JOURNAL OF ENERGY SYSTEMS

2021, *5*(4)

2602-2052

DOI: 10.30521/jes.1003454

Review Article

A new tool for energy security and secure energy communication

Erol Kurt

Gazi University, Technology Faculty, Department of Electrical and Electronics Engineering, Beşevler, TR-06500 Ankara, Turkey, ekurt52tr@yahoo.com

Batuhan Arpacı

Gazi University, Informatics Institute, Information Systems Department, Ankara, Turkey bthnrpc@gmail.com

Submitted: *02.10.2021*
Accepted: *02.12.2021*
Published: *31.12.2021*

Abstract: A recently proposed secure communication technique in Refs. [1,2,3] is initially applied to the energy sector. For this, especially the energy sector images which have a secret character for the companies and sectoral institutions have been ciphered and deciphered successfully. The applied tests have proven that the proposed method is fast and secure. The technique, itself, consists of a Kurt-modified Chua's circuit (KMCC) for the generation of chaotic number sequences. The KMCC is a non-autonomous nonlinear circuit having hyper-chaotic character, thereby two positive Lyapunov exponents can easily make a strong ciphering action. The method is efficient for the images used in energy plants and networks. The algorithm created for the encryption/decryption uses a scrambling feature implemented at the bit level.

Keywords: *Chua's circuit, Ciphering, Deciphering, Energy image, Energy plant, Secure communication*

Cite this paper as: Kurt, E., Arpacı, B., A new tool for energy security and secure energy communication. *Journal of Energy Systems* 2021; *5*(4): 376-389, DOI: 10.30521/jes.1003454

1. INTRODUCTION

Achievements in information and network technologies take the image security to an important position [1-3]. The secret communication has been used especially for important industrial projects and military applications. Among the industrial projects, the information security on energy systems is a key point for the companies as well as the countries. In the secret communication concept, the cryptography has been an attractive topic. While the energy partners communicate with each other, the images related to the energy plants, networks, etc should be transferred securely. For this aim, encryption/decryption processes can be used over the images of energy systems. Meanwhile, a problem arises. Indeed, traditional encryption methods such as AES, IDEA, DES have certain security flaws since they can be decrypt by the conventional techniques as in Refs. [4-6].

Among the secure communication issues of coloured images, there exist two well-known processes called permutation and diffusion. Although they can be used for image encryption procedures, their implementation for a bit or pixel level cannot satisfy a required security. For this reason, realization of only the exchange property in the bit level would not give sufficient security, especially in permutation and diffusion [6]. Therefore a combined technique with sensitive, ergodic and random can be applied. Those features quantifies a chaotic system indeed. According to the literature, many scholars have used chaos-based encryption systems [7-10]. In the present paper, to our knowledge, for the first time, we apply this methodology with an advanced algorithm and chaotic output to the energy sector. The reason for that is increasing importance of energy plants, internet attacks to energy plants, and project-based secure communication requirements. Since energy has been a strategic concept for countries, such a secure communication technique can assist to provide the transfer of the energy-related images. For this aim, a new chaos-based algorithm has been used.

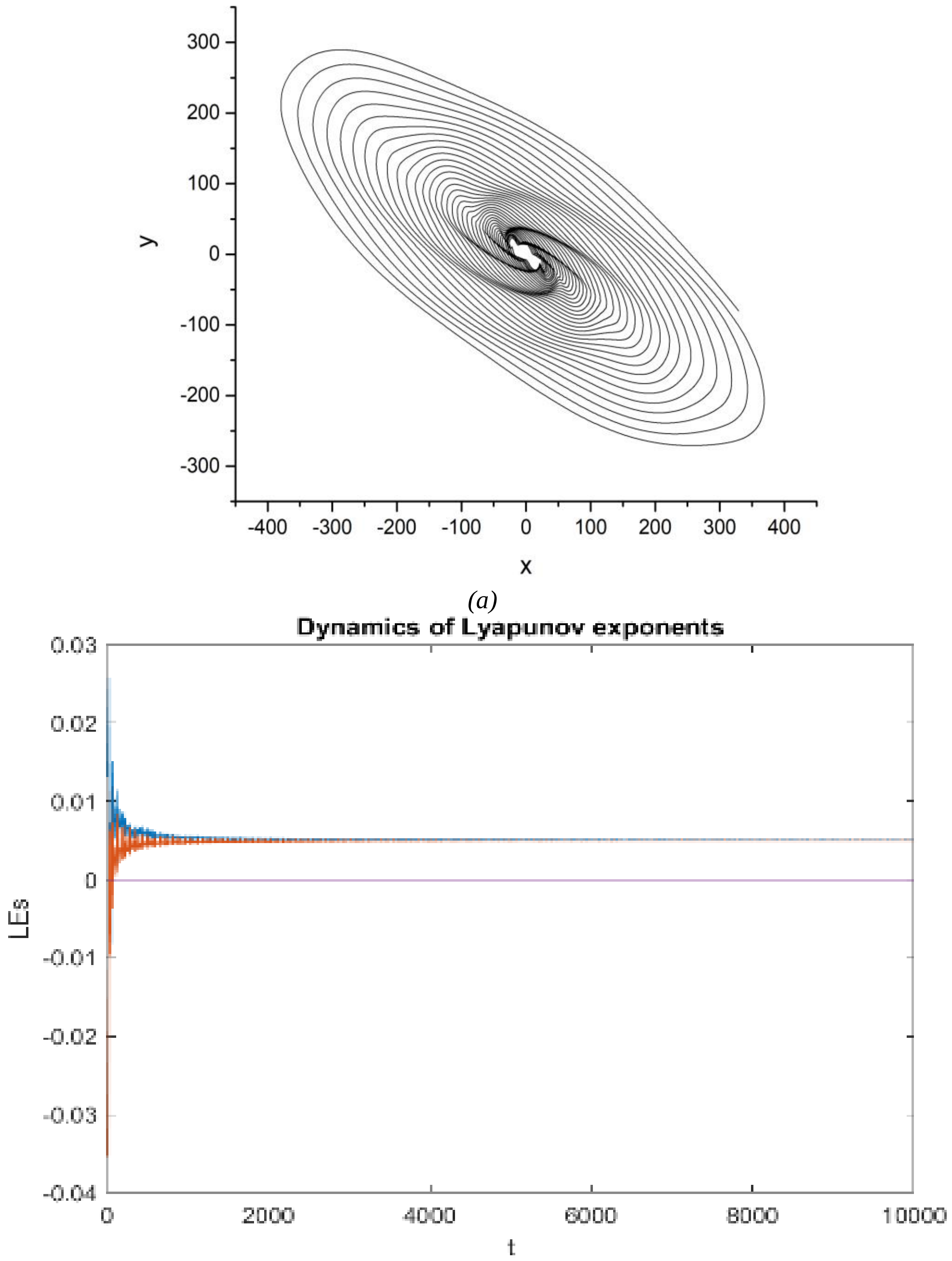
In this paper, the novel part comes from the topic of energy security, invented algorithm and the usage of Kurt modified Chua's circuit (KMCC) in the ciphering/deciphering processes. The image security has been a continuous importance due to conventional and renewable energy sector, thereby this area as energy security will be an increasing trend for the energy independence of the countries. Section 2 introduces the hyperchaotic Kurt-modified Chua's circuit. The image encryption and main experimental results are presented in Section 3 and 4, respectively. The main test results are also discussed in Section 4. Finally, the paper ends with a concluding section.

