



IECA: an efficient IoT friendly image encryption technique using programmable cellular automata

Satyabrata Roy¹ · Umashankar Rawat¹ · Harsh Ajay Sareen¹ · Sanjeet Kumar Nayak²

Received: 29 August 2019 / Accepted: 18 February 2020 / Published online: 28 February 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

Digital images play a vital role in multimedia communications in the modern era. With the advent of internet-of-things (IoT) applications, multimedia transfers are happening at a rapid pace. However, providing security to these data is essential if the data is sensitive. It is a challenging task in case of IoT applications due to the limitations of the sensors in terms of memory and computational efficiency. Therefore, conventional ciphers cannot be applied in the IoT devices. However, cellular automata (CA) can be used in this resource-constrained environment for providing security to the IoT devices, as it is inherently capable of creating complex patterns and pseudo-random sequences. It is also easy to implement in hardware. In this work, an IoT friendly programmable cellular automata (PCA) based block cipher called IECA is proposed. Further, randomness in the generated cipher-image has been tested using various statistical testings present in NIST test suite and DIEHARD test suite. The test results show that IECA generates high degree of randomness in the produced cipher-image. In addition to this, correlation, entropy and differential analysis of the proposed scheme justifies the robustness against different types of attacks. Experimental results show the efficiency of IECA as compared to the existing block ciphers.

Keywords Image cipher · IoT · Programmable cellular automata · Lightweight · Block cipher

1 Introduction

Presently the internet of things (IoT) is considered as one of the incredible technologies for mankind. Cloud-enabled IoT infrastructure plays a significant role in the modern digital era as far as data storage and communication are concerned (Singh et al. 2014). In this infrastructure, small devices collect data using sensors and send them to the cloud storage servers via the Internet. These data, so collected, are used for further processing and analysis. These sensors are helpful in

collecting data from physical locations where it is impossible for human beings to monitor physically. With the advent of multimedia technology, people are also transferring data in the form of image, video, and audio. The digital images provide a lot of information altogether very easily as compared to normal text data (Shaheen et al. 2019).

Though the IoT infrastructure provides several benefits to mankind, many security issues arise in data transfers, especially for transferring the sensitive images. These sensors collect data from the environment and transfer them through insecure public communication channels (Suri and Vijay 2019). Any adversary can access and manipulate the data during communication, resulting in various kinds of security breaches. In a typical IoT deployment scenario, sensors and actuators are placed at the perception layer, whereas the gateway devices with some computing capabilities are placed at the network layer. The users can interact with the cloud present in the application layer. There are many conventional ciphers available for providing security at the application layer or network layer, but in the perception layer or physical layer, due to resource constraints of the sensory devices, these conventional ciphers cannot be applied.

✉ Umashankar Rawat
umashankar.rawat@jaipur.manipal.edu

Satyabrata Roy
satyabrata.roy@jaipur.manipal.edu

Harsh Ajay Sareen
harhsareen97@gmail.com

Sanjeet Kumar Nayak
sanjeet.nayak@bennet.edu.in

¹ Department of Computer Science and Engineering, Manipal University Jaipur, Jaipur, India

² Department of Computer Science Engineering, Bennett University, Noida, India

Consequently, there arises a need for lightweight encryption mechanism in the perception layer (Li et al. 2015).

CA is an abstract structure that can generate chaotic sequences and has inherent properties of creating pseudo-random numbers (Beniani et al. 2018). CA can be easily implemented in hardware. In the recent past, CA has gained popularity among the researchers for developing cryptographic algorithms. The ciphers developed by CA is also lightweight in nature. In this paper, a PCA based image encryption scheme called IECA for IoT applications has been proposed, which is a block cipher. The CA used in the proposed scheme is a 2-state, 1-D with radius $r = 1$. The images are captured and encrypted at the perception layer and then they are sent to the network layer, where the decryption is performed. IECA has undergone the National Institute of Standard and Technology (NIST) statistical test suite and DIEHARD test suite. The test results prove that it is capable of generating random cipher-images. Further, to prove the robustness of IECA against different attacks and noise levels, standard tests like differential analysis, correlation coefficient analysis, histogram analysis, entropy analysis, and key sensitivity analysis have been performed. The experimental results show that the IECA outperforms the existing ciphers like Advanced Encryption Standard (AES), Data Encryption Standard (DES), 3DES and Chacha in terms of both total time for execution and randomness. In particular, the research paper has following contributions:

- An image encryption technique called IECA is proposed. It is lightweight in nature and is a symmetric-key cipher which is developed by using CA rule vectors for IoT applications. The design of the image encryption is performed by efficiently selecting the CA-rules which can show random behavior and are correlation immune. Such CA-rules are applied in a pixel-wise manner to obtain the cipher image.
- The proposed scheme has undergone many standard analysis like key sensitivity, histogram, correlation coefficient, information entropy, image quality, and differential analysis to check the robustness. In addition to the above analysis, the scheme has passed the tests present in NIST and DIEHARD test suits. Furthermore, resistance of IECA against different levels of noise has been tested.
- IECA has been implemented using Raspberry Pi and camera sensors and is compared with different existing ciphers. The evaluation result shows that the performance of IECA is better than the existing ones in terms of runtime, whereas the generated cipher-image contains high degree of randomness, making it suitable to be implemented in IoT applications.

The subsequent organization of this paper is as follows. Section 2 contains the related works previously done in this

field. Section 3 briefly contains some preliminary ideas that are essential for the readers to understand the proposed scheme, described in Sect. 4. Experimental results and performance analysis of the proposed scheme has been provided in Sect. 5. Finally, Sect. 6 concludes the work.

2 Related works

Wolfram has pioneered the application of CA in cryptography (Wolfram 1985, 1986a, 2002, 1986b). Prediction of next state of a CA operating under a certain CA-rule is very hard and the reversion of the status from some time-stamp t to $t - 1$ is almost impossible. Due to this invariant property of CA, it has wide applicability in cryptography. Since the inception of CA, it has been applied in creating pseudo-random sequence that plays a significant role in case of stream ciphers. A 1-D, 3-neighborhood CA is used to control the CA-rules and CA-states dynamically and then applying Evolutionary Multi Objective Optimization (EMOO), pseudo-random numbers (PRNs) are generated in (Guan and Zhang 2003).

CA along with linear feedback shift registers (LFSRs) are used in parallel to generate different random numbers. These random number generators are then combined using additive feedback and thus generates new PRNs (Hortensius et al. 1989; Tsalides et al. 1991). In Guan and Tan (2004), CA-rule is constructed as a factor of its neighbors' states to increase the behavioral complexities. Neighborhood consideration for state transition and rule selection have been flexibly adjusted to generate a new neighbor. This new neighbor is a function of two distinct mappings having two distinct temporal dependencies. In Neebel and Kime (1997), authors have proposed a weighted CA (WCA) based weighted PRN generation technique, where a large range of linear and non-linear rules have been used. In Sirakoulis (2012), the authors have introduced Hybrid Autonomous DNA Cellular Automata (HADCA), a molecular machinery synthesized from DNA, that is capable of generating high quality random numbers by executing multiple CA-rules in parallel.

In Bakhshandeh and Eslami (2013), the authors implemented an image encryption technique using CA, chaotic map and permutation-diffusion architecture. Here, the plain image is made to appear confusing by utilizing a piece-wise linear chaotic map. This achieves sufficiently high degree of confusion property. In diffusion phase, a reversible memory cellular automata is used to construct an efficient cipher. The proposed method is easy to implement, achieves highly secure diffusion and it is computationally efficient. Based on the same concept, CA is amalgamated with chaotic map to construct various structurally different cryptosystems (Beniani et al. 2018; Chai et al. 2018; Wuensche 2011).

Eslami et al. proposed a CA based image encryption technique that breaks the input plain image in blocks. Then a permutation is applied to implement confusion property in the blocks. Then, CA based technique is applied to change each pixel value for encryption. This technique has the capability of detecting any minor tampering in the cipher image. But, this technique requires lot of complex computations for the permutations and then it needs memory for the CA technique. Eslami and Kabirirad (2019) These make the technique inappropriate for application in resource constrained IoT applications. In Babaei et al. (2020), the authors have proposed an image encryption technique using CA and DNA sequence, which demands a lot of computing and memory making the approach not suitable for IoT applications. CA is used along with DNA sequence also in Enayatifar et al. (2019). This is an indexed based technique that demands heavy computation and because of using DNA sequence, it is not suitable for application where resource constrained sensory devices are deployed.

In Anghelescu (2012), hardware implementation of Programmable Cellular Automata (PCA) based cryptosystem has been proposed. This is a symmetric cryptosystem that uses a single secret key for encryption and decryption. PCA state transitions have defined the block ciphering scheme, which is easy to implement, provides good amount of security and is suited ideally for the FPGA devices. In recent years CA is also used for generating random numbers using optimized FPGA (Petrica 2018). Here the authors have explored characteristics of self-programmable cellular automaton (SPCA) based pseudo random number generators (PRNGs). A set of parameter constraints has been derived to produce random bits per clock cycle using splittable Look-Up tables (LUTs).

The pseudo random number generators that were implemented using CA-rules or WCA rules have complex operational structures. As a consequence, they cannot be implemented in memory constrained devices. The implementation complexity of FPGA based structures is also high. The CA based ciphers that use evolutionary computation or HADCA techniques cannot be implemented in memory constrained camera sensors for generating image ciphers. These ciphers can be very easily implemented in cloud servers where there is ample amount of resources available. But, cloud applications can be set up once the data is received from the sensor devices. Hence, the communication between perception layer and network layer of a typical IoT deployment architecture remains vulnerable. The proposed scheme does address the problems of designing a lightweight CA based image cipher which can be efficiently implemented for providing security in the IoT applications.

3 Preliminaries

In this section fundamental concepts about CA required for better understanding of the proposed work has been discussed. A detailed discussion about CA-rule generation and thereafter the formation of group cellular automata is also described. Further, the mathematical representations of some of the useful rules are shown in this section.

3.1 Cellular automata

von Neumann (1951), Neumann et al. (1966) is considered as the father of CA. Having worked on the self replication theory, he was trying to come up with a system that would generate an exact replica of itself. Though the 'prima facie' of biology is the realm of continuous dynamics and fluid systems, inspired by a suggestion from his colleague, Stanislaw Ulam (Beyer et al. 1985; Ulam 1952) at Los Alamos National Laboratory in 1940, Neumann shifted his focus to discrete and 2-D systems. After application of 29 different states and complex dynamics, he got success in self-reproduction and thus proposed the concept of CA. This is also the first discrete model with the capability of parallel computation. It was formally proved to be considered as a universal computer, i.e. it can emulate the behaviour of a universal Turing machine by computing all recursive functions (Toffoli and Margolus 1990).

CA are abstract and discrete computational systems used as models of complexity. CA are also used to specifically represent non-linear dynamics in various scientific domains. CA have temporal and spatial discrete features, i.e., they are made of a finite, denumerable set of uniform and simple units, called cells. At each timestamp, each cell can be instantiated to one of the finite states. They can evolve at discrete time-steps in parallel by a transition function or dynamic transition rule or simply, CA-rule. This CA-rule depends on the states of its neighbors and of itself, depending on various radius considerations (Nandi et al. 1994). If radius is 1, the neighbors are the left and right cell excluding the particular cell under consideration.

