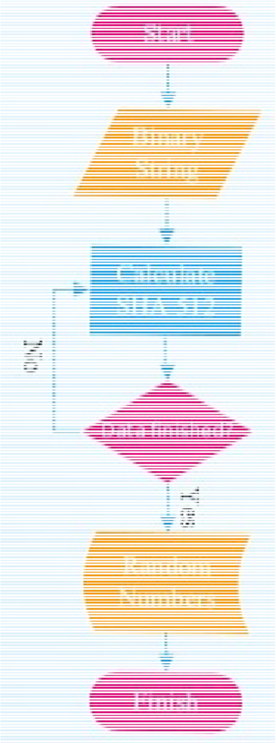


Figure 2. The post-processing structure of the proposed method

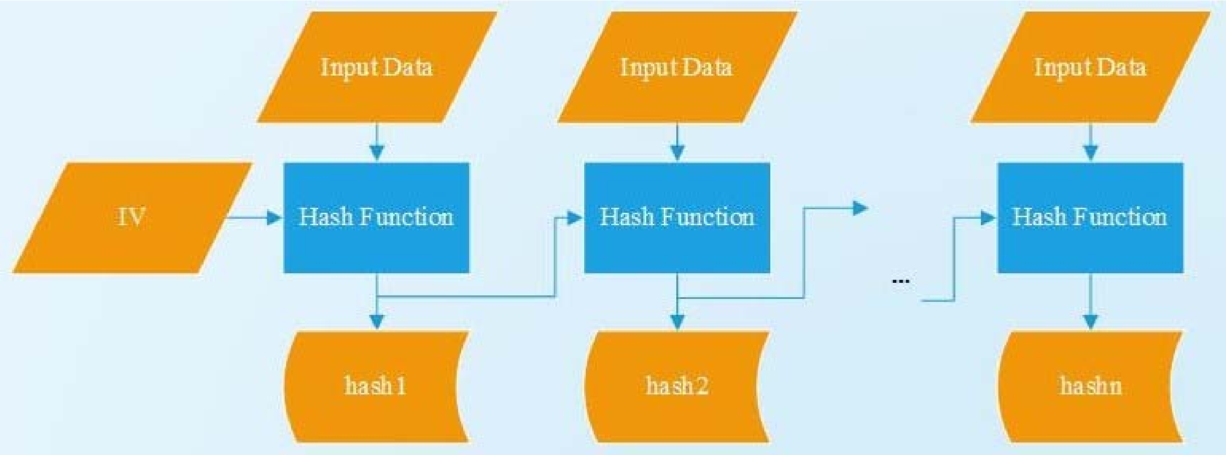


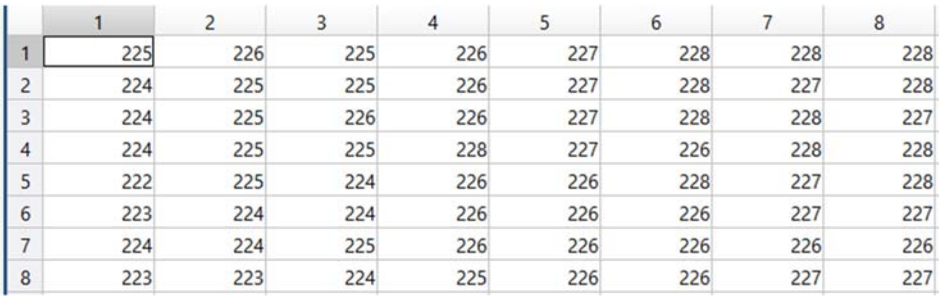
Figure 3. Cascade structure using cryptographic hash functions of the proposed method

3. RESULTS AND DISCUSSION

The security of applications used in many areas depends on the security of random numbers. Random numbers are an important part of the applications. Therefore, various security parameters must be met for the numbers used in these fields to be considered safe. The first of these parameters is that the source and method of the random numbers should be safe. Secondly, the numbers do not contain any statistical weakness. Another important parameter is that these numbers should meet the security requirements for cryptographic applications. In addition, the production cost of these numbers is another important parameter.