2. HYPERCHAOTIC SYSTEM DEFINITION

In this study, a Kurt-modified Chua's circuit (KMCC) is used. This circuit is mainly used to get random numbers. However the generated numbers should have a chaotic character. Indeed, we use this circuit since it can generate hyperchaotic data which gives a strong randomness in the phase space of system depending on the parameters. The system equations are as follows [11]:

$$\begin{cases} \dot{x} = y - bx - \tfrac{1}{2}(a-b)\left[\,|x+\sin(z)| - |\,x-\sin(z)\,|\,\right], \\ \dot{y} = -\beta(y+x) + f\sin(v), \\ \dot{z} = \phi, \\ \dot{v} = \omega \end{cases} \tag{1}$$

Here, $a, \mathrm{b}, \phi, \beta, \omega, f$ are nothing else than the control parameters. The solutions run in MatLab environment via the well-known Runge-Kutta method as shown in Fig. 1(a,b).



(a)

(b)

Figure 1. (a) The attractor from Kurt-modified Chua's circuit for the parameters $a = -2.91$, $b = -0.56$, $\beta = 0.55$, $f = 12.99$, $\phi = -15.1$ *and* $\omega = 2.91$, *(b) corresponding Lyapunov spectrum.*

In a chaotic system, at least one positive exponent should exist. If there are two positive exponents, the system is called as "hyperchaotic" as shown in Fig. 1(b). In such situation, depending on the initial conditions, two positive lyapunov exponents such as (0,0, +, +) determines the hyperchaoticity.

3. IMAGE ENCRYPTION SCHEME

The image encryption procedure worked in this paper has been explained in Refs. [1,2,3], clearly. Therefore, in the current paper we do not give the details. However, as a brief explanation, we explain the secret key generation, initial value definition, and algorithms as a background information.

In our technique, as a cryptographic hash algorithm, SHA-256 generates a hash value of 256-bit. It varies entirely, when a slight change occurs in any input part of the algorithm as one expects. Indeed, a digest output with 48-bit also determined as *PK* has been obtained from the plain image for the input information to the function of SHA-256. Random noise (i.e. *RN*) has already been produced at the start of each encryption scheme. Note that a digest hash value of 256-bit with *SK* is later created by processing an SHA-256 with both *PK* and *RN* inputs. Thereby, the generated secret key has been entirely unique due to the function of SHA-256, even if there exist a slight difference in the images, or even no differences at all. The row, column, color and other information algorithms from the hash key definition can be found clearly in Refs. [1,2].

The initial values for the functions x_1, y_1, z_1, v_1 and can be obtained as following:

$$\begin{cases} x_1' = \left(hex2de\left(subset\left(1,10,SK\right)\right)10^{-11}\right) \\ \qquad\qquad + \left(hex2de\left(subset\left(11,16,SK\right)\right)10^{-14}\right) \\ y_1' = \left(hex2de\left(subset\left(17,26,SK\right)\right)10^{-11}\right) \\ \qquad\qquad + \left(hex2de\left(subset\left(27,32,SK\right)\right)10^{-14}\right) \\ z_1' = \left(hex2de\left(subset\left(33,42,SK\right)\right)10^{-11}\right) \\ \qquad\qquad + \left(hex2de\left(subset\left(43,48,SK\right)\right)10^{-14}\right) \\ v_1' = \left(hex2de\left(subset\left(49,58,SK\right)\right)10^{-11}\right) \\ \qquad\qquad + \left(hex2de\left(subset\left(59,64,SK\right)\right)10^{-14}\right) \end{cases} \tag{2}$$

Note that here the function $hex2de(.)$ transforms the secret key from a hexadecimal number to a decimal one given by $subset\,(i,\,j,K)$. Then, it gives elements between the indexes *i*th and *j*th of the *K 1-D* array. In Eq. (2), the multiplications are determined as 10^{11} and 10^{14} for the adjustment of the relevant decimals of x_1, y_1, z_1 and v_1. The values of a_1, a_2, b_1 and b_2 are found according to the program flows in Ref. [3,12]. Note that the subset and functions hex2de are determined above. Besides, the algorithm also uses the function $concat(./.)$ to concatenate the values given into it. In addition, the function $roundD(.)$gives the decimal part of a given decimal number and $sum(.)$ aggregates the function in this regard.

$$\begin{cases} x_1 = x_1'\left(2-a_1\right) \\ y_1 = y_1'\left(2-a_2\right) \\ z_1 = z_1'\left(2-b_1\right) \\ v_1 = v_1'\left(2-b_2\right) \\ f = 9.1 + a_1 \end{cases} \tag{3}$$

Here, 9.1 gives a low chaotic parameter for *f* in order to cipher the image. The elements $a_{1,2}$ and $b_{1,2}$ refer to the numbers smaller than 1. Thus one should make the multiplication higher by introducing the term $\left(2-(a,b)_{1,2}\right)$.

Fig. 2 presents the encryption procedure. This algorithm has superiorities on the algorithms existing in the literature. Indeed, it divides the image into 2 pieces shown in Fig. 2. Here *Input* is a Plain image *P* and a secret key *SK*, whereas *Output* is a Cipher image *C*. For the horizontal and vertical magnitudes, *W* and *H*, the size of the image is given by $W \times H \times 3$.

***Step 1*:** The initial values (x_1, y_1, z_1, v_1) are taken as input. The initial parameter *f* of the system is determined by using Eq. (3).

***Step 2*:** The iteration method is used and the chaotic number sets with array *CN* and size are $(s \times 4) + 5000$ generated by solving time integration from the KMCC system. Following the first 1000 data, one uses *n*=4*s* and *cs*=*n*+4000.

Step 3: CN is produced by the chaotic generator. The key matrix *KM* is applied in the diffusion and scrambling stages.