CA are abstract Roy et al. (2019), i.e., it can be represented through purely mathematical terms and implementation can be performed on physical structures. Apart from this, CA are computational systems capable of computing and solving algorithms and even it can compute anything that is computable according to Church–Turing thesis (Bernays 1936). The main characteristic of CA lies in displaying complex progressive behaviour that starts from simple cells obeying simple CA-rule. As a consequence, CA attracted a huge number of researchers from cognitive and cryptography domain. They were interested

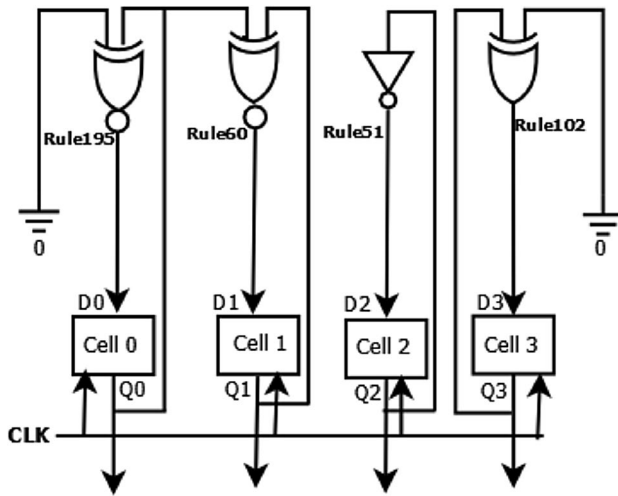


Fig. 1 A Hybrid CA with null boundary

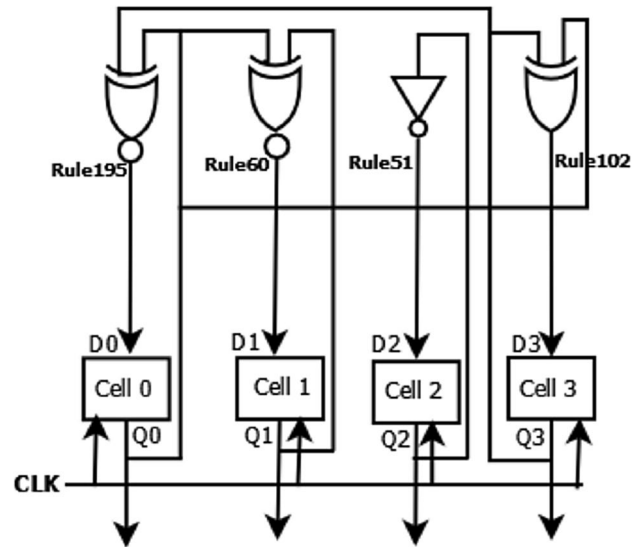


Fig. 2 A hybrid CA with periodic boundary

in studying pattern formation and complexity in an intrinsically abstract setting.

The cells of a CA progress by a local function, called CA-rule that controls the future cell-values or states based on the states of its neighbors. Mathematically, the state of a cell at time $t + 1$ is a function of the states of its neighbors at time t as shown in Eq. (1).

$$V_x^{t+1} = f_x(V_{x-1}^t, V_x^t, V_{x+1}^t) \tag{1}$$

An affine CA, under field G and $\forall V_i \in G$, can be defined by Eq. (2).

$$f_x(V_{x-1}^t, V_x^t, V_{x+1}^t) = c_{x-1} \cdot V_{x-1}^t + c_x \cdot V_x^t + c_{x+1} \cdot V_{x+1}^t + k_x \tag{2}$$

Here, $c_{x-1}, c_x, c_{x+1}, k_x \in G$, \cdot represents multiplication operation and $+$ represents the addition operation over G .

A 1-D CA of size n is called a non-circle CA, if all the cells are connected as a straight line. In a non-circled CA, if the left and right neighbors of the leftmost and rightmost cells are zero, it is called a null-boundary CA. On the other hand, if the rightmost and leftmost cells are adjusted to be neighbors of one another, it is called a CA with periodic boundary. When all the applied f_x 's are same $\forall x = 1, 2, \dots, n$, it becomes a uniform CA, otherwise it is a hybrid CA. Figures 1 and 2 show an example of CA having null boundary and periodic boundary conditions, respectively. In Fig. 1, left neighbor of cell-0 and right neighbor of cell-3 are set as 0. Hence, it is a null boundary CA. In Fig. 2, left neighbor of cell-0 is cell-3, whereas right neighbor of cell-3 is cell-0. Hence, it is a periodic boundary CA. Since same CA-rule is not applied on each cell, they are hybrid CA.

A one dimensional CA with two states and three neighbors can generate 2^3 CA-rules. Each cell of a CA changes its state depending on the CA-rule applied in it. For example, CA-rule 150 can be formed and expressed as described below:

111	110	101	100	011	010	001	000
1	0	0	1	0	1	1	0

Now, the 1's can be placed at the appropriate cells of a 3-variable K-map to form Eq. (3) that represents CA-rule 150.

$$Rule\ 150 : V_x^{t+1} = (V_{x-1}^t \oplus V_x^t \oplus V_{x+1}^t) \tag{3}$$

It can be observed from the expression that the state of x th cell at time-stamp $t + 1$ depends on the states of itself, left and right neighbors at time-stamp t . Likewise, the logic expressions for some of the CA-rules used in this work, can be represented through Eqs. (4) to (16).

$$Rule\ 18 : V_x^{t+1} = (\overline{V_{x-1}^t} \overline{V_x^t} V_{x+1}^t) = \overline{V_x^t} (V_{x-1}^t \oplus V_{x+1}^t) \tag{4}$$

$$Rule\ 22 : V_x^{t+1} = V_{x-1}^t \overline{V_x^t} \overline{V_{x+1}^t} + \overline{V_{x-1}^t} V_x^t \overline{V_{x+1}^t} + \overline{V_{x-1}^t} \overline{V_x^t} V_{x+1}^t \tag{5}$$

$$Rule\ 30 : V_x^{t+1} = V_{x-1}^t \overline{V_x^t} \overline{V_{x+1}^t} + \overline{V_{x-1}^t} V_x^t + \overline{V_{x-1}^t} V_{x+1}^t \tag{6}$$

$$Rule\ 45 : V_x^{t+1} = V_{x-1}^t \overline{V_x^t} V_{x+1}^t + \overline{V_{x-1}^t} \overline{V_{x+1}^t} + \overline{V_{x-1}^t} V_x^t \tag{7}$$

$$\text{Rule 51 : } V_x^{t+1} = \overline{V_x^t} \quad (8)$$

$$\text{Rule 54 : } V_x^{t+1} = \overline{V_{x-1}^t} V_x^t \overline{V_{x+1}^t} + V_{x-1}^t \overline{V_x^t} + \overline{V_x^t} V_{x+1}^t \quad (9)$$

$$\text{Rule 60 : } V_x^{t+1} = (V_{x-1}^t \oplus V_x^t) \quad (10)$$

$$\text{Rule 90 : } V_x^{t+1} = V_{x-1}^t \oplus V_{x+1}^t \quad (11)$$

$$\text{Rule 102 : } V_x^{t+1} = (V_x^t \oplus V_{x+1}^t) \quad (12)$$

$$\text{Rule 105 : } V_x^{t+1} = \overline{V_{x-1}^t \oplus V_x^t \oplus V_{x+1}^t} \quad (13)$$

$$\text{Rule 153 : } V_x^{t+1} = \overline{(V_x^t \oplus V_{x+1}^t)} \quad (14)$$

$$\text{Rule 161 : } V_x^{t+1} = \overline{V_{x-1}^t} \overline{V_x^t} \overline{V_{x+1}^t} + V_{x-1}^t V_x^t V_{x+1}^t \quad (15)$$

$$\text{Rule 195 : } V_x^{t+1} = \overline{(V_{x-1}^t \oplus V_x^t)} \quad (16)$$

In general, a CA-rule can be represented using Eq. (17).

$$\text{CA - Rule} = \sum_{j=0}^{n-1} V_p \cdot x^p \quad (17)$$

Here, V_p represents the state of the CA (either 0 or 1 in this case), x is the cell under consideration and p is a positive integer ranging from 0 to $n-1$ ($n = 8$). Now, if a CA is expressed by Exclusive-OR and/or Exclusive-NOR logical expressions, it is called additive CA (Fig. 3). Additionally it is worth noting that if an additive CA uses Exclusive-OR property, it is non-complemented CA, otherwise it becomes a complemented CA. For example, CA represented by Eqs. (3), (10), (11) and (12) are non-complemented ones, while Eqs. (13), (14) and (16) represent complemented CA. A sample complemented and non-complemented PCA structure is shown in Figs. 4 and 5 respectively. When many heterogeneous CA-rules are applied on a hybrid CA, the set of applied rules is called a CA Rule Vector (CARV). Every CA has its own characteristic matrix C and a corresponding characteristic polynomial Nandi et al. (1994). A cellular automata is called a Group Cellular Automata (GCA) if $C^d = I$, where d is a positive integer and I is identity matrix. GCA maintains idempotency under complement operation, i.e., the complement operation on a GCA produces again a GCA. State transition of a CA can be expressed in terms of its characteristic matrix by using Eq. (18).

$$V_x^{t+1} = [C][V_x^t] \quad (18)$$

CA cells transit from one state to another by a CA-rule and this can be represented through time-space diagrams. If a CA-rule, R_1 maps a certain configuration to a state, a_k and another CA-rule R_2 maps the same to state $a_k + 1$, the rules R_1 and R_2 are called next to each other. The Hamming distance between R_1 and R_2 is 1. With this consideration of the term, 'distance', we say that all the CA-rules stay in a space, called *Rule Space*. In rule space, each point represents a rule table or CA-rule. All the points are arranged in such a way that the Hamming distance of all the nearby points is equal to 1.

There is a quantifying measure for deciding the probability that two such rules will have similar behavior. Different regions of the rule space consist of rules with different behaviors. The behaviors can be regular, complex or random Li et al. (1990). An activity parameter, λ can be defined in case of CA having binary states, as the density of 1's in the rule table. For example, value of λ for CA-rule (10011101) is $5/8$. In the rule space, one subset can be moved to another by suitable adjustment of this λ . Figure 3 shows a schematic illustration of CA-rule space on the basis of λ . As the value of λ consistently increases from 0 to 1, the prominent behavior of the rules varies from homogeneous fixed point to in-homogeneous fixed point, periodic and chaotic. When λ is greater than 0.5 the same is reversed. The reason can be justified by toggling the 0's and 1's.

The CA-rules can be classified into five classes (Li and Packard 1990): (A) null rules, (B) fixed-point rules, (C) periodic rules, (D) locally chaotic rules and (E) global chaotic rules. This means the characterization of rules by the dynamics from a particular random initial configuration. This classification is not based on the dynamics from all possible initial configurations as described in Culik II and Yu (1988). This is also not based on mathematical characterization of CA-rules (Aizawa and Nishikawa 1986; Gutowitz 1989). This is more of the phenotype, rather than genotype that is used in classification. However, Wolfram classification Wolfram (1984) has four classes (I, II, III and IV). The relationship between these classifications is as follows: Wolfram's Class I is same as Class A; Wolfram's Class II consists of Class B and C; Wolfram's Class III rules are same as Class D. Wolfram's Class IV rules cannot be included in any of the categories defined by Li et. al. (1990). In this work, we have used Class C and D rules along with some Class B rules. Table 1 shows the detailed classification of the elementary binary state CA-rules.

