Nonlinear Dyn (2013) 72:399–406
DOI 10.1007/s11071-012-0723-5

ORIGINAL PAPER

A novel image encryption algorithm based on chaotic maps and GF(2^8) exponent transformation

Iqtadar Hussain · Tariq Shah · Muhammad Asif Gondal · Hasan Mahmood

Received: 11 October 2012 / Accepted: 16 December 2012 / Published online: 10 January 2013

Abstract In this paper, we propose an image encryption algorithm that is based on GF(2^8) transformations, using the Arnold cat map and incorporating the nonlinear chaotic algorithm. The plain image is processed with the nonlinear chaotic algorithm and is shuffled iteratively with the Arnold cat map, while transforming the image pixel values into GF(2^8). We show that the encryption characteristics of this approach are better as compared to some well known encryption algorithms.

Keywords Image encryption · Arnold cat maps · Nonlinear chaotic algorithm

I. Hussain (✉) · T. Shah
Department of Mathematics, Quaid-i-Azam University, Islamabad, Pakistan
e-mail: iqtadarqau@gmail.com

T. Shah
e-mail: stariqshah@gmail.com

M.A. Gondal
Department of Sciences and Humanities, National University of Computer and Emerging Sciences, Islamabad, Pakistan
e-mail: asif.gondal@nu.edu.pk

H. Mahmood
Department of Electronics, Quaid-i-Azam University, Islamabad, Pakistan
e-mail: hasan@qau.edu.pk

1 Introduction

In this fast growing telecommunication field, huge volumes of data are transmitted on unsecure transmission lines, and it is imperative to ensure security of individual users who utilize these communication facilities. In order to mitigate the cryptanalysts deciphering methods, robust and efficient cryptographic algorithms are required. The chaotic systems, with their attractive properties such as unpredictability, randomness, and control by virtue of initial conditions, are becoming popular in encryption applications [1, 2]. Several approaches are seen in the literature that applies to concepts from the chaotic systems. In [3], a hyperchaotic encryption scheme is presented. This scheme shuffles the image matrix and creates confusion by application of hyperchaotic systems. The use of the chaotic Kolmogorov is also demonstrated in the application to the encryption of data [4]. While the use of chaotic approach in two- dimensional system provides some interesting results, the application of 3D chaotic cat maps assisted in establishing more secure systems [5]. The cat map based algorithms provide the shuffling capability and exhibit inherent property of repeating within a definite time period. The Arnold cat map [5, 6] exhibits this property, therefore, in this work, we employ an encryption system that overcomes the issues of small time period, which is inherent in this system. A detailed study of nonlinear components of block ciphers is presented in [10–27].

Table 1 Cameraman's iterated period

		f									
		1	2	3	4	5	6	7	8	9	10
e	1	192	128	192	256	96	64	96	256	192	128
	2	128	128	64	128	128	128	32	128	128	128
	3	192	64	192	256	48	48	96	056	192	16
	4	256	128	256	64	256	256	256	64	256	128
	5	96	128	48	256	192	192	192	256	48	128
	6	64	128	128	128	16	16	128	128	64	128
	7	96	32	96	256	192	192	192	256	12	64
	8	256	128	256	64	256	256	256	32	256	128
	9	192	128	192	256	48	64	12	256	192	128
	10	128	128	16	128	128	128	64	128	128	128

2 The proposed encryption algorithm

The proposed image encryption algorithm is a two-step process. In the first step, the objective is to shuffle the pixel locations in the original image. The second step employs the nonlinear chaotic algorithm (NCA) chaotic map to encrypt the image.