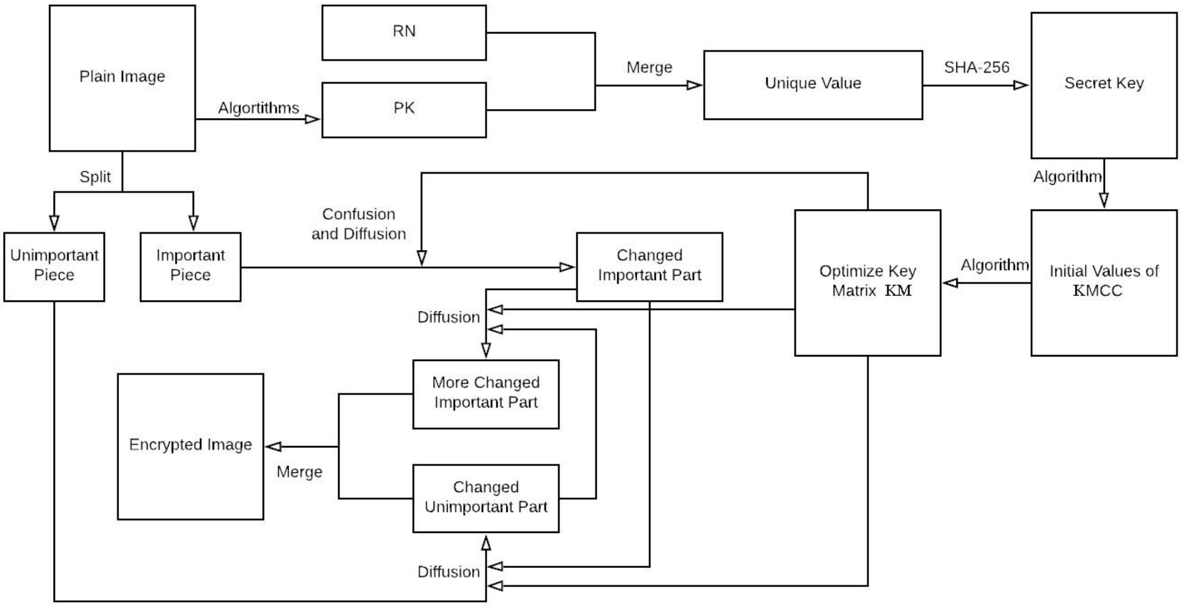


Figure 2. The flow chart of encryption process.

Step 4: Plain image *P* is resized for each pixel by beginning from component *R*, sequentially from upper to bottom point, then left to right with *G* and *B* components. Then each pixel is converted into 8-digit binary format.

Step 5: Mapping method is applied to the PB_2 matrix using the *KM* key matrix.

Step 6: Diffusion method is applied to matrices PB_1 and PB_2 using by key matrix *KM*.

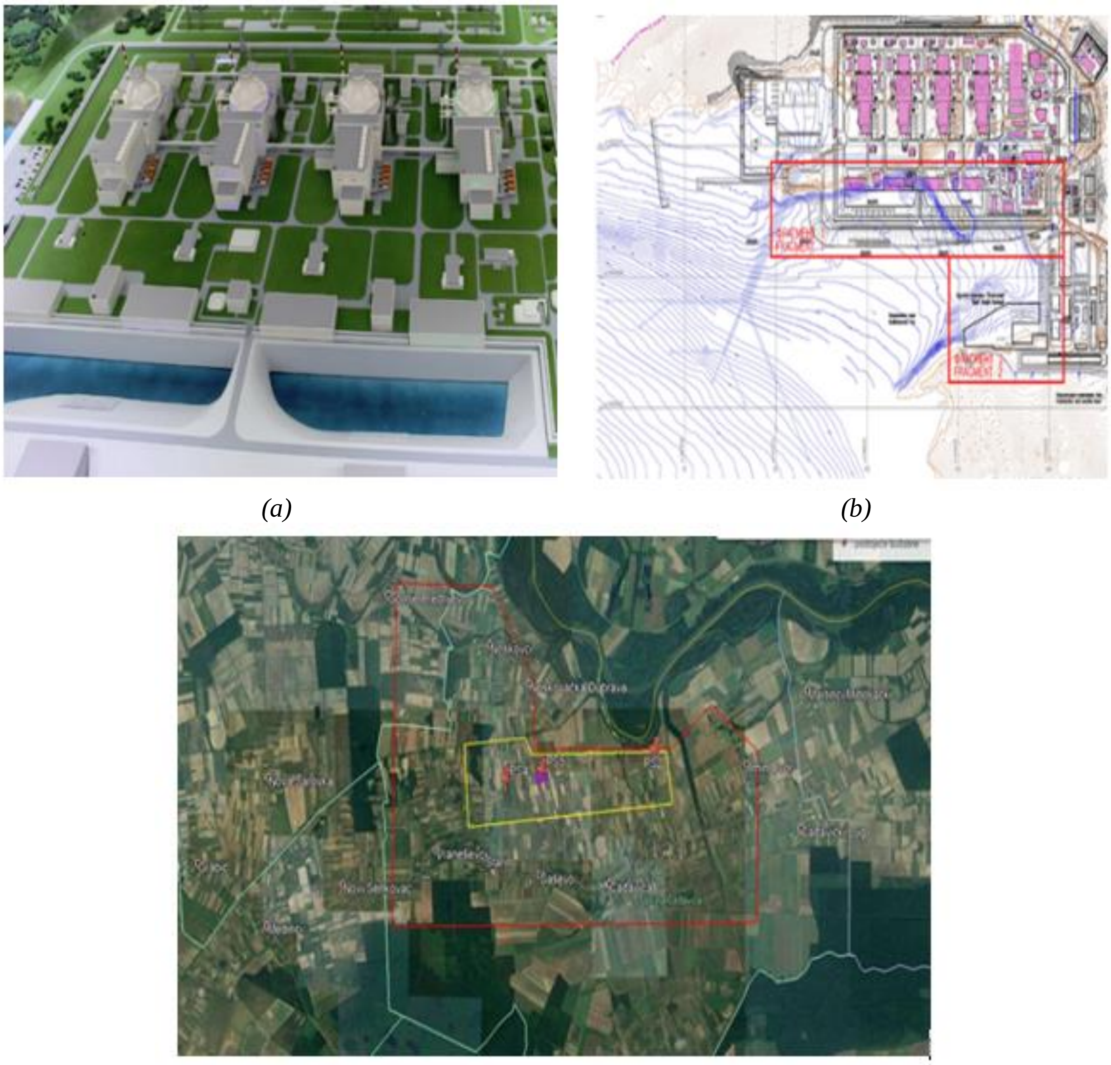
The vice-versa of this algorithm is used in the decryption process. The details can be found in Ref. [1, 2, 12].

4. EXPERIMENTAL RESULTS AND RESPONSES TO THE SECURITY TESTS

The initial parameters of KMCC are $a = -2.91$, $b = -0.56$, $\beta = 0.55$, $\phi = -0.13$,and $\omega = 1.29$ for the experiment. The system is hyperchaotic for $f \geq 9.1$. The secret key is,

2A8649DDF54B044DC1A50329C54B4960010066BA8FD005D4392B536545B04ECE.