3.2 Programmable cellular automata (PCA)

The logical equations of CA-rules imply that these rules mostly differ only in one or two positions; for some other rules, it may be more. So, this leads to the idea of applying different rules at different times at different cells. So, a n -cell

Table 1 Classification of elementary CA-rules into five classes Li and Packard (1990)

Rule number	Class ID
0, 8, 32, 40, 128, 136, 160, 168	Class A: null rules
2, 4, 10, 12, 13, 24, 34, 36, 42, 44, 46, 56, 57, 58, 72, 76, 77, 78, 104, 130, 132, 138, 140, 152, 162, 164, 170, 172, 184, 200, 204, 232, 1, 3, 5, 6, 7, 9, 11, 14, 15, 19, 23, 25, 27, 28, 29, 33, 35, 37, 38, 41, 43, 50, 51, 74, 108, 131, 133, 134, 142, 156, 178	Class B: fixed-point rules
26, 73, 154	Class C: periodic rules
18, 22, 30, 45, 54, 60, 90, 105, 106, 129, 137, 146, 150, 161	Class D: locally chaotic rules
	Class E: chaotic rules

cellular automata can be fully utilized to accommodate 2^m different compositions. Each of these compositions can be realized in hardware through various switches, control lines and a ROM having pre-loaded control program. This structure is called Programmable Cellular Automata (PCA). Now, if the ROM is replaced by EEPROM, a huge flexibility can be allowed by changing the control signals periodically or according to requirements.

4 IECA - The Proposed Scheme

In a typical Cloud enabled IoT application framework as shown in Fig. 6, the camera sensors capture the sensitive images at perception layer or physical layer and later on,

Fig. 3 Schematic representation of structure of CA-rule space

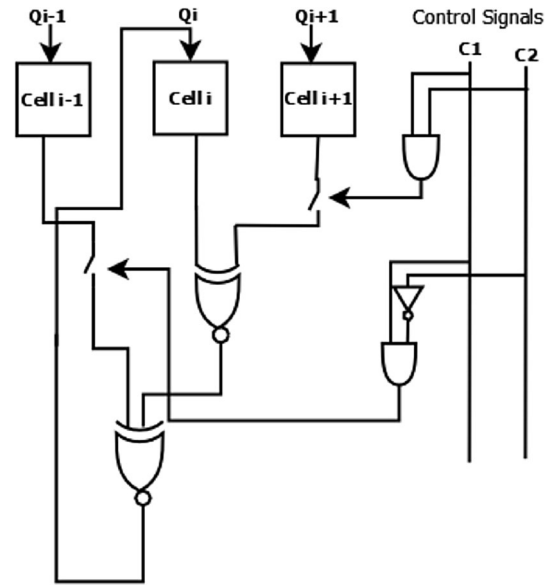
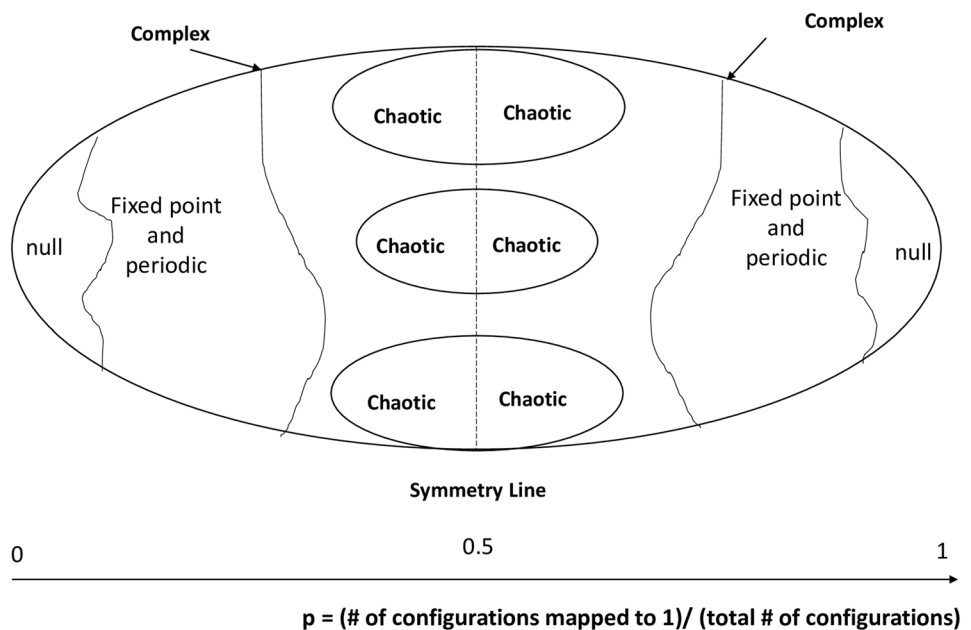


Fig. 4 A complemented PCA structure

those images are sent to the fog nodes deployed at the network layer or middle layer for further processing. The fog nodes contain devices with comparatively more computation capabilities and storage than that of sensors. Purpose of deploying fog nodes at the network layer is to minimize the network traffic and latency for providing quick response to achieve better quality of service (QoS).

The fog nodes then sends the images to the cloud at the application layer as and when required. Since, the images captured by the camera sensors are sensitive ones, they are encrypted at the perception layer and are sent to the network layer, where they are decrypted. This scheme restricts

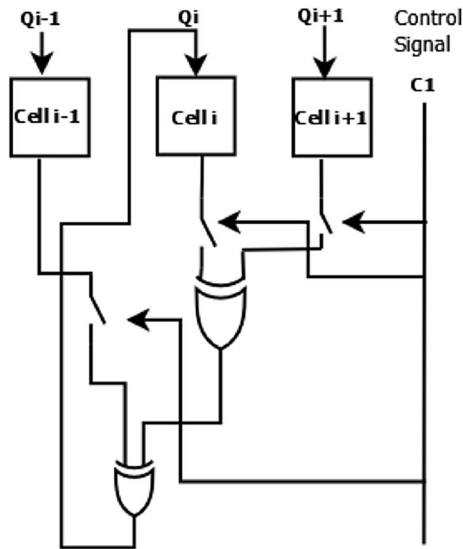


Fig. 5 A non-complemented PCA structure

the adversary from getting plain images by attacking the insecure public communication channel established between perception layer and network layer.

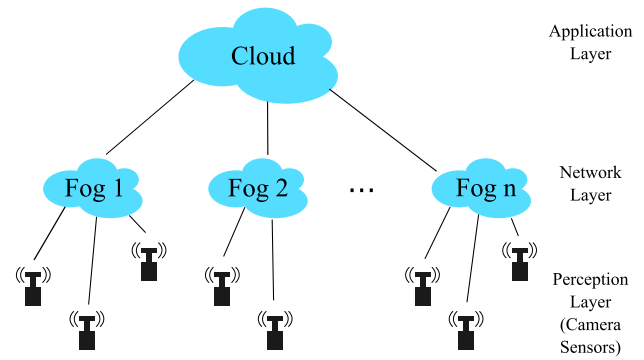


Fig. 6 Architectural model

Image is a collection of pixel values of heterogeneous types. The primary focus of this work is to alter these pixel values efficiently using group cellular automata (GCA) rule vectors. As a result, security at the perception layer is achieved. As, conventional encryption algorithms require more memory and computation power, they cannot be used to encrypt the sensitive images by the camera sensors having limitations in terms of memory, computation power, bandwidth, and battery capacity. On the other hand, GCA is more preferable in this situation due to its lightweight nature

Algorithm 1 Rule-vector Generator

Input: Pool of 256 CA-rules, 8-bit IV.

Output: CARVList8, consisting of 8 length CA Rule-vector, CAR8.

```

1: for  $i \leftarrow 1$  to 8 do ▷ Initialize PCA cells
2:    $PCA[i] \leftarrow i \bmod 2$ 
3:    $temp[i] \leftarrow PCA[i]$ 
4: end for
5: for  $p \leftarrow 1$  to number of all possible CART8 do
6:   for  $i \leftarrow 1$  to 8 do
7:      $tempList[i] \leftarrow \text{randomSelect}(\text{Pool of CA-Rule no. 1 to 256})$ 
8:   end for
9:   for  $j \leftarrow 1$  to 8 do
10:    for  $k \leftarrow 1$  to 8 do
11:      Apply  $tempList[k]$  to  $PCA[k]$ 
12:    end for
13:   end for
14:   for  $u \leftarrow 1$  to 8 do
15:     if  $(PCA[u] = temp[u])$  then
16:       Add CAR8 present in tempList to CARVList8
17:     else
18:       Check other CAR8s
19:     end if
20:   end for
21: end for

```

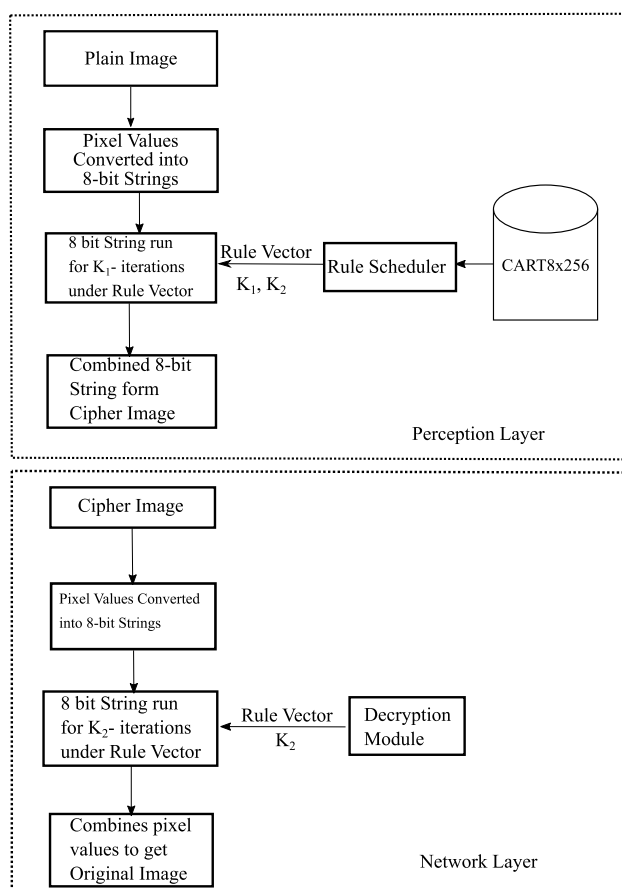


Fig. 7 Proposed scheme

and capability of generating random and complex patterns. Therefore, we motivated towards the concept of using GCA for image encryption in this particular Cloud-enabled IoT framework.

In the proposed scheme, a 1-D CA structure has been considered with null-boundary condition as discussed in Sect. 3. In this scheme a hybrid PCA having GCA characteristics Nandi et al. (1994) is used for implementing a symmetric key based image encryption cipher. Figures 1 and 2 show a typical hybrid null-boundary CA and a hybrid periodic boundary CA respectively.

Algorithm 2 Rule-table Generator

Input: CARVList8

Output: CART8x256

- 1: Call Rule-vector Generator to get CARVList8
- 2: **for** $i \leftarrow 1$ to 8 **do**
- 3: ST \leftarrow toBinaryString (i) if length 3
- 4: **for** $i \leftarrow 1$ to 256 **do**
- 5: RT[i][j] \leftarrow Apply Rule j to ST
- 6: **end for**
- 7: **end for**

A schematic representation of the proposed scheme is shown in Fig. 7. A 1-D 3-neighborhood CA can generate

2^{23} ($= 256$) rules, out of which only those rule vectors are chosen that satisfies GCA property, i.e. they can re-generate the initial configuration after a certain number of iterations. Here, the number of iterations used is 8. For these GCA, $d = 8$, where $C^d = I$.