2.1 Arnold cat map

The Arnold cat map is classified as a two-dimensional invertible chaotic map [5, 6]. The discrete form of this two-dimensional map with dimensions of $M \times M$ is defined as

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod M. \tag{1}$$

In Eq. (1), the original location of the pixel (x_n, y_n) is transformed to new coordinates, (x_{n+1}, y_{n+1}). The positive integers a, b, c, and d are used in this transformation, with the condition that $ad - bc = 1$. All the quantities in Eq. (1) are integers and are bounded by the dimension "M" where M can take values $\{0, 1, 2, \ldots, M-1\}$ and the values of a, b, c, and d are chosen in such a way that $ab - cd$ is always equal to 1. Equation (1) can be modified as

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & e \\ f & ef+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod M. \tag{2}$$

The value of the secret key can be mapped as suitable parameters in Eq. (2). For example, the parameters e and f in combination with number of iterations N, can be extracted from the key, and incorporated in the Arnold cat map. With the progression of iterations, the pixels are randomized, and ultimately a stage arrives, which restores the original locations of the pixels. In other words, if the period is represented by T, the original image is reconstructed after T number of iterations. The choice of N is critical with reference to the inherent period T that depends upon other initial conditions. In image encryption applications, it is desired to have a large value of Cameraman's periods, given as T. Table 1 lists the Cameraman's period for Arnold cat map with parameters f and e from Eq. (2). It is advisable to choose the number of iterations by keeping in view the period and its implications.

2.2 NCA map

The nonlinear chaotic algorithm map evolved in an effort to address the security concerns of one-dimensional logistic chaotic maps. This NCA map relies on nonlinear functions, time and space parameters, and continuous change in the encryption key [7]. In this work, the power function $(1-x)^{\beta}$ is applied to NCA maps. In addition, the tangent function is used in place of linear functions, originally proposed for the one-dimensional chaotic maps, therefore, the NCA is defined as

$$X_{N+1} = \lambda \,\mathrm{tg}(\alpha X_N)(1 - X_N)^{\beta}$$

where

$$X_N \in (0, 1), \quad n = 0, 1, 2, \ldots.$$

The values of the parameters λ, α and β are critical, therefore, it is important to list some important properties pertaining to their selection. These parameters

take positive values with the slope not less than 1 and $x_{N+1} > x_N$ when $X_N = 1/(1+\beta)$. The parameter λ can be defined as

$$\lambda = \mu \operatorname{ctg}\left(\frac{\alpha}{1+\beta}\right)\left(1+\frac{1}{\beta}\right)^{\beta},$$

and

$$\mu > 0.$$

The final version of the NCA map after the incorporation of $\mu = 1-\beta^{-4}$, which is obtained from experimental analysis, is given as

$$X_{N+1} = \left(1-\beta^{-4}\right)\operatorname{ctg}\left(\frac{\alpha}{1+\beta}\right)\left(1+\frac{1}{\beta}\right)^{\beta}\operatorname{tg}(\alpha X_N) \times (1-X_N)^{\beta}$$

where $x_n \in (0,1)$, $\alpha \in (0, 1.4]$, $\beta \in [5, 43]$, or $x_n \in (0,1)$, $\alpha \in (1.4, 1.5]$, $\beta \in [9, 38]$, or similarly these variable are also defined as $x_n \in (0,1), \alpha \in (1.5, 1.57]$, $\beta \in [3, 15]$. The range of α and β are determined by iterative experimental analysis.

2.3 Galois field exponent transformation

A primitive irreducible polynomial, $x^8 + x^4 + x^3 + x^2 + 1$ is selected from Table 2 and its elements from Galois field GF(2^8) in binary and exponent form are listed in Table 3. Multiple irreducible polynomials are available to generate the elements of GF(2^8). These polynomials are listed in Table 2.