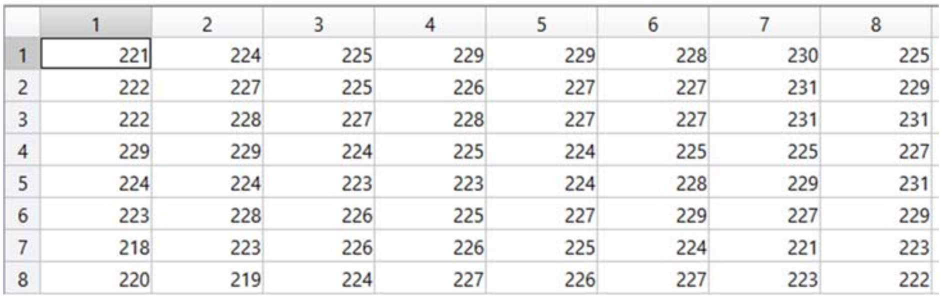
The proposed approach is based on the DCT method, which is widely used in lossy compression operations. Compressed data obtained with this approach is created with some loss from the original data. Here, the amount of loss varies according to the compression ratio in the original image. However, since some coefficients of the data transferred to the frequency space are not taken into account, the original picture is not produced from the compressed picture. Thus, the difference values between the compressed data and the original data can be used as an entropy source. In this study, raw random numbers are generated by using the entropy source. Here, is original image value, compressed image value, and difference matrix value is given in Table 2, Table 3, and Table 4 respectively.

Table 2. Original image value



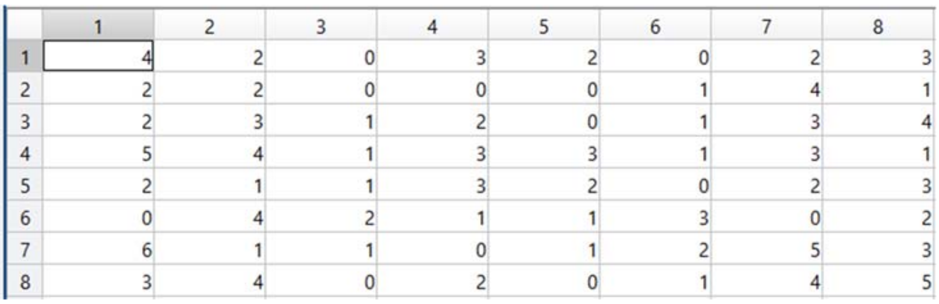
	1	2	3	4	5	6	7	8
1	225	226	225	226	227	228	228	228
2	224	225	225	226	227	228	227	228
3	224	225	226	226	227	228	228	227
4	224	225	225	228	227	226	228	228
5	222	225	224	226	226	228	227	228
6	223	224	224	226	226	226	227	227
7	224	224	225	226	226	226	226	226
8	223	223	224	225	226	226	227	227

Table 3. Compressed image value



	1	2	3	4	5	6	7	8
1	221	224	225	229	229	228	230	225
2	222	227	225	226	227	227	231	229
3	222	228	227	228	227	227	231	231
4	229	229	224	225	224	225	225	227
5	224	224	223	223	224	228	229	231
6	223	228	226	225	227	229	227	229
7	218	223	226	226	225	224	221	223
8	220	219	224	227	226	227	223	222

Table 4. Difference matrix value



	1	2	3	4	5	6	7	8
1	4	2	0	3	2	0	2	3
2	2	2	0	0	0	1	4	1
3	2	3	1	2	0	1	3	4
4	5	4	1	3	3	1	3	1
5	2	1	1	3	2	0	2	3
6	0	4	2	1	1	3	0	2
7	6	1	1	0	1	2	5	3
8	3	4	0	2	0	1	4	5

In the proposed approach, cryptographic hash functions are used to remove any possible statistical weaknesses and biases in the generator. The use of these functions guarantees the security of the generated numbers. Primarily, these functions meet the requirements for cryptographic systems. These requirements, expressed as R1-R4, are parameters such as the numbers produced do not contain any statistical weakness, do not reproduce, and do not be reversed. These requirements are briefly explained in Table 1. Hash functions meet these requirements because they are both unidirectional and conflict-proof.

Random number generators should not contain any statistical weaknesses because of security reasons. The weaknesses make possible various attacks on these numbers. So the numbers used in critical applications do not contain any statistical weaknesses. Various tests are used to statistically examine the random number generators. These tests have similar structures generally. The 800-22 test suite, proposed by NIST, and the autocorrelation method are widely used. The 800-22 test suite contains 15 different tests [26]. This test suite statistically analyzes the random numbers. The safety of the numbers is carried out using these tests. The successful test results are given in Table 5 show the safety of the proposed method.