Algorithm 3 Encryption

Input: Gray_Scale_Image, CART8x256

Output: Cipher-Image, K_1 , K_2

- 1: $iterations \leftarrow$ randomInteger (1 to 8) + 1
- 2: $K_1 \leftarrow iterations - 1$
- 3: $K_2 \leftarrow 8 - K_1$
- 4: $SCARV8 \leftarrow$ randomRuleVector (CART8x256)
- 5: $random \leftarrow$ randomInteger (0 to 255)
- 6: **for** $j \leftarrow 1$ to H **do**
- 7: **for** $k \leftarrow 1$ to W **do**
- 8: IMG \leftarrow (pixel[j][k] \times $random$) % 256
- 9: IM \leftarrow toBinaryString (IMG) of length 8
- 10: **for** $x \leftarrow 1$ to $iterations$ **do**
- 11: STR \leftarrow "00000000"
- 12: IM \leftarrow "0" + IM + "0"
- 13: **for** $i \leftarrow 1$ to 8 **do**
- 14: row \leftarrow toInteger (IM[i] to IM[$i+3$])
- 15: rule \leftarrow RV[i]
- 16: STR[i] \leftarrow RT[row][rule]
- 17: **end for**
- 18: IM \leftarrow STR
- 19: **end for**
- 20: pixel[j][k] \leftarrow toInt (STR)
- 21: **end for**
- 22: **end for**

Algorithm 1 is used to generate CARV8, all possible GCA rule vectors. These rule vectors are stored in an array named as CARVList8. This algorithm takes as input 256 CA-Rules (CAR8) to be applied on the null boundary PCA cells. The PCA cells are initialized by an arbitrary initial vector (IV) of size 8. After that from CARVList8, a random CAR8 is chosen and is applied on each PCA cell. So, all the eight CAR8 altogether form a CARV8. Now, if this CARV8 can re-generate the IV after 8 iterations, it is stored in CARVList8, otherwise it is rejected. Algorithm 2 maps the CARVList8 to a rule table, named as CART8x256, having 8 rows and 256 columns. This data structure helps in reducing the execution time by decreasing the CARV8 access time and thus contributes greatly in increasing battery-life. Through this CART8x256, the algorithm has satisfied one of the necessary criteria for being termed as lightweight. This CART8x256 data structure is used for both image encryption and decryption.

Algorithm 4 Decryption

Input: Cipher_Image, SCARV8, CART8x256, K_2
Output: Plain_Image

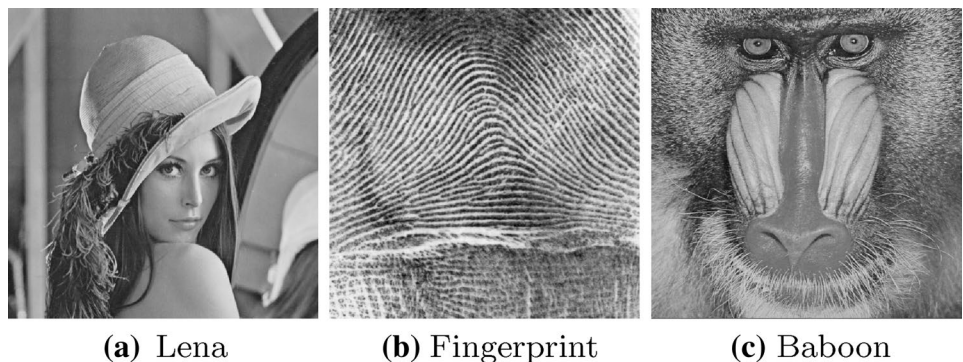
```

1:  $de\_iterations \leftarrow K_2 + 1$ 
2:  $inv\_random \leftarrow random^{-1} \% 256$ 
3: for  $j \leftarrow 1$  to  $H$  do
4:   for  $k \leftarrow 1$  to  $W$  do
5:      $IMG \leftarrow pixel[j][k]$ 
6:      $IM \leftarrow toBinaryString(IMG)$  of length 8
7:     for  $x \leftarrow 1$  to  $de\_iterations$  do
8:        $STR \leftarrow "00000000"$ 
9:        $IM \leftarrow "0" + IM + "0"$ 
10:      for  $i \leftarrow 1$  to 8 do
11:         $row \leftarrow toInteger(IM[i] \text{ to } IM[i+3])$ 
12:         $rule \leftarrow RV[i]$ 
13:         $STR[i] \leftarrow RT[row][rule]$ 
14:      end for
15:       $IM \leftarrow STR$ 
16:    end for
17:     $pixel[j][k] \leftarrow (toInt(STR) \times inv\_random) \% 256$ 
18:  end for
19: end for

```

Algorithm 3 encrypts the gray scale plain image captured by the camera sensors deployed at the perception layer. It takes the image and CART8x256 as inputs. It randomly chooses one CARV8 from CART8x256 and applies it to the plain-image in a pixel-wise fashion. Further, it randomly selects K_1 , the number of iterations to be executed on the pixel and generates the cipher image. Since, the cycle length of CARV8 is 8, it is required to send $K_2 (= 8 - K_1)$ to the network layer along with the cipher-image. Here, $(K_2, CARV8)$ is the secret key which is sent to the fog node at the network layer through a secure channel. The communication of the secret key falls beyond the scope of this work. So, the receiver at the network layer gets the triple $\langle \text{Cipher-Image}, K_2, \text{CART8x256} \rangle$ as input. The fog node then executes Algorithm 4 to decrypt and get back the plain image. This algorithm is used for getting back the plain image. This image encryption scheme is a loss-less image encryption scheme as shown in the subsequent section. Algorithm 3 and 4 have time complexity $\mathcal{O}(H*W*I*L)$, where H and W signifies the height and width of the image, respectively. I is the number of times (*iterations*) the rule vector is to be applied on the binary strings, and L is the length of rule vector.

Fig. 8 Plain images used in the proposed scheme



Since, in this work CA rule vector of length 8 is used, L is constant ($=8$) and the order, d of GCA is 8. Order of GCA is the number of iterations after which it regenerates the initial configuration. Hence, the time complexity of Encryption and Decryption algorithms is $\mathcal{O}(H*W)$.

5 Experimental results and analysis

The proposed scheme has been simulated on Intel (R) Core (TM) i5-3230M 2.60 GHz CPU, 4 GB RAM, WINDOWS 8.1 pro operating system. The comparison with existing algorithms such as AES, DES, 3DES and Chacha is done in Raspberry Pi 3, using the Python programming language. The Raspberry Pi used for the experiments has a 1.2 GHz, 64-bit quad core ARMv8 CPU with RAM size of 1 GB. The plain images used in this work are Lena, Fingerprint and Baboon as presented in Fig. 8a–c. All these gray scale images have the essential features e.g. the images possess all the middle grays in sufficient amount with zero blacks and zero whites, nice textures and flat regions, etc. required for any image processing algorithm. Encryption of these images are shown in Fig. 9a–c respectively. Similarly, decrypted images are shown in Fig. 10a–c respectively. It can be clearly noted that the cipher-images do not reveal any idea of the plain images.

5.1 Security analysis

In this subsection, a detailed analysis of the proposed scheme is presented to confirm the robustness of the scheme against different attacks. The scheme is also compared with various existing similar type of ciphers like AES, DES, 3DES and Chacha. The results of the analysis show the feasibility of implementation of IECA in real time IoT applications where camera sensors are deployed.

Fig. 9 Encryption of several images using the proposed scheme

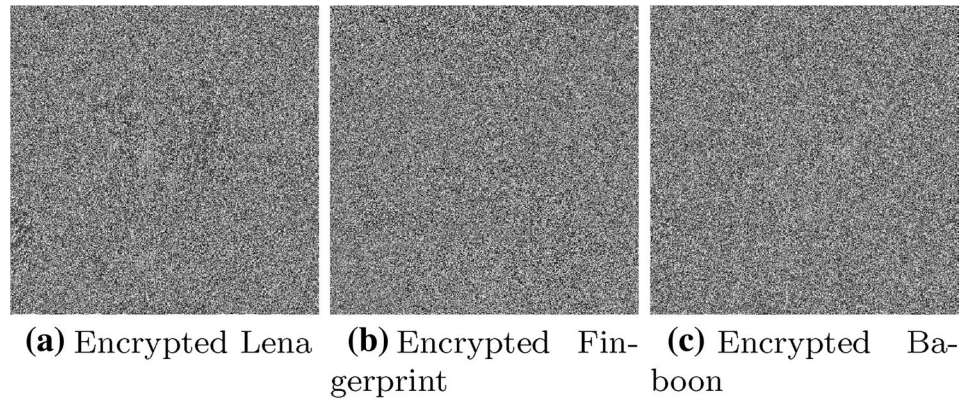


Fig. 10 Decryption of several images using the proposed scheme

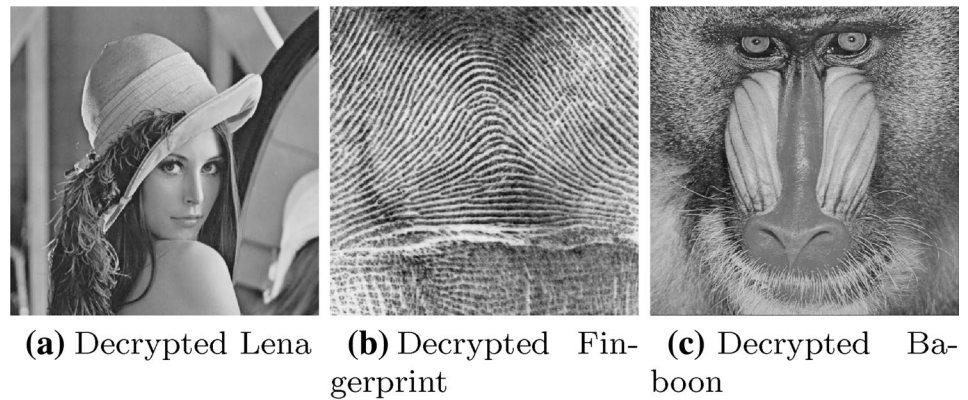
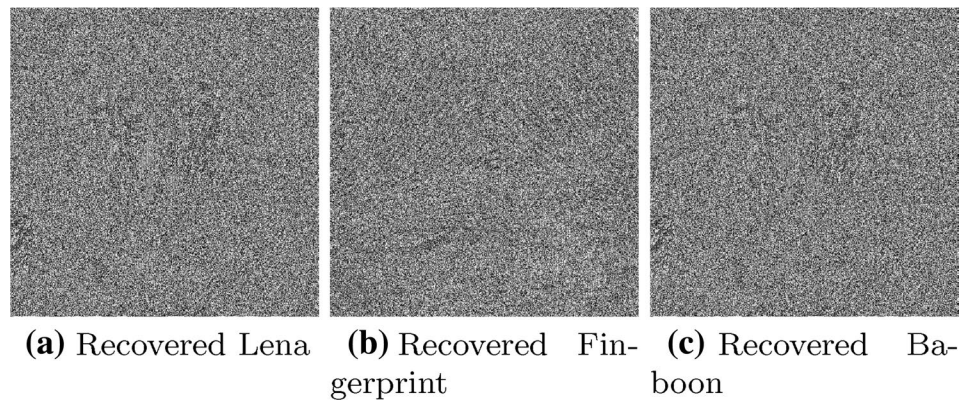


Fig. 11 Decryption of several images with a tiny change in CARV8



5.1.1 Key Sensitivity Analysis

The secret key for the proposed scheme is $\langle K_2, CARV8 \rangle$. Even if there is a tiny change in even one bit of the selected CARV8, the receiver will not be able to retrieve back the original image received from the perception layer. The results in Fig. 11a–c show the decrypted image of Lena, Fingerprint and Baboon respectively, after changing one bit in the CARV8 where only one bit position of the rule vector is changed.