2.4 The proposed image encryption system

In order to distort the image, the pixels are shuffled so that it becomes difficult to reconstruct or identify the original image after certain number of iterations. The Galois field GF(2^8) transformation is applied to image data so that the picture appears distorted. It is difficult to figure out the period of the Arnold cat map of a distorted image. The design of the key accommodates the information about the block size and seed of NCA for various parameters. The three subsystems of the proposed encryption system utilize the NCA map, Arnold cat map, and GF(2^8) exponent substitution. The NCA map incorporates the shuffle capability in subblocks of equal sizes. The subblocks can be of sizes 128, 64, 32, 16, 8, 4, and 2 pixels. The smaller subsizes incur more processing overhead and as a result, increase the complexity of the proposed algorithm. For example, for an image of dimensions 256×256 and subblock size of 16×16, the original image is split in to 256 (16×16) nonoverlapping linearly organized blocks. The subblocks are numbered from 1 to 256, so that in the next step of the algorithm, a sequence S_i of NCA maps is generated. These maps are scaled by a factor of 1000 and the resulting values are quantized in the integer interval of $[1, 256]$. The exponent values are substituted to distort the image and to enhance the confusion capability of the proposed algorithm (see Table 3). This process is similar to the *substitution* and introduces nonlinearity in the original data.

Table 2 Primitive irreducible polynomial of degree 8

Sr #	Primitive Irreducible Polynomial
1	$x^8+x^4+x^3+x^2+1$
2	$x^8+x^5+x^3+x^1+1$
3	$x^8+x^5+x^3+x^2+1$
4	$x^8+x^6+x^3+x^2+1$
5	$x^8+x^6+x^4+x^3+x^2+x+1$
6	$x^8+x^6+x^5+x+1$
7	$x^8+x^6+x^5+x^2+1$
8	$x^8+x^6+x^5+x^3+1$
9	$x^8+x^6+x^5+x^4+1$
10	$x^8+x^7+x^3+x^2+1$
11	$x^8+x^7+x^5+x^3+1$
12	$x^8+x^7+x^6+x+1$
13	$x^8+x^7+x^6+x^3+x^2+x+1$
14	$x^8+x^7+x^6+x^5+x^2+x+1$
15	$x^8+x^7+x^6+x^5+x^4+x^2+1$

2.5 Security analysis

In this work, we test the resistance of the proposed encryption method against statistical attacks. We present the results of correlation analysis, number of pixel change rate (NPCR) analysis, and unified averaged changed intensity (UACI) analysis. These analyses provide an insight into encryption capability of the proposed algorithm.

3 Correlation

In a plain image, the adjacent pixels show a high level of correlations. The encryption process organizes and substitutes data in order to increase randomness.

Table 3 The element of Galois field GF(2^8) in binary and exponent form with respect to primitive irreducible polynomial $x^8 + x^4 + x^3 + x^2 + 1$