The initial state variables and driving amplitude f of KMCC are obtained from this secret key as used in similar procedure in Ref. [13]. We have used several images of energy plants and energy projects. Their names are 3Dnuclear, 2Dnuclear, and Plant as shown in Figs. 3(a-c), respectively.

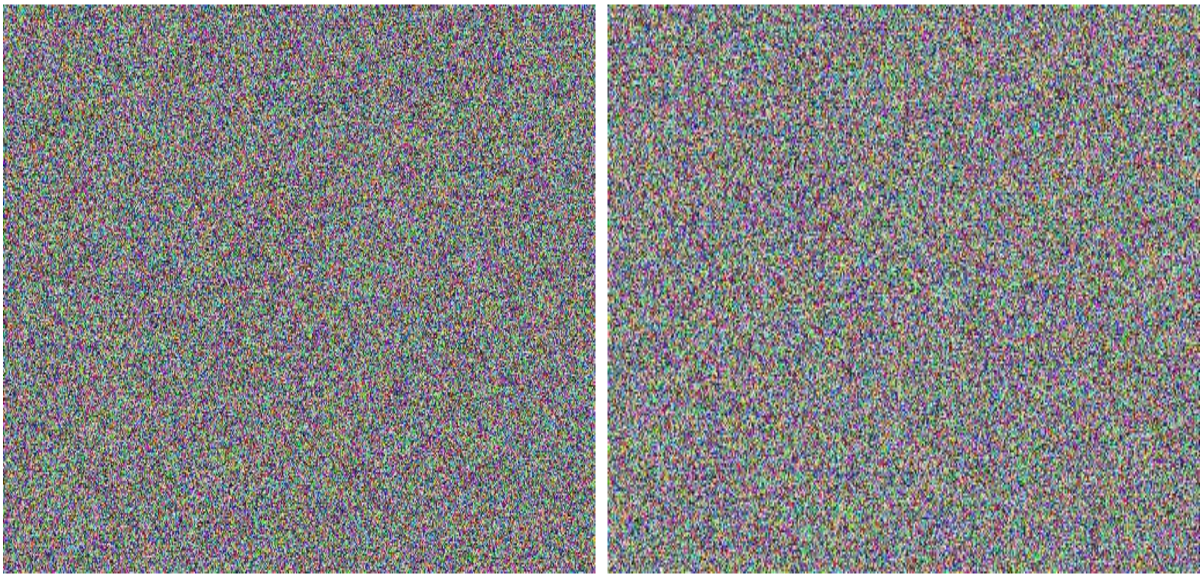


(a) (b)

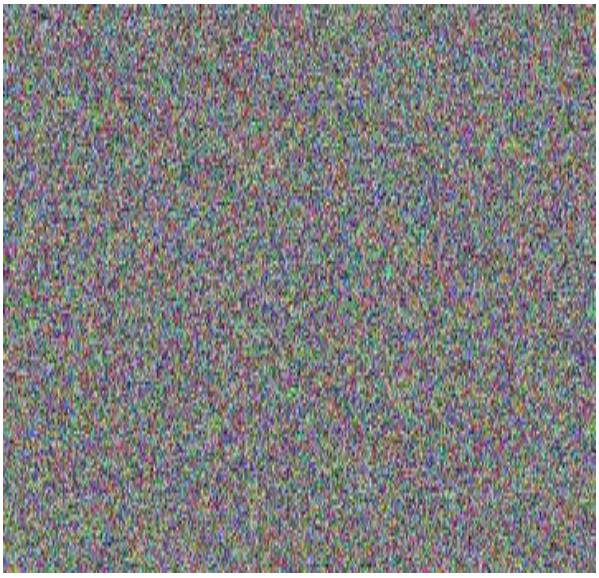
(c)

Figure 3. Energy plant images used for ciphering/deciphering

The images have been ciphered by following the algorithm in Fig. 2 and pictured as in Fig. 4(a-c).



(a) (b)



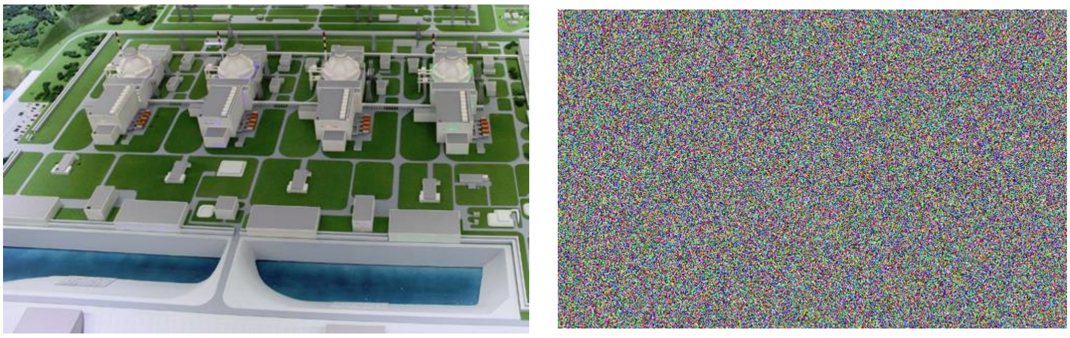
(c)

Figure 4. The ciphered forms of energy plant images in Figs. 3(a-c), respectively.

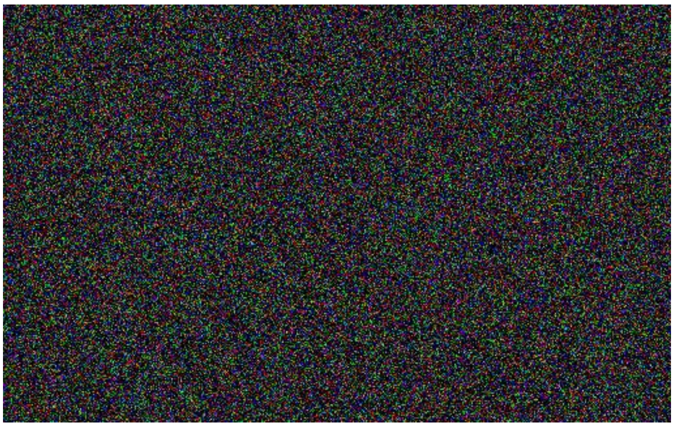
It is obvious that all the ciphered images cannot be seen truly with a naked eye. However, it is not sufficient, since the responses to the security tests should be discussed in order to be at the safe side. For this reason, we have applied several security tests including key space, plain image sensitivity and key sensitivity, resistance against a known plaintext and chosen plaintext attacks, differential attacks, information entropy, histogram, correlation coefficient, and resisting noise attack analyses, respectively.

For the key space analysis, the key space must be capable to neutralize the brute-force attacks. The encryption procedure key consists of the initial values given by (x_1, y_1, z_1, v_1) and initial parameter of f as also stated in the previous section. For the chaotic characteristics, the precision of the initial conditions muct be as high as possible, for instance, 14 or 15 digits after the comma are required [5], so that the key space can reach 10^{70}. In that case, the key space is $S = 10^{70} \cong 2^{232} > 2^{100}$ [3], so that the cryptosystem can resist to the brute-force attacks.