5.1.2 Analysis of key space

The key space with reference to an encryption scheme signifies the total number of possible keys (Li et al. 2015; Bisht et al. 2019). A larger key space of a cipher signifies larger resistance of the cipher against brute-force attack (Gupta et al. 2019; Kaur and Kumar 2018e). The block-size used in this scheme is 8 bits. Then, the number of iteration is randomly chosen and rule vector of length 8 is also randomly chosen out of 256 available rules. The block size is

the total pixel values of a row of the gray scale image at one go. 1 out of 256 possible CA-rules can be applied on each pixel value. Suppose, the dimension of a gray scale image is 512×512 ; so, the key space would be $8 \times 2^{64 \times 512} = 2^{32768 \times 3}$. Hence, brute force attack is very difficult with the computation power of modern computers.

5.1.3 Resilience against cryptanalysis attacks

The symmetric secret key of the proposed scheme is generated by considering a 1-D, 3-neighborhood hybrid CA rule vector. Out of all the possible rules, GCA rules with cycle length = 8 have been chosen. These rules are highly dynamic and are dependent on the neighbors at different iterations at different timestamps. Besides, K_1 and K_2 are chosen randomly at runtime. Hence, it is always varying in spite of considering same initial vector or pixel values. The same image can generate several cipher image depending on this condition. Due to this high randomness is produced. As a consequence, the attacker will not be able to get any fruitful outcome even if she feeds the image with all zero values to the encryption system. Thus, known plaintext, chosen plaintext, chosen ciphertext and known ciphertext attacks do not affect the proposed scheme.

5.1.4 Analysis of histogram

Histogram is a crucial aspect of a digital image. It is the graphical representation of the tonal distribution of an image (Kaur and Kumar 2018c, d). It also represents the distribution of the number of pixel values for every tone-value (Kaur and Kumar 2018a; Kaur et al. 2019). The flatter and uniform distribution of the histogram signifies higher randomness in generated cipher-image. Figure 12a–c show the histogram plots of different plain images. Similarly, the histogram plots of corresponding encrypted images are shown in Fig. 12d–f. It can be clearly observed from the histogram of the cipher images that the distribution is quite uniform in nature. It sufficiently justifies the security of the proposed scheme.

5.1.5 Correlation coefficient analysis

Correlation coefficient ρ , is used to measure the relationship of two adjacent pixels of an image (Kaur and Kumar 2018f, 2020). If both the pixels belong to an image, then the correlation coefficient will be very high or near to 1. If the correlation between two pixels of an image is very less or near to 0, it shows that the image is a random image having no meaningful content or fact for understanding (Kaur and Kumar 2018b; Gupta et al. 2019). The correlation coefficient ρ is computed through Eq. (19).

$$\rho = \frac{\sum_m \sum_n (G_{mn} - \bar{G})(H_{mn} - \bar{H})}{\sqrt{(\sum_m \sum_n (G_{mn} - \bar{G})^2)(\sum_m \sum_n (H_{mn} - \bar{H})^2)}} \quad (19)$$

Here, \bar{G} and \bar{H} represent the mean of all pixel values of the image. G_{mn} signifies the pixel value at m^{th} row and n^{th} column. Here, we have performed horizontal, vertical and diagonal correlation coefficient analysis. The detailed values in each case for both plain image and cipher image is shown in Table 2. Figures 13, 14 and 15 show the correlation coefficient analysis plots of Lena, Fingerprint and Baboon images, respectively. It can be noted that the plots of cipher-images contain ample amount of random bits and thus justify the strength of the proposed scheme.

5.1.6 Information entropy analysis

One of the important measure of randomness in cipher image is information entropy (Wu et al. 2017). Information entropy, ϵ can be measured using Eq. (20).

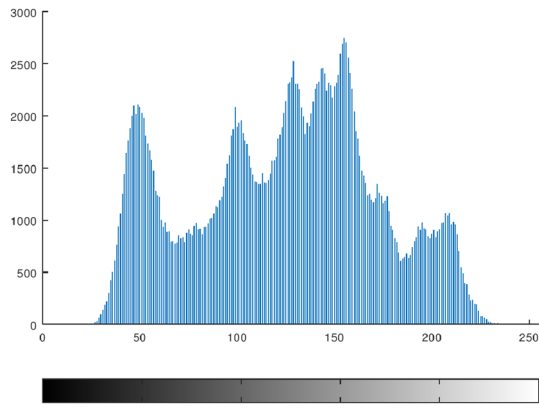
$$\epsilon = \sum_{i=0}^{mn} R(G_i) \log_2 \frac{1}{R(G_i)} \quad (20)$$

Here, $R(G_i)$ is the histogram count of the image. m and n are the total number of rows and columns of the cipher image, respectively. The detailed test results are given in Table 3. From Table 3, it can be concluded that no successful attack can be performed because the entropy values achieved are very close to the hypothetical values of entropy (Zhu 2012; Hua et al. 2018).

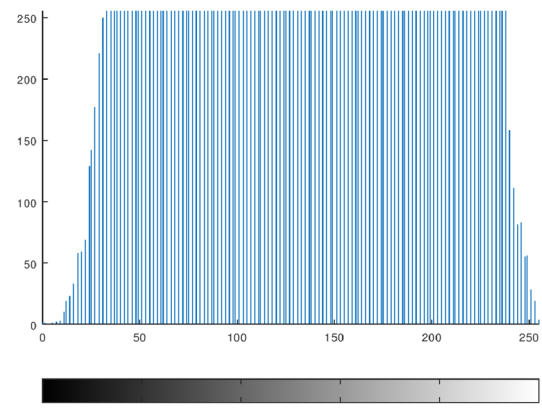
5.1.7 Encrypted image quality analysis

In this analysis, the quality of the encrypted image has been investigated using two standard error metrics, mean squared error (MSE) and peak signal-to-noise ratio (PSNR) Hamza et al. (2019); Wu et al. (2015); Nayak et al. (2018). To achieve high randomness in encrypted image, high PSNR score (> 30) is required. These MSE and PSNR metrics are computed using Eqs. (21) and (22). Here, I_p and I_c are the plain image and cipher image, respectively. W and H represents row and column of the image. Z is the maximum supported pixel value in the image matrix.

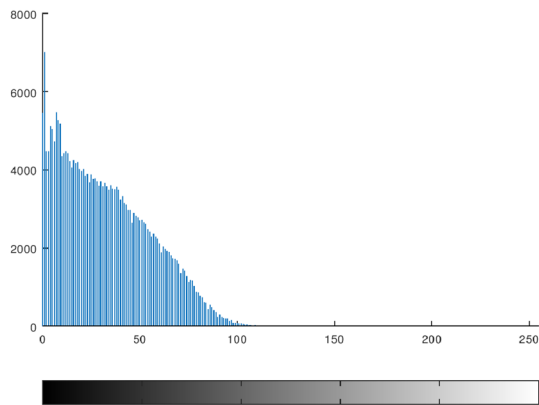
High PSNR and low MSE values signify good encryption quality. A comparison has been performed with the encryption schemes of Gupta and Silakari (2012) and Huang and Nien (2009). The result is shown in Table 4. The values of MSE and PSNR when plain image and decrypted image are compared is shown in Table 5. From this table it can be observed that value of MSE is '0' and value of PSNR is ∞ . This proves that the proposed encryption scheme is a loss-less encryption scheme.



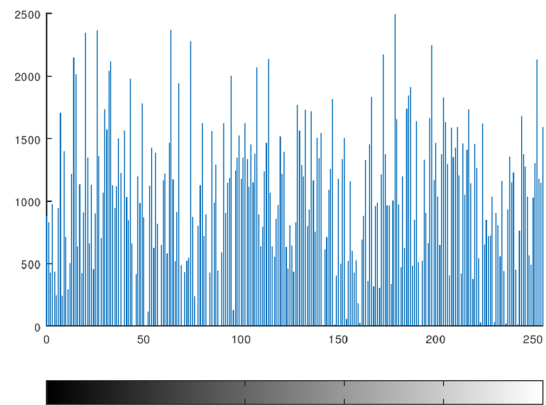
(a) Histogram of Lena Plain Image



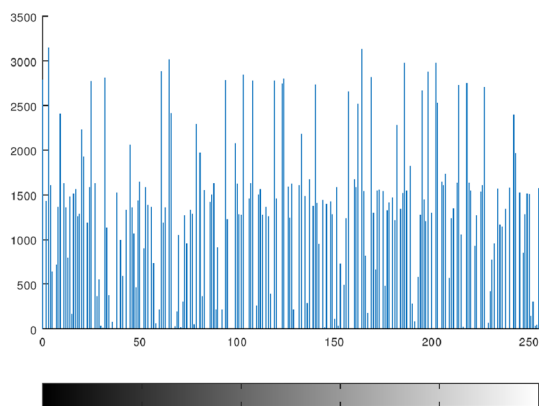
(b) Histogram of Fingerprint Plain Image



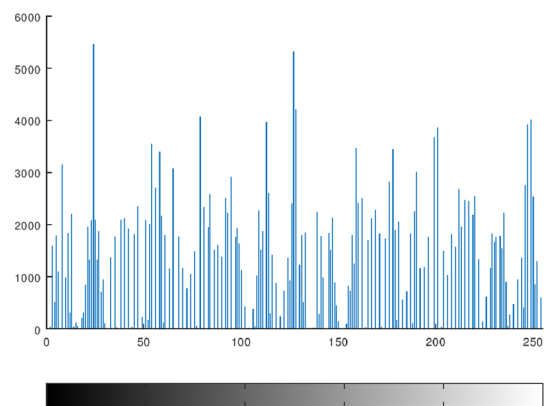
(c) Histogram of Baboon Plain Image



(d) Histogram of Encrypted Lena Image



(e) Histogram of Encrypted Fingerprint Image



(f) Histogram of Encrypted Baboon Image

Fig. 12 Histogram plot of original and encrypted images

Table 2 Result of correlation coefficient analysis

Image	Type	Horizontal	Vertical	Diagonal
Lena	Plain	0.97193	0.98503	0.95933
	Cipher	0.0080415	0.089618	0.010518
Fingerprint	Plain	0.93598	0.91870	0.85756
	Cipher	-0.0007267	0.0057593	0.00019278
Baboon	Plain	0.86650	0.75870	0.72614
	Cipher	-0.0038987	0.021038	-0.0022059