Binary values	GF(2^8)	Binary values	GF(2^8)	Binary values	GF(2^8)	Binary values	GF(2^8)	Binary values	GF(2^8)
00000000	0	11100100	w^{33}	01110010	w^{34}	10101110	w^{21}	00111001	w^{35}
10000000	1	00010100	w^{53}	11110010	w^{136}	01101110	w^{121}	10111001	w^{32}
01000000	w	10010100	w^{147}	00001010	w^{54}	11101110	w^{43}	01111001	w^{137}
11000000	w^{25}	01010100	w^{142}	10001010	w^{208}	00011110	w^{78}	11111001	w^{46}
00100000	w^2	11010100	w^{218}	01001010	w^{148}	10011110	w^{212}	00000101	w^{55}
10100000	w^{50}	00110100	w^{240}	11001010	w^{206}	01011110	w^{229}	10000101	w^{63}
01100000	w^{26}	10110100	w^{18}	00101010	w^{143}	11011110	w^{172}	01000101	w^{209}
11100000	w^{198}	01110100	w^{130}	10101010	w^{150}	00111110	w^{115}	11000101	w^{91}
00010000	w^3	11110100	w^{69}	01101010	w^{219}	10111110	w^{243}	00100101	w^{149}
10010000	w^{223}	00001100	w^{29}	11101010	w^{189}	01111110	w^{167}	10100101	w^{188}
01010000	w^{51}	10001100	w^{181}	00011010	w^{241}	11111110	w^{87}	01100101	w^{207}
11010000	w^{238}	01001100	w^{194}	10011010	w^{210}	00000001	w^7	11100101	w^{205}
00110000	w^{27}	11001100	w^{125}	01011010	w^{19}	10000001	w^{112}	00010101	w^{144}
10110000	w^{104}	00101100	w^{106}	11011010	w^{92}	01000001	w^{192}	10010101	w^{135}
01110000	w^{199}	10101100	w^{39}	00111010	w^{131}	11000001	w^{247}	01010101	w^{151}
11110000	w^{75}	01101100	w^{249}	10111010	w^{56}	00100001	w^{140}	11010101	w^{178}
00001000	w^4	11101100	w^{185}	01111010	w^{70}	10100001	w^{128}	00110101	w^{220}
10001000	w^{100}	00011100	w^{201}	11111010	w^{64}	01100001	w^{99}	10110101	w^{252}
01001000	w^{224}	10011100	w^{154}	00000110	w^{30}	11100001	w^{13}	01110101	w^{190}
11001000	w^{14}	01011100	w^9	10000110	w^{66}	00010001	w^{103}	11110101	w^{97}
00101000	w^{52}	11011100	w^{120}	01000110	w^{182}	10010001	w^{74}	00001101	w^{242}
10101000	w^{141}	00111100	w^{77}	11000110	w^{163}	01010001	w^{222}	10001101	w^{86}
01101000	w^{239}	10 111100	w^{228}	00100110	w^{195}	11010001	w^{237}	01001101	w^{211}
11101000	w^{129}	01111100	w^{114}	10100110	w^{72}	00110001	w^{49}	11001101	w^{171}
00011000	w^{28}	11111100	w^{166}	01100110	w^{126}	10110001	w^{197}	00101101	w^{20}
10011000	w^{193}	00000010	w^6	11100110	w^{110}	01110001	w^{254}	10101101	w^{42}
01011000	w^{105}	10000010	w^{191}	00010110	w^{107}	11110001	w^{24}	01101101	w^{93}
11011000	w^{248}	01000010	w^{139}	10010110	w^{58}	00001001	w^{227}	11101101	w^{158}
00111000	w^{200}	11000010	w^{98}	01010110	w^{40}	10001001	w^{165}	00011101	w^{132}
10111000	w^8	00100010	w^{102}	11010110	w^{84}	01001001	w^{153}	10011101	w^{60}
01111000	w^{76}	10100010	w^{221}	00110110	w^{250}	11001001	w^{119}	01011101	w^{57}
11110000	w^{113}	01100010	w^{48}	10110110	w^{133}	00101001	w^{38}	11011101	w^{83}
00000100	w^5	11100010	w^{253}	01110110	w^{186}	10101001	w^{184}	00111101	w^{71}
10000100	w^{138}	00010010	w^{226}	11110110	w^{61}	01101001	w^{180}	10111101	w^{109}
01000100	w^{101}	10010010	w^{152}	00001110	w^{202}	11101001	w^{124}	01111101	w^{65}
11000100	w^{47}	01010010	w^{37}	10001110	w^{94}	00011001	w^{17}	11111101	w^{162}
00100100	w^{225}	11010010	w^{179}	01001110	w^{155}	10011001	w^{68}	00000011	w^{31}
10100100	w^{36}	00110010	w^{16}	11001110	w^{159}	01011001	w^{146}	10000011	w^{45}
01100100	w^{15}	10110010	w^{145}	00101110	w^{10}	11011001	w^{217}	01000011	w^{67}
11000011	w^{216}	01010011	w^{73}	00011011	w^{251}	10000111	w^{89}	01110111	w^{44}
00100011	w^{183}	11010011	w^{236}	10011011	w^{96}	01000111	w^{95}	11110111	w^{215}
10100011	w^{123}	00110011	w^{126}	01011011	w^{134}	11000111	w^{176}	00001111	w^{79}

Table 3 (*Continued*)