The key of the KMCC powered ciphering/deciphering system is a 'nonce', based on a hash value created by the plain image and a random sequence. Thus, if the starting conditions of the ciphering system is changed slightly, this operation would yield to a creation of a diverse encrypted image. In the KMCC system, taking into the experimental results, it is achieved that the algorithm is delicate to the smallest variation in the key. Fig. 5(a) is only one bit modified version of Fig. 3(a) image and its encrypted state is shown in Fig. 5(b). The differences between Figs. 4(a) and 5(b) are also shown in Fig. 5(c). It is obvious that the results of the encryptions are also diverge from each other.



(a) *(b)*



(c)

Figure 5. Test results of key sensitivity test: (a) 1 bit-modified version of Fig. 3(a), (b) its encrypted image, and (c) the difference between the images in Figs. 4(a) and Figs. 5(b).

In the proposed system shown in Fig. 2, the key strictly depends on the hash value of an original image file. In this case, different keys should be created for different images. An attacker cannot decipher a particular image with a key obtained from another image. In this case, the implemented software can be declared as “resistant” to the known - plaintext and chosen - plaintext attacks.

Frequently, it is hoped that that the encrypted material should be different from its raw version in any image encryption system. In order to measure such a difference, the criteria NPCR [14] and UACI [15] are frequently applied to the image. The ciphering/deciphering system which is recommend in the present work should guarantee that the encrypted version of two plain images should be dissimilar to each other, if one-bit modification is applied in one of them. With this respect, Tables I and II show the NPCR and UACI results of 1500 randomly selected pairs. It is obvious that this test gives satisfactory values for the ciphering/deciphering system. To conclude, the software is found to be robust against any differential attacks in this manner.

Information entropy is applied for a measurement of an arbitrary distribution in a media file. The formulation of the operation is presented as follows [16]:

$$H(m)=\sum_{i=0}^{2^n-1} p(m_i)\log_2 \frac{1}{p(m_i)} \tag{4}$$

The information entropy of any encrypted plain image should be as large as possible. Strictly speaking, it should be 8 as discussed in Ref. [17]. This reality makes information difficult to reveal. Table 3 shows the information entropy values of 3 pieces of an encrypted image by using Eq. (4). This equation also proves that the optimum value should be 8.

Table 1. The minimum, maximum and average values when UACI(%) is applied.

Image	*R*			*G*			*B*		
	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min
3Dnuclear	31.4156	30.6879	30.3479	30.3825	28.6145	28.3108	33.2982	32.1271	31.3550
2Dnuclear	39.2922	33.6726	33.4281	38.2511	35.5371	33.4115	39.5920	33.7994	33.4635
Grid	31.4428	30.8832	30.3841	31.4087	30.3856	27.8536	33.3667	31.3421	29.4814

Table 2. The minimum, maximum and average values when NPCR(%) is applied.

Image	*R*			*G*			*B*		
	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min
3Dnuclear	99.6106	99.5642	99.5012	99.6595	99.5556	99.5044	99.6693	99.5764	99.5123
2Dnuclear	99.1184	98.4079	98.2753	99.0128	98.9136	98.5852	99.4512	99.1001	98.9081
Grid	99.6029	99.5814	99.5516	99.6152	99.5844	99.5814	99.6649	99.5491	99.4421

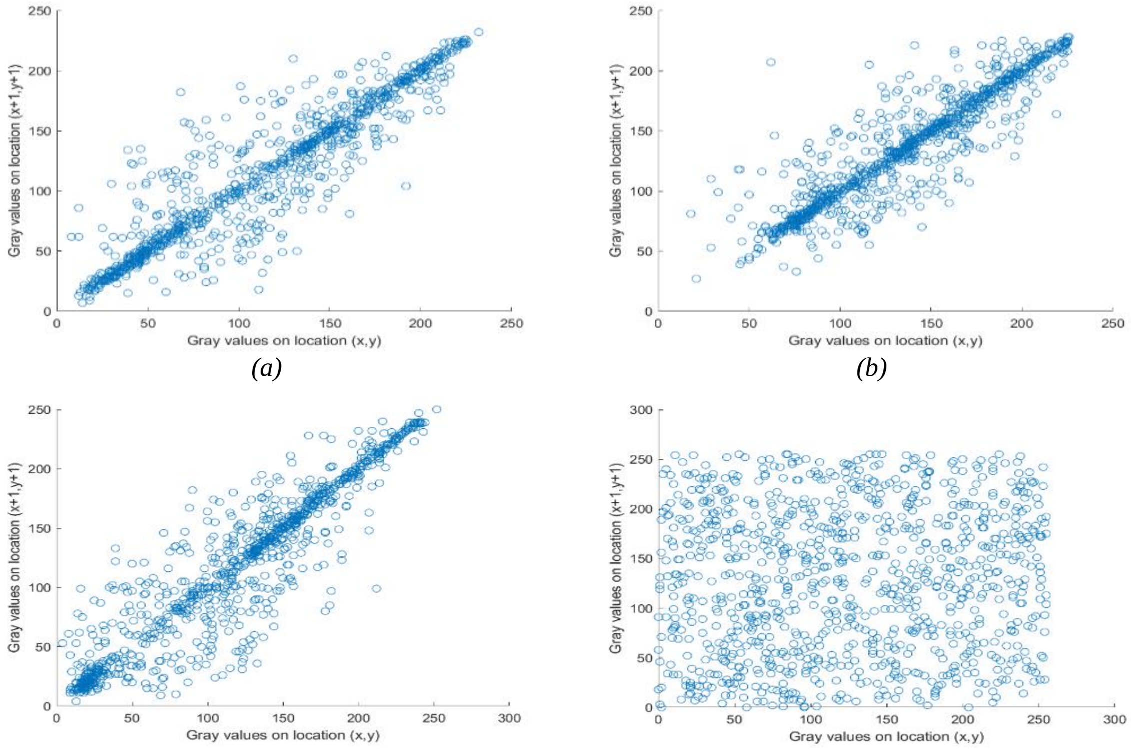
Table 3. The information entropy values of the ciphered images.