$$MSE = \frac{\sum_{w,h} [I_p(w,h) - I_c(w,h)]^2}{W \times H} \quad (21)$$

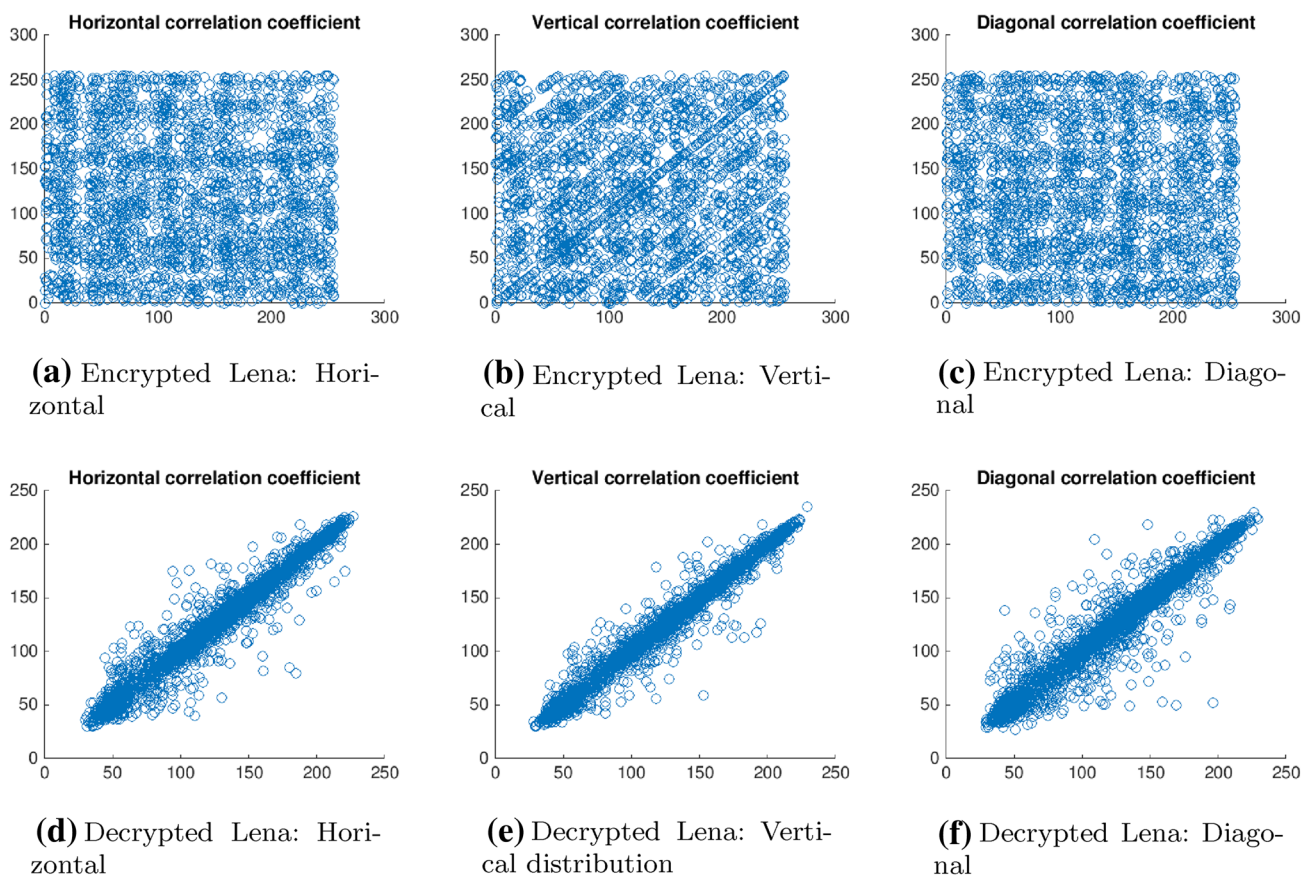
$$PSNR = 10 \log_{10} \left(\frac{Z^2}{MSE} \right) \quad (22)$$

5.1.8 Robustness test against different levels of noise

A thorough testing against different levels of noise has been performed to justify the robustness of the proposed scheme. We have added Salt and Pepper noise to the plain images and then they are sent to the network layer. The result of decryption at 1% and 10% noise level are shown in Figs. 16 and 17 respectively. It can be clearly understood from the figures that the decryption works sufficiently well to identify the plain images.

5.1.9 Differential analysis

Differential analysis is performed through number of pixel change rate (NPCR) and unified averaged changing intensity (UACI). These two quantities signify resistance of any encryption algorithm against differential cryptanalysis attacks. NPCR and UACI was first used in Chen et al. (2004). Since then, they are widely used for image

**Fig. 13** Correlation of two adjacent pixels of Lena image in horizontal, vertical and diagonal spectrum

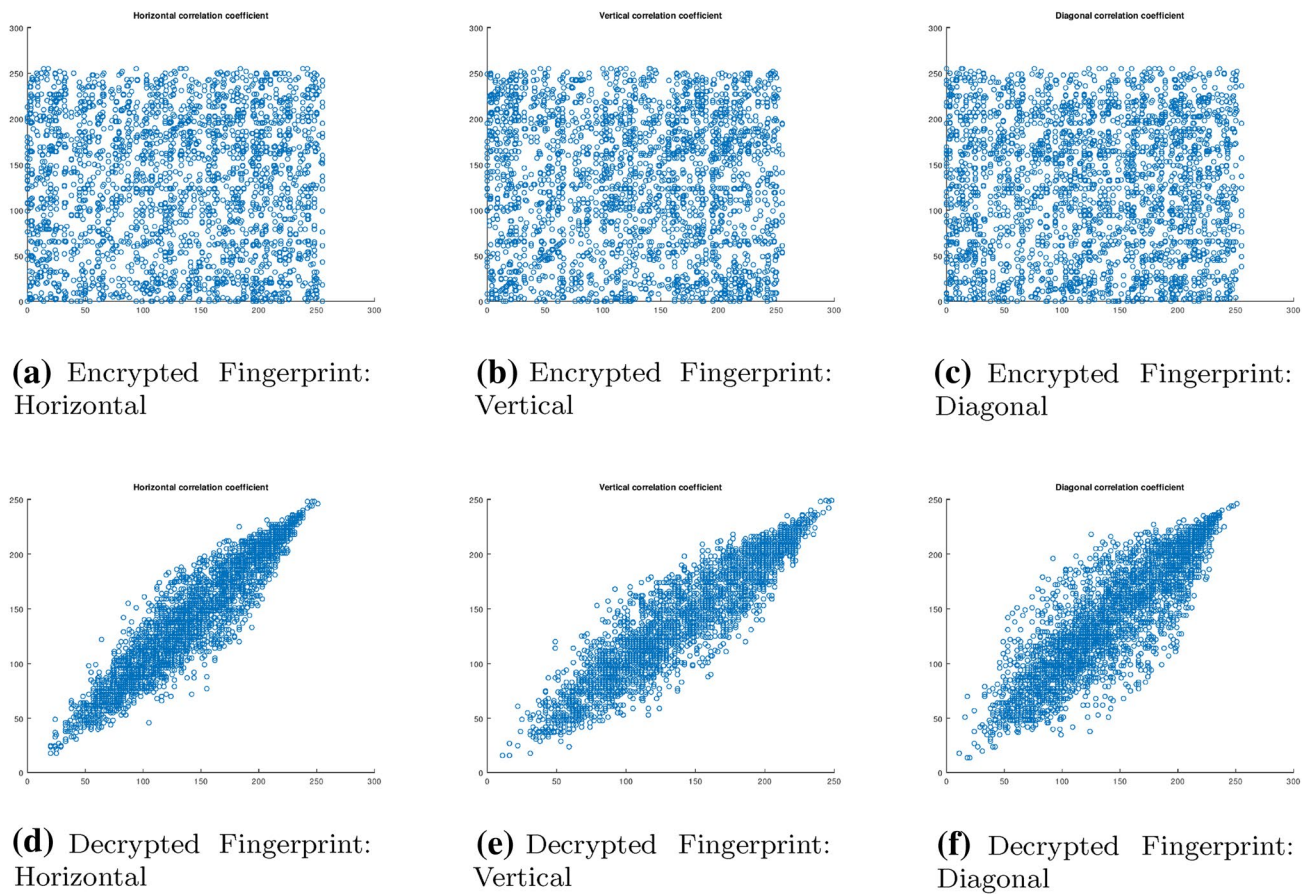


Fig. 14 Correlation of two adjacent pixels of Fingerprint image in horizontal, vertical and diagonal spectrum

encryption algorithms. NPCR refers to the rate of change of numbers of pixels when only one pixel is changed in the plain image. On the other hand, UACI is a measure of finding average intensity of differences between the plain image and the cipher image (Hua and Zhou 2016). Suppose, I_1 is the cipher image before one pixel change in plain image and I_2 is the cipher image after one pixel change in the plain image. Pixel values of I_1 and I_2 in the image matrix I , defined by Eq. (23), are $I_1(i, j)$ and $I_2(i, j)$, respectively. NPCR and UACI values are computed through Eqs. (24) and (25). Here, G and H are width and height of the image, respectively, while B is the maximum supported pixel value in the cipher image. In case of gray-scale image, it will be 255.

$$I(i, j) = \begin{cases} 0 & \text{if } I_1(i, j) = I_2(i, j) \\ 1 & \text{if } I_1(i, j) \neq I_2(i, j) \end{cases} \quad (23)$$

$$NPCR, N(I_1, I_2) = \sum \frac{I(i, j)}{G \times H} \times 100\% \quad (24)$$

$$UACI, U(I_1, I_2) = \sum_{i,j} \frac{|I_1(i, j) - I_2(i, j)|}{B \times G \times H} \times 100\% \quad (25)$$

The expected values of NPCR and UACI should be close to 99.61% and 33.44% (Hua and Zhou 2016). The detailed results of NPCR and UACI are shown in Table 6. It can be observed from Table 6 that the obtained values of NPCR and UACI is greater than the threshold values as mentioned in Hua and Zhou (2016). Hence, It can be concluded that the proposed scheme is resistant against differential cryptanalysis attacks.

5.1.10 DIEHARD tests

DIEHARD tests were first introduced by George Marsaglia¹ (1995) for testing the quality of a given random-number generator. It contains a battery of highly effective statistical

¹ <https://web.archive.org/web/20160125103112/http://stat.fsu.edu/pub/diehard/>.