Table 5. Statistical test results of the proposed method

Statistical tests	*P*-value	Results
Frequency Test	0.3751	Successful
Block-frequency Test	0.4972	Successful
Runs Test	0.5137	Successful
Test for the Longest Run of Ones in a Block Test	0.3254	Successful
Binary Matrix Rank Test	0.6176	Successful
Discrete Fourier Transform Test	0.1865	Successful
Non-overlapping Template Matching Test	0.3268	Successful
Overlapping Template Matching Test	0.6249	Successful
Maurer's Universal Statistical Test	0.7630	Successful
Linear Complexity Test	0.4534	Successful
Serial Test1	0.3017	Successful
Serial Test2	0.4782	Successful
Approximate Entropy Test	0.2164	Successful
Cumulative Sums Test	0.6595	Successful

The autocorrelation test is used to decide whether the produced random numbers are self-contained. The mathematical expression of the test is given in Eq. (4), where $\oplus$ is the *XOR* operation; *n* is the dimension, and *d* is an integer value in the range of $0 \le d \le n/2$. Eq. (5) shows the relationship between zero and one number. The results of the equation being in the range of |*X*_5¬|<1.6449 show that the proposed generator is successful [4]. Table 6 shows the successful results obtained using the method.

$$A(d) = \sum_{i=0}^{n-d-1} (b_n \oplus b_{n+d}) \tag{4}$$

$$X_5 = \frac{2[\mathrm{A(d)} - \frac{\mathrm{n-d}}{2}}{\sqrt{\mathrm{n-d}}} \tag{5}$$

Table 6. Autocorrelation test results

Test	D value	X5 value	Results
Autocorrelation	8	-0.4234	Successful
	10	0.5029	Successful
	13	-0.5341	Successful
	16	1.1878	Successful
	20	-0.7347	Successful
	25	-0.8628	Successful
	100	0.9652	Successful
	500	-0.9443	Successful
	1000	-1.2024	Successful

4. CONCLUSIONS

Random numbers form an important part of many algorithms. These numbers can be generated from different sources such as atmospheric phenomena and electrical noise. In the study, random numbers are generated by using lossy compression operations based on the discrete cosine transform. In the transformation, any data in the space domain is transferred to the frequency domain. In the coefficient matrix produced by this transfer process, some important coefficients are preserved and the unimportant ones are neglected, taking into account human visual psychology. These operations cause differences between the original image and the compressed image. In the study, random numbers are generated by using these differences as an entropy source. The cryptographic hash function is used to remove any possible weaknesses in these generated numbers. This function is unidirectional and conflict-proof, which makes the numbers safe. Thus, the generated numbers meet the security requirements for many areas, especially cryptographic applications. In addition, the NIST and autocorrelation tests are used to show that the produced numbers do not contain any statistical weakness. Thus, the generated random numbers can be used in many fields.

REFERENCES

[1] Flores-Vergara, A., Garcia-Guerrero, E. E., Inzunza-González, E., López-Bonilla, O. R., Rodríguez-Orozco, E., Cardenas-Valdez, J. R., Tlelo-Cuautle, E. (2019). Implementing a chaotic cryptosystem in a 64-bit embedded system by using multiple-precision arithmetic, *Nonlinear Dynamics*, 96(1): 497-516.

[2] Yu, J.-Y., Lee, E., Oh, S.-R., Seo, Y.-D., Kim, Y.-G. (2020). A Survey on Security Requirements for WSNs: Focusing on the Characteristics Related to Security, *IEEE Access*, 8: 45304–45324.

[3] Koç, Ç. (2009). *Cryptographic Engineering*. Springer, New York.

[4] Menezes, A.J., van Oorschot, P.C., Vanstone, S.A. (1996). *Handbook of Applied Cryptography, 1st edn.*, CRC Press, Boca Raton.

[5] Paar, C., Pelzl, J. (2009). *Understanding cryptography: a textbook for students and practitioners*, Springer, Bochum.

[6] Yakut, S. (2019). Design and Analysis of Real Random Number Generators, Ph.D. Thesis, Fırat University Institute of Science and Technology, Elazığ.