Image	*R*	*G*	*B*
3Dnuclear	7.9902	7.9897	7.9902
2Dnuclear	7.9271	7.9261	7.9362
Grid	7.9888	7.9887	7.9888

There exists a relationship between adjacent pixels in an original image. To apply the statistical attacks, the correlation of adjacent pixels in the encrypted image should be minimum. Otherwise, one can estimate the ciphered image. The formulation below should be applied in order to calculate the correlation between two adjacent pixels in this regard [18]:

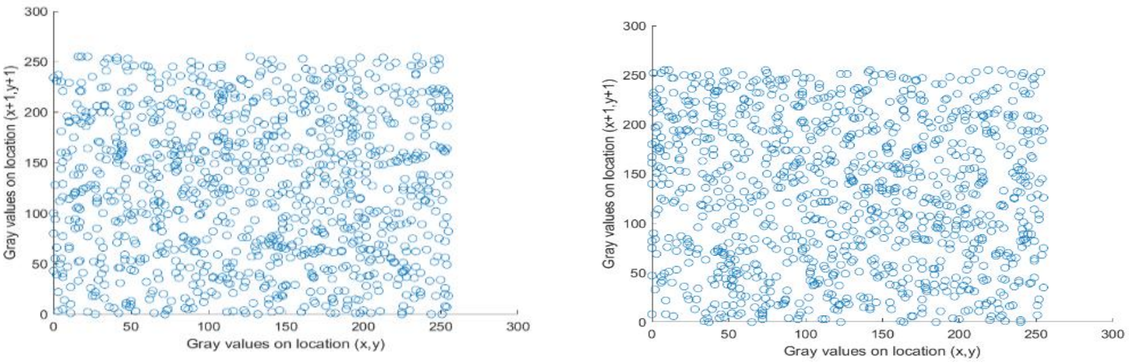
$$r_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \ \operatorname{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)),$$
$$E(x) = \frac{1}{N}\sum_{i=1}^{N}x_i, \qquad D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2. \tag{5}$$

Fig. 6 shows the correlation distributions on the plain images of energy plants. Here there are two horizontally, vertically and diagonal adjacent pixels. Note that one is received from the plain and the other is received from the ciphered image in Fig. 3(a). One can understand that the correlation between the adjacent pixels decreases with a large amount. Table 4 indicates the correlation between images and their encrypted versions. The findings proves that the correlation between the neighboring pixels of their encoded plain images is very small. But, it is also seen that the correlation between the plain images is quite high, so the encryption is found to be effective.



(a) *(b)*

(c) *(d)*



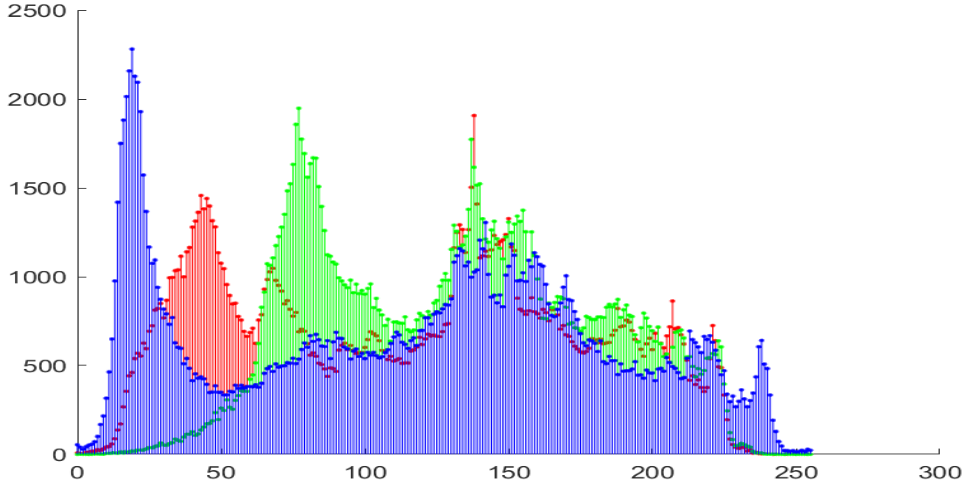
(e) *(f)*

Figure 6. Correlation distributions between the plain and the encoded images: (a), (b), (c) are diagonal, vertical and horizontal of energy plant image 3Dnuclear. (d), (e), (f) are the diagonal, vertical and horizontal of ciphered form, respectively.

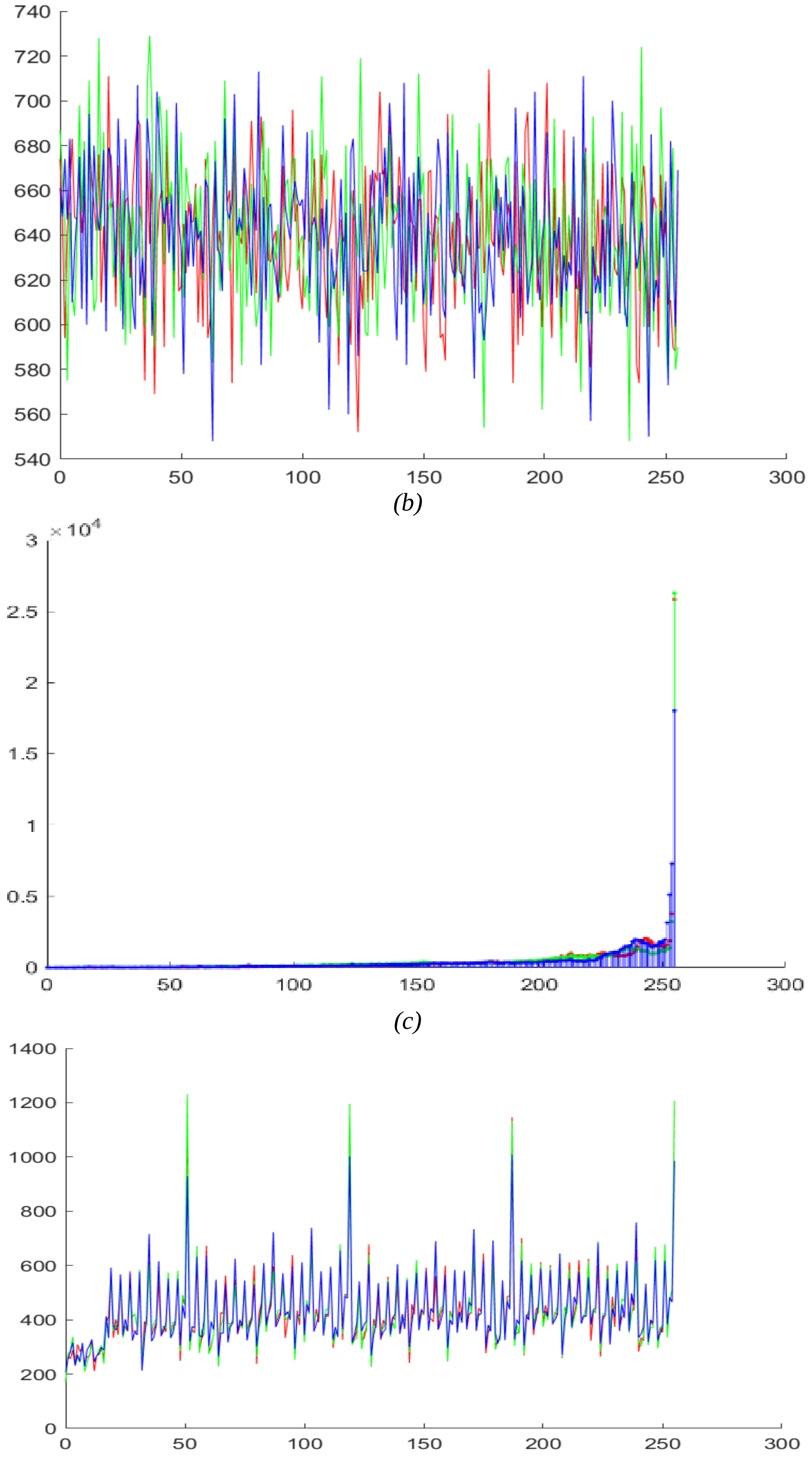
Table 4. The correlation coefficients for adjacent pixels in the original images and their ciphered forms.