Binary values	GF(2^8)	Binary values	GF(2^8)	Binary values	GF(2^8)	Binary values	GF(2^8)	Binary values	GF(2^8)
01100011	w^{164}	10110011	w^{12}	11011011	w^{177}	00100111	w^{156}	10001111	w^{174}
11100011	w^{118}	01110011	w^{111}	00111011	w^{187}	10100111	w^{169}	01001111	w^{213}
00010011	w^{196}	11110011	w^{246}	10111011	w^{204}	01100111	w^{160}	11001111	w^{233}
10010011	w^{23}	00001011	w^{108}	01111011	w^{62}	11100111	w^{81}	00101111	w^{231}
10101011	w^{157}	10001011	w^{161}	11111011	w^{90}	00010111	w^{11}	10101111	w^{230}
11101011	w^{170}	11001011	w^{82}	00101011	w^{41}	01010111	w^{22}	11101111	w^{232}
00111111	w^{168}	01111111	w^{88}	01011111	w^{244}	11010111	w^{235}	00011111	w^{116}
10111111	w^{80}	11111111	w^{175}	11011111	w^{234}	00110111	w^{122}	10011111	w^{214}

Fig. 1 (**a**) Colored plain-image. (**b**) Cipher-image

With the data transformed into highly random format, the correlation of adjacent pixels decreases drastically. Therefore, a measure of correlation among pixels is a good indication in determining the resistance to statistical attacks [8]. The correlation coefficient for each pair of pixels is determined as

$$R_{AB} = \frac{\operatorname{cov}(A, B)}{\sqrt{D(A)}\sqrt{D(B)}}$$

where

$$D(A) = \frac{1}{N}\sum_{i=1}^{N}\big(A_i - E(A)\big)^2$$

and

$$\operatorname{cov}(A, B) = \frac{1}{N}\sum_{i=1}^{N}\big(A_i - E(A)\big)\big(B_i - E(B)\big)$$

the quantity $E(A)$ is determined as

$$E(A) = \frac{1}{N}\sum_{i=1}^{N} A_i.$$

Table 4 Correlation coefficients of Red, Green, Blue components of plain image and ciphered image

Correlation	Red component	Green component	Blue component
Plain image	0.817970	0.816331	0.729072
Ciphered image	−0.040889	−0.048110	−0.007104

The gray scale values of two adjacent pixels are represented by A and B. The image used in this work is shown in Fig. 1 and 3,000 pairs of adjacent pixel samples are randomly selected. Figures 1(a) and 1(b) show the original image and its encrypted version, respectively. It is evident from Table 4 that the correlation coefficient of the red, green, and blue component is drastically reduced after applying the proposed encryption algorithm; hence, the zero correlation property is approximately satisfied. In addition, the proposed algorithm reduces the correlation between red, green, and blue components.

Table 5 Same position correlations between Red, Green, Blue components

Correlation	Same position between red, green components	Same position between red, blue components	Same position between green, blue components
Plain image	0.735625	0.569469	0.713084
Ciphered image	−0.003803	−0.050968	0.021267

Table 6 Adjacent position correlations between Red, Green, Blue components

Correlation	Adjacent position between red, green components	Adjacent position between red, blue components	Adjacent position between green, blue components
Plain image	0.724585	0.560971	0.710684
Ciphered image	−0.021694	−0.00775	−0.045334

Table 7 Comparison of correlations between Red, Green, Blue components

Correlation	Adjacent position between red, green components	Adjacent position between red, blue components	Adjacent position between green, blue components
Proposed scheme	−0.021694	−0.00775	−0.076334
Rhouma's	0.248026	0.139054	0.171310
Sahar's	0.305352	0.204247	0.252515
Liu's	0.231220	0.125403	0.161153

In order to further enhance the correlation analysis, we test the same position correlation and relative adjacent position correlation, and apply it to all three layers of color.

The results of same position correlation analysis for red, green, and blue colors are shown in Table 5. This table shows that the correlation coefficient for the cipher image is very close to zero. In Table 6, the correlation analysis is performed for adjacent position analysis for all layers of colors. Once again, the results show that the encryption process substantially reduces the correlation coefficients.

A comparison of correlation coefficient among Rhouma's, Sahar's, and Liu's methods is presented in Table 7. The values of the results of the correlation analysis for the proposed algorithm depict better performance for all the three colors.