[7] Bakiri, M., Guyeux, C., Couchot, J. F., Oudjida, A. K. (2018). Survey on hardware implementation of random number generators on FPGA: Theory and experimental analyses, *Computer Science Review*, 27: 135-153.

[8] Aljohani, M.; Ahmad, I.; Basheri, M.; Alassafi, M.O. (2019). Performance Analysis of Cryptographic Pseudorandom Number Generators, *IEEE Access*, 7: 39794–39805.

[9] García-Martínez, M., Campos-Cantón, E. (2015). Pseudo-random bit generator based on multi-modal maps, *Nonlinear Dynamics*, 82(4): 2119–2131.

[10] Avaroglu, E. (2017). Pseudorandom number generator based on Arnold cat map and statistical analysis. *Turkish J. Electr. Eng. Comput. Sci.,* 25: 633-643.

[11] Avaroglu, E., Tuncer, T., Özer, A.B., Türk, M. (2015). A new method for hybrid pseudo random number generator, *J. Microelectron. Electron. Compon. Mater*., 44: 303–311

[12] Yakut, S., Tuncer, T., Özer, A. B. (2020). A New Secure and Efficient Approach for TRNG and Its Post-Processing Algorithms. Journal of Circuits, Systems and Computers.

[13] Avaroğlu, E., Tuncer, T. (2020). A novel S-box-based postprocessing method for true random number generation, *Turk. J. Elec. Eng. & Comp. Sci*., 28: 288–301.

[14] Garipcan, A. M., Erdem, E. (2020). A GRSÜ using chaotic entropy pool as a post-processing technique: analysis, design and FPGA implementation, *Analog Integrated Circuits and Signal Processing*, 103(3): 391-410.

[15] Łoza, Sz., Matuszewski Ł., Jessa M. (2015). A Random Number Generator Using Ring Oscillators and SHA-256 as Post-Processing, *Int. Journal of Electronics and Telecommunications*, 61(2): 199-204.

[16] Aljohani, M., Ahmad, I., Basheri, M., Alassafi, M. O. (2019). Performance analysis of cryptographic pseudorandom number generators, *IEEE Access*, 7: 39794-39805.

[17] Starosolski, R. (2020). Hybrid Adaptive Lossless Image Compression Based on Discrete Wavelet Transform, *Entropy*, 22(7): 751.

[18] Hudson, G., Yasuda H., Sebestyen I. (1988). The international standardization of a still picture compression technique, *IEEE Global Telecommunications Conference and Exhibition. Communications for the Information Age*, Hollywood, 1016-1021.

[19] Fuad, M., Ernawan F. (2020). Video steganography based on DCT psychovisual and object motion, *Bulletin of Electrical Engineering and Informatics*, 9(3): 1015-1023.

[20] Wedaj, F. T., Kim S., Kim H. J., et al. (2017). Improved reversible data hiding in JPEG images based on new coefficient selection strategy, *Eurasip Journal on Image & Video Processing*, (1): 63.

[21] Ajmera, A., Divecha M., Ghosh S. S., Raval I. and Chaturvedi R. (2019). Video Steganography: Using Scrambling- AES Encryption and DCT, DST Steganography, *2019 IEEE Pune Section International Conference (PuneCon)*, Pune, 1-7.

[22] Mao, B.H., Wang, Z.C., Zhang X.P. (2019). Asymmetric JPEG Steganography Based on Correlation in DCT Domain, *Computer Science*, 46(01):203-207.

[23] Ahmed, N., Natarajan T. and Rao K. R. (1974). Discrete Cosine Transform, *IEEE Transactions on Computers*, 23(1): 90-93.

[24] Al-Roithy, B.O., Gutub, A. (2021). Remodeling randomness prioritization to boost-up security of RGB image encryption, *Multimed Tools Appl*., 80: 28521–28581.

[25] Sumagita, M., Riadi I., Soepomo J.P., Warungboto U. (2018). Analysis of secure hash algorithm (SHA) 512 for encryption process on web based application, *Int J Cyber-Secur Digital for (IJCSDF),* 7(4): 373–381.

[26] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., Vo, S. (2010). *A statistical test suite for random and pseudorandom number generators for cryptographic applications*, NIST Special Publication, Gaithersburg.