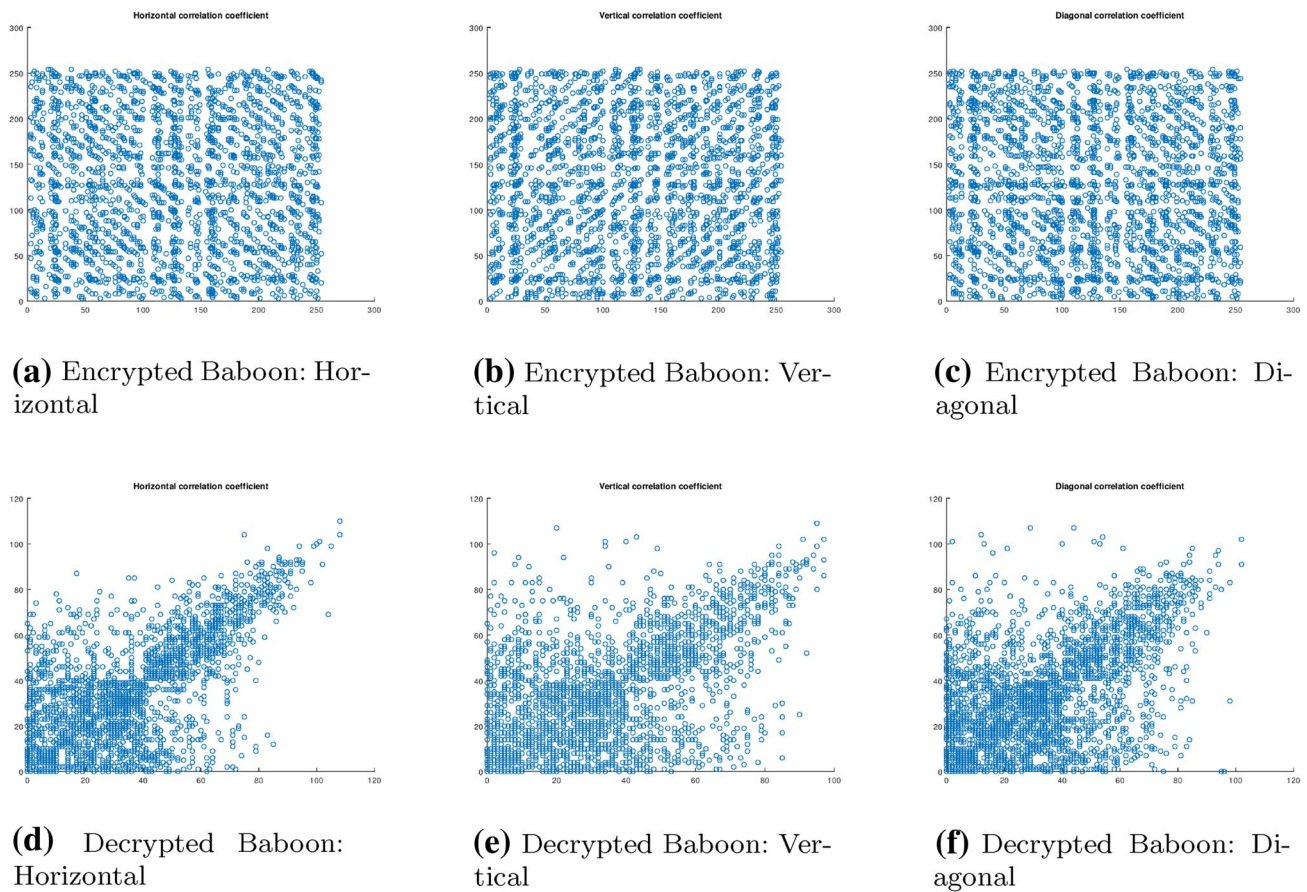


Fig. 15 Correlation of two adjacent pixels of Baboon image in horizontal, vertical and diagonal spectrum

Table 3 Results of information entropy analysis

Image	Image type	Entropy
Lena	Plain	7.445
	Cipher	7.9646
Fingerprint	Plain	6.7279
	Cipher	7.3044
Baboon	Plain	7.1391
	Cipher	7.9823

Table 4 Comparison of quality of encrypted image with existing schemes

Image	Lena		Baboon	
	MSE	PSNR	MSE	PSNR
IECA	49.99	65.12	0.277	42.56
Gupta and Silakari (2012)	11.589	37.48	8.149	39.02
Huang and Nien (2009)	0.604	50.32	0.404	52.07

tests for testing random numbers. The detailed results of the proposed cipher after going through DIEHARD tests are shown in Table 7. It can be observed from Table 7 that all the p-values obtained are under the range [0, 1). This denotes that the cipher image contains truly independent bits of random sequences.

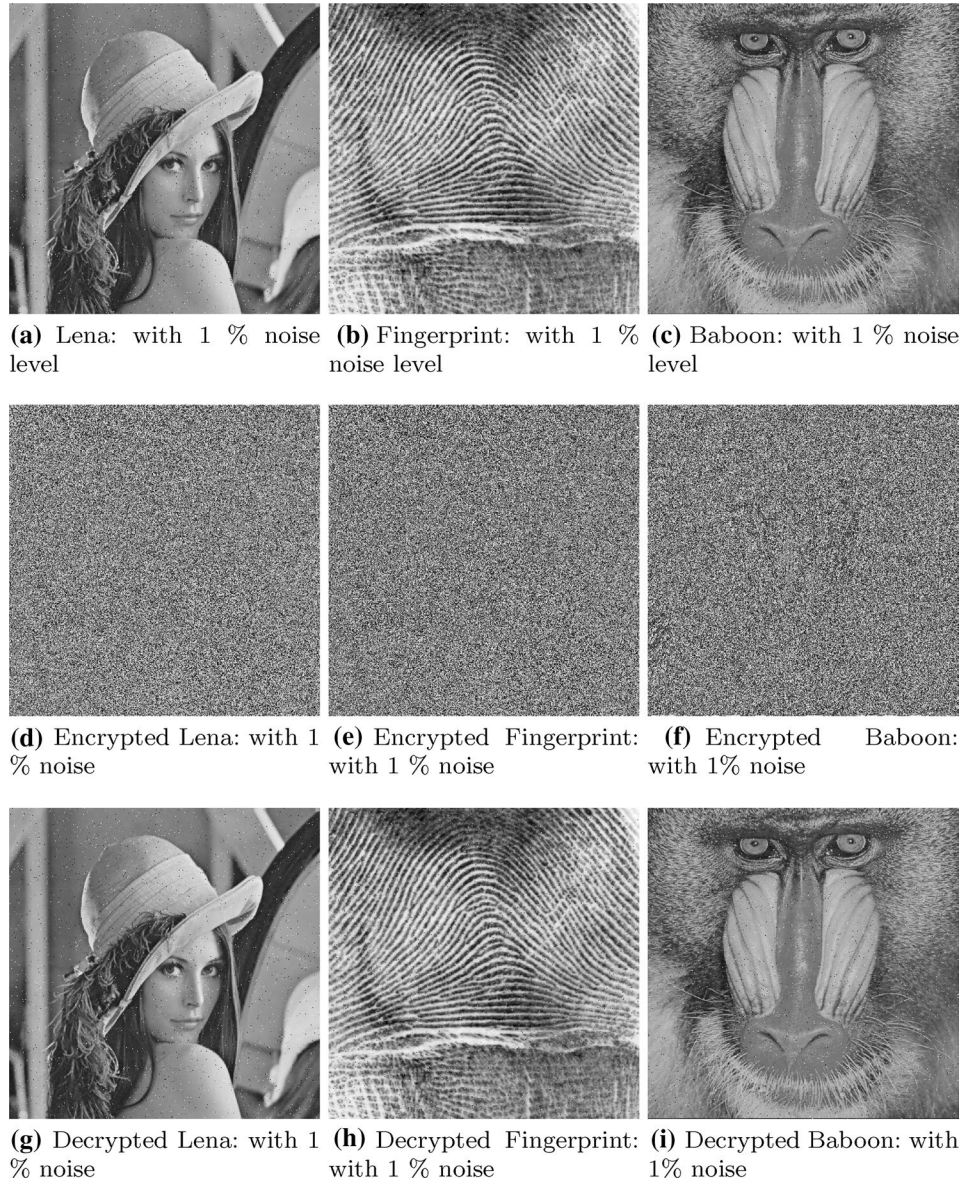
5.1.11 NIST randomness tests

Different randomness tests prescribed by NIST² have been performed to check the randomness of the proposed scheme. The test results are shown in Table 8 for cipher image of Lena. Similarly, NIST tests have been performed for the Fingerprint and Baboon images. Here, the reference distribution and decision rule for being non-random is also mentioned against each test. For example, Block Frequency Test follows Chi-square (χ^2) distribution. If the measured p-value is lower than 0.01, then the sequence is considered

² <https://www.nist.gov/publications/statistical-test-suite-random-and-pseudorandom-number-generators-cryptographic>.

Table 5 Result of MSE and PSNR between Plain Image and Decrypted Image

Image Scheme/metrics	Lena		Baboon		Fingerprint	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
IECA	0	∞	0	∞	0	∞

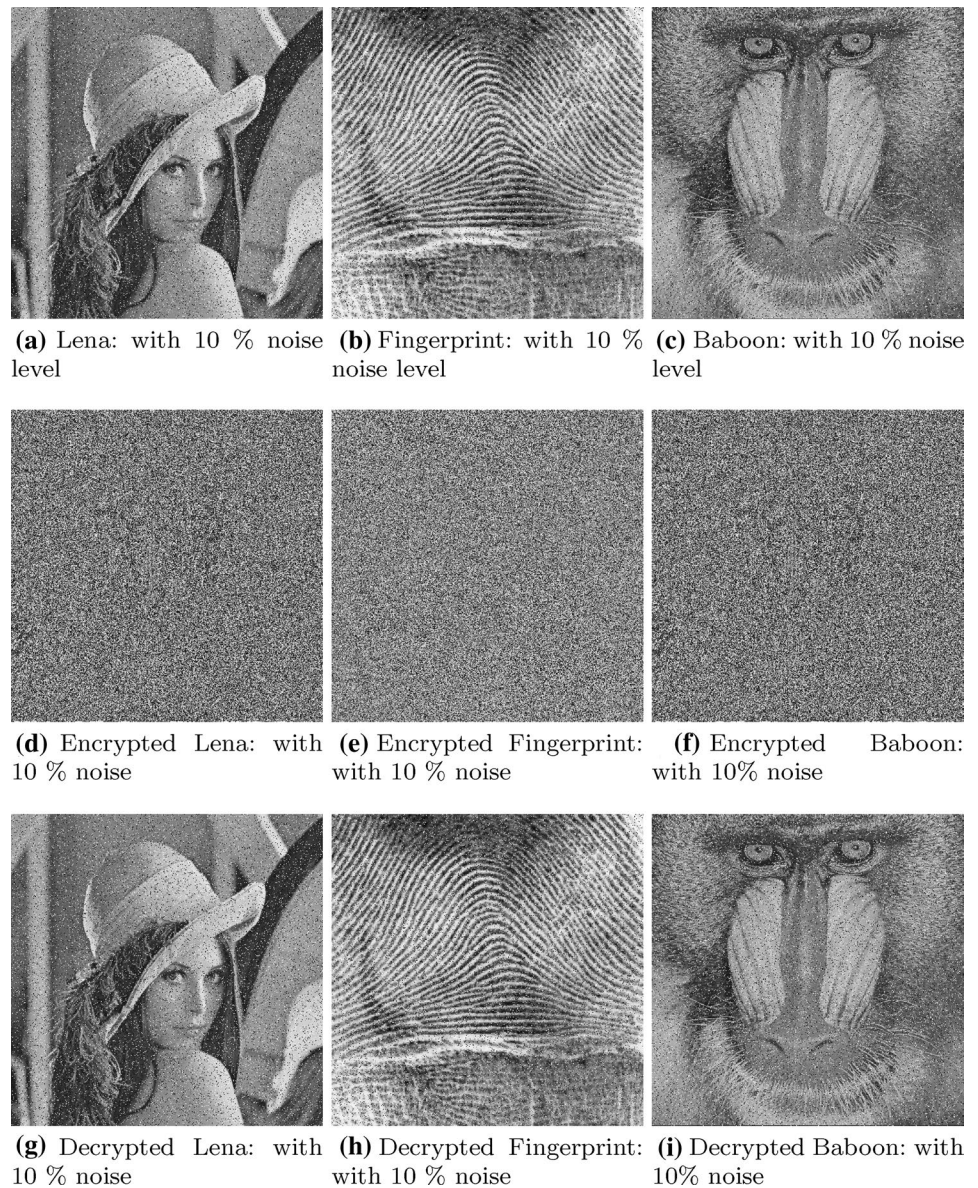
Fig. 16 Robustness test with 1 % noise level

as non-random, otherwise it is random. From Table 8, it can be observed that all the p-values are above the decision threshold value. Hence, the cipher image generated by the proposed scheme is random.