3.1 NPCR and UACI analysis

The number of pixel change rate analysis tests the behavior of all the pixels in an image in response to a change in one pixel in the original image. A value closer to 1 shows high sensitivity of the encryption system in a reaction to a single change in the input. The mathematical representation is presented as follows:

$$\mathrm{NPCR} = \frac{\sum_{l,m,n} D(l,m,n)}{P \times Q \times 3}.$$

In another test called UACI, the average intensity of difference between original image and encrypted image is evaluated. The higher values of UACI show more effectiveness of the encryption algorithm and resistance to differential attacks. The expression for this test is given as

$$\mathrm{UACI} = \frac{1}{P \times Q \times 3} \times \left(\sum_{l,m,n} \frac{|C_1(l,m,n) - C_2(l,m,n)|}{255} \right)$$

Table 8 NPCR and UACI of ciphered image with one bit different between the plain images

Image name	Analysis item	Layer	Image size 512 × 512		
			Proposed	Ref. [8]	Ref. [9]
White flowers	NPCR	Red	0.8758463	0.8360365	0.0003814
		Green	0.8758011	0.8359035	0.0003814
		Blue	0.8758070	0.8359488	0.0003814
	UACI	Red	0.3551028	0.3339424	0.0000580
		Green	0.3543820	0.3349684	0.0000534
		Blue	0.3542398	0.3338638	0.0000291

where C_1 and C_2 are cipher images resulting from two images that differ only by one byte. The $D(l, m, n)$ is of size $M * N * 3$ and is defined as

$$D(l,m,n) = \begin{cases} 1, & \text{if } C_1(l,m,n) = C_2(l,m,n) \\ 0, & \text{otherwise.} \end{cases}$$

The results of NPCR and UACI analyses are shown in Table 8. The proposed encryption algorithm shows relatively lower values of NPCR, while the UACI analysis yields comparable results to the benchmark algorithm. The performance of the proposed algorithms is better than the method presented in [8, 9].

4 Conclusion

In this work, a novel encryption method based on Galois field GF(2^8) transformation, Arnold cat map and NCA chaotic system is proposed. The behavior of this method is similar to the substitution box like encryption algorithms. The proposed algorithm is tested for its encryption strength by the use of statistical analysis. The results show that the performance of the proposed algorithms is comparable to other prevailing methods.