Images	*Directions*		*Original*			*Encrypted*	
		R	G	B	R	G	B
	Diagonal	0.9293	0.9133	0.9278	0.0691	0.0020	-0.0658
3Dnuclear	Vertical	0.9534	0.9328	0.9534	0.0476	-0.0213	0.0217
	Horizontal	0.9711	0.9711	0.9762	0.0284	-0.0233	0.0623
	Diagonal	0.6402	0.7254	0.6734	-0.0328	0.0635	0.0001
2Dnuclear	Vertical	0.7477	0.7672	0.7456	0.0069	-0.0082	-0.0274
	Horizontal	0.8446	0.8736	0.8904	-0.0004	-0.0010	-0.0245
	Diagonal	0.8638	0.8004	0.7974	-0.0039	-0.0052	-0.0061
Grid	Vertical	0.9031	0.8996	0.8296	-0.0069	-0.0314	-0.0203
	Horizontal	0.9177	0.9013	0.8880	-0.0035	-0.0094	0.0201

The histogram results of the energy plant image provides information about the distribution of its pixel values and shows the image. As seen in Fig. 7(a-d), the histogram analysis of the raw image has several peaks associated with colors, while the encrypted image has a nearly constant distribution. This proves that the deciphered image is a good candidate for the hidden image.



(a)



(b)

(c)

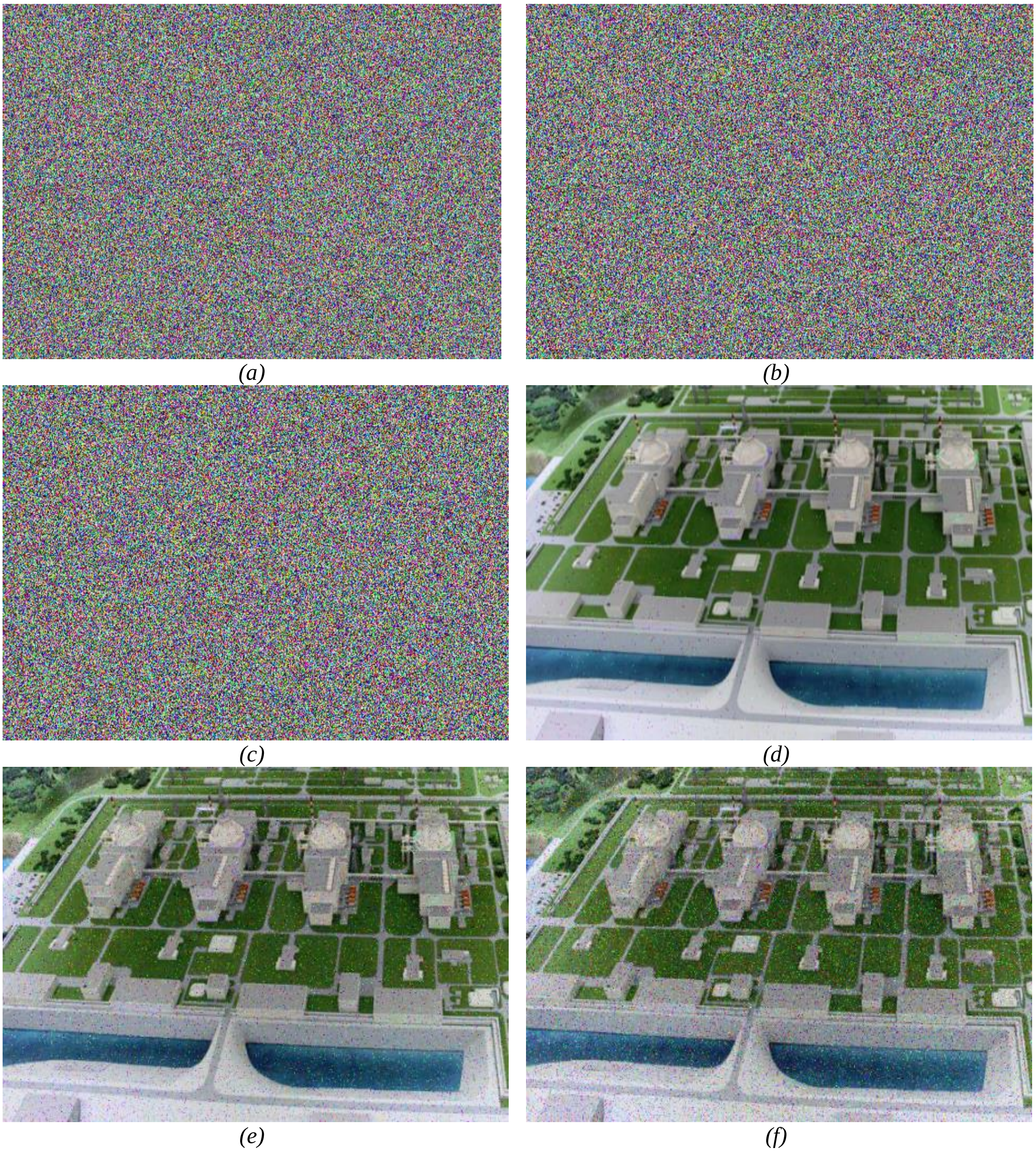
(d)

Figure 7. (a,b) Histogram analyses of the plain and encrypted images of 3Dnuclear, respectively. (c,d) Histogram analyses of the plain and encrypted images of 2Dnuclear, respectively.

The ciphered plain image is exposed to many kinds of noise application. However, it passes through the real communication channels and the noise can cause problems during the raw plain image acquisition. For this reason, the algorithm should be resistant to the noise so that the encryption can have high validity. The Peak Signal-to-Noise Ratio (PSNR) is applied in order to measure the quality of the decoded plain image after the attack. For the plain image components, PSNR can be calculated as follows [19]:

$$PSNR = 10 \times \log_{10}\left(\frac{255 \times 255}{MSE}\right)(dB) \qquad MSE = \frac{1}{mn}\sum_{i=1}^{m}\ \sum_{i=1}^{n}\left\| I_1(i,j) - I_2(i,j) \right\|^2 \qquad (6)$$

MSE is the mean square error between the original and recovered images and is represented as $I_1(i,j)$ and $I_2(i,j)$ respectively, with the size of *mxn*. Figs. 8(a-c) shows the encrypted images exposed to the Salt Pepper noise with different density of this and its deciphered ones. The MSE and PSNR of these decoded images are shown in Table 5. It is obvious that the original image is entirely obtained again, which is noticeable, the PSNR value is about 30 *dB*, and the decoded images are highly correlated (Fig. 8(d-f)). This means that the decoded images are very close to the original image. Thus, it can be said that the proposed algorithm is resistant to resisting noise attacks to some degree.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 8. The ciphered images with salt & pepper noise and their deciphered forms of 3Dnuclear. The noises with (d) d=0.01, (e) d=0.05 and (f) d=0.1, respectively.

Table 5. The quantitative results of resisting noise attacks.

Image	*Density*	*MSE*			*PSNR*			*Correlation*		
		R	G	B	R	G	B	R	G	B
3Dnuclear	0.01	108	109	113	27.7777	27.7514	27.6186	0.9950	0.9945	0.9944
	0.05	548	537	542	20.7406	20.8270	20.7876	0.9758	0.9755	0.9761
	0.1	1072	1067	1091	17.8274	17.8475	17.7529	0.9554	0.9471	0.9580
2Dnuclear	0.01	110	107	115	27.7324	27.8219	27.5190	0.9921	0.9950	0.9919
	0.05	557	571	558	20.6741	20.5640	20.6610	0.9633	0.9655	0.9665
	0.1	1152	1151	1082	17.5165	17.5193	17.4484	0.9240	0.9269	0.9369
Grid	0.01	106	103	113	27.8674	28.0215	27.6126	0.9911	0.9924	0.9882
	0.05	539	536	533	20.8138	20.6828	20.8599	0.9506	0.9484	0.9365
	0.1	1071	1073	1074	17.8315	17.8234	17.8210	0.9039	0.8984	0.8730

5. CONCLUSIONS

A new image security tool is proposed for the application of energy maps and plants. Initially, the raw encryption/decryption algorithm is defined for both encryption and decryption purposes. For the encryption/decryption, the random numbers from a hyperchaotic electrical circuit, namely Kurt-modified Chua's circuit are used. The scrambling characteristics, which is implemented at a bit level and novel diffusion system by using the hyperchaotic nature of the circuit are applied in the proposed algorithm. Following the encryption, some important tests are applied to the ciphered image in order to prove the ciphering ability. The encrypted colored images are tested by many test types including the secret key size and secret key sensitivity, correlation analysis, histogram analysis, differential analysis and information entropy analysis. It has been proven that the proposed image security tool can be used for energy- related secure communication issues. The algorithm is found to be quite effective and fast enough for the energy industry.

REFERENCES

[1] Arpacı, B, Kurt, E, Çelik, K. A new algorithm for the colored image encryption via the modified Chua's circuit. *Engineering Science and Technology, an International Journal* 2020; *23*(3): 595-604. DOI:10.1016/j.jestch.2019.09.001.

[2] Arpacı, B, Kurt, E, Çelik, K. Ciylan, B., Colored image encryption and decryption with a new algorithm and a hyperchaotic electrical circuit. *J. Electr. Eng. Technol.* 2020; *15*: 1413–1429. DOI: 10.1007/s42835-020-00393-x

[3] Arpacı, B, Kurt, E. An Innovative Tool for the Chaotic Image Encryption, Decryption and Security Tests. In: ICECCE 2020 Int. Conf. Electr., Commun. Computer Engin.*;* 12-13 June 2020: IEEE, pp. 1-8. DOI: 10.1109/ICECCE49384.2020.9179274.

[4] Alvarez, G, Li, S. Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bif. Chaos* 2006; *16*(08): 2129-2151.

[5] Kiraz, MS, Uzunkol, O. Efficient and verifiable algorithms for secure outsourcing of cryptographic computations. *Int. J. Information Security* 2016; *15*(5): 519-537.

[6] Stinson, DR. Cryptography: Theory and practice. CRC press. 2005

[7] Zhu, ZL, Zhang, W, Wong, KW, Yu, H. A chaos-based symmetric image encryption scheme using a bit-level permutation. *Information Sci.* 2011; *181*(6): 1171-1186.

[8] Fu, C, Lin, BB, Miao, YS, Liu, X, Chen, JJ. A novel chaos-based bit-level permutation scheme for digital image encryption. *Optics Commun.*2011; *284*(23): 5415-5423.

[9] Wang, Y, Wong, KW, Liao, X, Xiang, T, Chen, G. A chaos-based image encryption algorithm with variable control parameters. *Chaos, Sol. & Fractals* 2009; *41*(4): 1773-1783.

[10] Guan, ZH, Huang, F, Guan, W. Chaos-based image encryption algorithm. *Phys. Let. A* 2005; *346*(1-3): 153-157.

[11] Xiao, D, Liao, X, Wei, P. Analysis and improvement of a chaos-based image encryption algorithm. *Chaos, Sol. & Fractals* 2009*; 40*(5): 2191-2199.

[12] Kurt, E. Nonlinearities from a non-autonomous chaotic circuit with a non-autonomous model of Chua's diode. *Physica Scripta* 2006; *74*(1): 22.

[13] Liu, H, Wang, X, Kadir, A, Chaos-based color image encryption using one-time keys and Choquet fuzzy integral. *Int. J. Nonlinear Sci. Numerical Sim.* 2014*; 15*(1): 1-10.

[14] Zhang, Y, Xiao, D. Self-adaptive permutation and combined global diffusion for chaotic color image encryption. *AEU-Int. J. Electronics and Commun*. 2014; *68*(4): 361-368.

[15] Seyedzadeh, S. M, Mirzakuchaki, S. A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. *Signal Proces.* 2012; *92*(5): 1202-1215.

[16] Wang, XY, Chen, F, Wang, T. A new compound mode of confusion and diffusion for block encryption of image based on chaos. *Commun. Nonlinear Sci. Numerical Sim*. 2021; *15*(9): 2479-2485.

[17] Mirzaei, O, Yaghoobi, M, Irani, H, A new image encryption method: parallel sub-image encryption with hyper chaos. *Nonlinear Dyn*. 2012; *67*(1): 557-566, 2012.

[18] Rhouma, R, Meherzi, S, Belghith, S. OCML-based colour image encryption. *Chaos, Sol. & Fractals* 2009; *40*(1): 309-318.

[19] Chai, X, Gan, Z, Zhang, M. A fast chaos-based image encryption scheme with a novel plain image-related swapping block permutation and block diffusion. *Multimedia Tools and Applications* 2017; *76*(14): 15561-15585.