References

1. Wheeler, D.D., Matthews, R.A.J.: Super computer investigations of a chaotic encryption algorithm. Cryptologia **15**, 40–52 (1991)
2. Zhang, L.H., Liao, X.F., Wang, X.B.: An image encryption approach based on chaotic maps. Chaos Solitons Fractals **24**, 759–765 (2005)
3. Belmouhoub, I., Djemai, M., Barbot, J.P.: Cryptography by discrete-time hyperchaotic systems. In: Proceedings of 42nd IEEE Conference on Decision and Control, 9–12 Dec. 2003, vol. 2, pp. 1902–1907 (2003)
4. Scharinger, J.: Secure and fast encryption using chaotic Kolmogorov flows. In: Information Theory Workshop, 22–26 June 1998, pp. 124–125 (1998)
5. Chen, G., Mao, Y., Chui, C.K.: A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Solitons Fractals **21**, 749–761 (2004)
6. Zhou, Z., Yang, H., Zhu, Y., Pan, W., Zhang, Y.: A block encryption scheme based on 3D chaotic Arnold maps. In: Intelligent Interaction and Affective Computing (ASIA'09), International Asia Symposium (2009)
7. Hussain, I., Shah, T., Mahmood, H.: A New Algorithm to Construct Secure Keys for AES. Int. J. Cont. Math. Sci. **5**(26), 1263–1270 (2010)
8. Sobhy, M.I., Shehata, A.R.: Methods of attacking chaotic encryption and countermeasures. IEEE Trans. Acoust Speech Signal Process **2**, 1001–1004 (2001)
9. Guo, Q., Liu, Z.G., Liu, S.T.: Colour image encryption by using Arnold and discrete fractional random transforms in IHS space. Opt. Lasers Eng. **48**(12), 1174–1181 (2010)
10. Rhouma, R., Soumaya, M., Safya, B.: OCML-based colour image encryption. Chaos Solitons Fractals **40**(1), 309–318 (2009)
11. Sahar, M., Amir, M.E.: Colour image encryption based on coupled nonlinear chaotic map. Chaos Solitons Fractals **42**(3), 1745–1754 (2009)
12. Liu, H.J., Wang, X.Y.: Colour image encryption based on one-time keys and robust chaotic maps. Comput. Math. Appl. **59**(10), 3320–3327 (2010)
13. Hussain, I., Shah, T., Gondal, M.A., Mahmood, H.: A projective general linear group based algorithm for the construction of substitution box for block ciphers. Neural Comput. Appl. (2012). doi:10.1007/s00521-012-0870-0
14. Hussain, I., Shah, T., Gondal, M.A., Mahmood, H.: Analysis of S-box in image encryption using root mean square error method. Z. Naturforsch. A **67a**, 327–332 (2012)
15. Hussain, I., Shah, T., Gondal, M.A., Mahmood, H.: Generalized majority logic criterion to analyze the statistical strength of S-boxes. Z. Naturforsch. A **67a**, 282–288 (2012)
16. Hussain, I., Shah, T., Mahmood, H.: A group theoretic approach to construct cryptographically strong substitution boxes. Neural Comput. Appl. (2012). doi:10.1007/s00521-012-0914-5
17. Hussain, I., Shah, T., Gondal, M.A.: Image encryption algorithm based on PGL(2,GF(2^8)) S-boxes and TD-ERCS

chaotic sequence. Nonlinear Dyn. (2012). doi:10.1007/s11071-012-0440-0
18. Hussain, I., Shah, T.: S8 affine power affine S-boxes and their application. Neural Comput. Appl. (2012). doi:10.1007/s00521-012-1036-9
19. Hussain, I., Shah, T., Gondal, M.A., Mahmood, H.: Construction of S8 Lui J S-boxes and their application. Comput. Math. Appl. (2012). doi:10.1016/j.camwa.2012.05.017
20. Hussain, I., Shah, T., Gondal, M.A.: An efficient image encryption algorithm based on S8 S-box transformation and NCA map. Opt. Commun. (2012). doi:10.1016/j.optcom.2012.06.011
21. Hussain, I., Shah, T., Gondal, M.A., Wang, Y.: Analyses of SKIPJACK S-box. World Appl. Sci. J. **13**(11), 2385–2388 (2011)
22. Hussain, I., Shah, T., Gondal, M.A., Khan, W.A.: Construction of cryptographically strong 8×8 S-boxes. World Appl. Sci. J. **13**(11), 2389–2395 (2011)
23. Khan, M., Shah, T., Mahmood, H., Asif Gondal, M.: An efficient method for the construction of block cipher with multi-chaotic systems. Nonlinear Dyn. doi:10.1007/s11071-012-0675-9
24. Khan, M., Shah, T., Mahmood, H., Asif Gondal, M., Hussain, I.: A novel technique for the construction of strong s-boxes based on chaotic Lorenz systems. Nonlinear Dyn. **70**, 2303–2311 (2012). doi:10.1007/s11071-012-0621-x
25. Hussain, I., Shah, T., Gondal, M.A., Hassan, H.: Construction of new s-boxes over finite field and their application to watermarking. Z. Naturforsch. **67a**, 705–710 (2012). doi:10.5560/ZNA.2012-0090
26. Hussain, I., Shah, T., Mahmood, H., Afzal, M.: Comparative analysis of S-boxes based on graphical SAC. Int. J. Comput. Appl. **2**(5), 975–8887 (2010)
27. Hussain, I.: A novel approach of audio watermarking based on *S*-box transformation. Math. Comput. Model. **57**(3–4), 963–969 (2013)