5.2 Comparison

IECA is compared with AES, DES, 3DES and Chacha. AES, DES and 3DES are symmetric key ciphers (Feistel 1973), whereas Chacha is a stream cipher from Salsa20 family

(Bernstein 2008). Chacha is lightweight and has comparatively less complex operations. All the existing ciphers were applied in a pixel wise manner to the input image. The comparison was carried out in terms of image size and the corresponding Z-value obtained in case of runs tests, where the normal distribution is considered as reference distribution. If the value of the normal variate, Z is lesser than 1.96, then it is considered as random (Ross 2017). The result of the comparison is shown in Fig. 18. It can be observed from Fig. 18 that the values of Z, in case of IECA is much lesser

Fig. 17 Robustness test with 10 % noise level**Table 6** Result of differential analysis

Test/cipher image	Lena	Fingerprint	Baboon
NPCR	99.78342	99.67668	99.83745
UACI	33.84126	33.751277	33.88235

than the threshold value. Hence the generated image-cipher contains high degree of randomness. The value of Z grows in almost similar to Chacha cipher, implementation complexity of which is more. However, the randomness obtained is better in case of standard symmetric key ciphers AES, DES and 3DES, but these ciphers cannot be implemented in perception layer of IoT applications because of resource constraints of the sensors. Besides, these ciphers have complex structures, where many intermediate operations are used to

provide high degree of randomness, confusion and diffusion property (Feistel 1973). Hence, these require larger memory size and computation power.

Apart from this, IECA is compared with the aforementioned ciphers to assess the runtime against increasing size of the images captured by camera sensors. Result of this comparison is shown in Fig. 19. Since, AES, DES and 3DES have more complex operations, their runtime is also larger as compared to Chacha and IECA. Chacha is a stream cipher but due to its implementation strategy, it takes more runtime than that of IECA. It can be noted from Fig. 19 that the runtime of IECA is the least among all the other ciphers and hence it establishes IECA as a better scheme to be applied in IoT. IECA is also compared with the aforementioned ciphers for each randomness tests prescribed by NIST. In each case, the proposed scheme gives better performance

Table 7 Results of DIEHARD Tests

Sl. no.	Test name	Result	p-value
I	Binary rank 32*32	Pass	0.36546728
II	Binary rank 6*8	Pass	0.91364748
III	Birthday spacings	Pass	0.75982631
IV	Overlapping 5-permutation	Pass	0.84562713
V	OQSO	Pass	0.95374828
VI	OPSO	Pass	0.94827162
VII	Bitstream	Pass	0.13264782
VIII	DNA	Pass	0.45289397
IX	Count-the-1's	Pass	0.79546271
X	Count-the-1's 2	Pass	0.91873232
XI	Minimum distance	Pass	0.84986721
XII	Squeeze	Pass	0.16726368
XIII	Overlapping sums	Pass	0.14782923
XIV	3D spheres	Pass	0.16739432
XV	Parking lot	Pass	0.61728936
XVI	Craps	Pass	0.87963727
XVII	Runs	Pass	0.84561216

than Chacha. However, the obtained result from these tests suggest that the randomness is high in case of AES and 3DES, but it gives almost similar randomness as that of DES and better randomness than Chacha, because of the structural complexities of these ciphers which are already mentioned before.

6 Conclusion

This work presents a lightweight, IoT friendly, and PCA-based image encryption scheme, called IECA. This scheme is capable of generating random cipher-images, nullifying

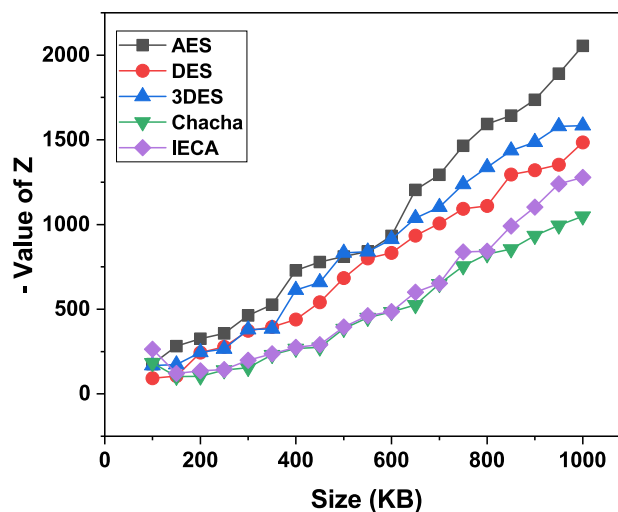


Fig. 18 Size vs. Z-value plot

the consequences of the attacks at the insecure communication channel between the perception layer and network layer, where the Fog nodes are deployed. The proposed algorithm can be implemented in critical real-world scenarios where the raw data play a vital role. For example, in cases of healthcare, defense-sector, biomedical image communication, etc. In each of these cases, the raw data sensed by the sensors need to be sent to the intended receiver unaltered. The communication needs to be fast as well as encrypted, as any adversarial attack would be life threatening. In such critical cases the proposed scheme can be useful. The IECA is efficient in terms of providing random cipher. The cipher-images generated by IECA have passed all the statistical randomness tests mentioned in NIST and DIEHARD test suites. Also, various standard analysis like differential analysis, key sensitivity analysis, correlation analysis, entropy

Table 8 Result of NIST randomness tests

Test name	Result	P-value	Reference distribution	Decision rule at 1%
Block frequency	Pass	0.193283	Chi-square	< 0.01
Frequency	Pass	0.853627	Half-normal	< 0.01
Cumulative sum	Pass	0.967352	Normal	< 0.01
Longest run of 1's	Pass	0.195346	Chi-square	< 0.01
Runs	Pass	0.195825	Chi-square	< 0.01
The binary matrix rank	Pass	0.172839	Chi-square	< 0.01
Discrete Fourier (Spectral)	Pass	0.048732	Normal	< 0.01
Maurer's universal statistical	Pass	0.983226	Half-normal	< 0.01
Non-overlapping template matching	Pass	0.982563	Chi-square	< 0.01
Approximate entropy	Pass	0.987236	Chi-square	< 0.01
Linear complexity	Pass	0.111527	Chi-square	< 0.01
Serial	Pass	0.120393	Chi-square	< 0.01

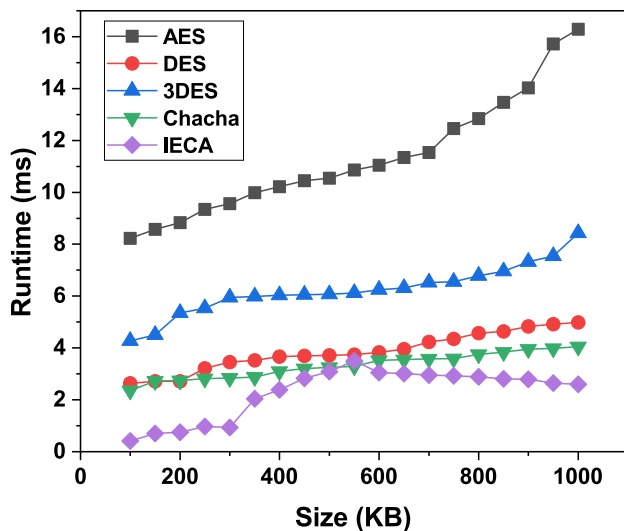


Fig. 19 Size vs. runtime plot

analysis, and image quality analysis confirms the robustness of IECA. It guarantees that the raw data would not be intelligible to the adversary who want to perform cryptanalysis. IECA also have significantly low time complexity that would contribute in faster communication. In future, IECA can be parallelized to have much lower runtime. It can be made more cache-efficient to reduce the time to fetch rule vectors. A cache-efficient algorithm will be more lightweight in nature and will help in increasing sensor's battery life more.

References

- Aizawa Y, Nishikawa I (1986) Dynamical systems and nonlinear oscillations. In: Proceedings of the symposium volume 1 of advanced series in dynamical systems, World Scientific, ISBN 9814704288, 9789814704281
- Angelescu P (2012) Hardware implementation of programmable cellular automata encryption algorithm. In: 35th International conference on telecommunications and signal processing (TSP), IEEE, pp 18–21
- Babaei A, Motameni H, Enayatifar R (2020) A new permutation-diffusion-based image encryption technique using cellular automata and DNA sequence. *Optik* 203:164000. <https://doi.org/10.1016/j.ijleo.2019.164000>
- Bakhshandeh A, Eslami Z (2013) An authenticated image encryption scheme based on chaotic maps and memory cellular automata. *Opt Laser Eng* 51(6):665–673
- Beniani R, Faraoun KM (2018) A mixed chaotic-cellular automata based encryption scheme for compressed jpeg images. *JMPT* 9(3):88–101
- Bernays P (1936) An unsolvable problem of elementary number theory. *Am J Math* 58:345–363 *J Symbol Log* 1(2):73–74
- Bernstein DJ (2008) Chacha, a variant of salsa20. *Works Rec SASC* 8:3–5
- Beyer WA, Sellers PH, Waterman MS (1985) Stanislaw M. Ulam's contributions to theoretical theory. *Lett Math Phys* 10(2):231–242
- Bisht A, Dua M, Dua S (2019) A novel approach to encrypt multiple images using multiple chaotic maps and chaotic discrete fractional random transform. *J Ambient Intell Hum Comput* 10(9):3519–3531
- Chai X, Zheng X, Gan Z, Han D, Chen Y (2018) An image encryption algorithm based on chaotic system and compressive sensing. *Sig Process* 148:124–144
- Chen G, Mao Y, Chui CK (2004) A symmetric image encryption scheme based on 3d chaotic cat maps. *Chaos Solit Fractal* 21(3):749–761
- Culik K II, Yu S (1988) Undecidability of CA classification schemes. *Complex Syst* 2(2):177–190
- Enayatifar R, Guimarães FG, Siarry P (2019) Index-based permutation-diffusion in multiple-image encryption using dna sequence. *Opt Lasers Eng* 115:131–140
- Eslami Z, Kabirirad S (2019) A block-based image encryption scheme using cellular automata with authentication capability. In: AIP conference proceedings, AIP Publishing, vol 2183, p 080002
- Feistel H (1973) Cryptography and computer privacy. *Sci Am* 228(5):15–23
- Guan SU, Tan SK (2004) Pseudorandom number generation with self-programmable cellular automata. *IEEE Trans Comput Aided Des Integr Circ Syst* 23(7):1095–1101
- Guan SU, Zhang S (2003) An evolutionary approach to the design of controllable cellular automata structure for random number generation. *IEEE Trans Evol Comput* 7(1):23–36
- Gupta K, Silakari S (2012) Novel approach for fast compressed hybrid color image cryptosystem. *Adv Eng Softw* 49:29–42
- Gupta A, Singh D, Kaur M (2019) An efficient image encryption using non-dominated sorting genetic algorithm-III based 4-D chaotic maps. *J Ambient Intell Hum Comput*. <https://doi.org/10.1007/s12652-019-01493-x>
- Gutowitz H (1989) Classification of cellular automata according to their statistical properties. Center for Nonlinear Studies, Los Alamos National Lab
- Hamza R, Yan Z, Muhammad K, Bellavista P, Titouna F (2019) A privacy-preserving cryptosystem for IoT e-healthcare. *Inf Sci* <https://doi.org/10.1016/j.ins.2019.01.070>
- Hortensius PD, McLeod RD, Card HC (1989) Parallel random number generation for vlsi systems using cellular automata. *IEEE Trans Comput* 38(10):1466–1473
- Hua Z, Zhou Y (2016) Image encryption using 2d logistic-adjusted-sine map. *Inf Sci* 339:237–253
- Hua Z, Zhou B, Zhou Y (2018) Sine chaotification model for enhancing chaos and its hardware implementation. *IEEE Trans Ind Electron* 66(2):1273–1284
- Huang CK, Nien HH (2009) Multi chaotic systems based pixel shuffle for image encryption. *Opt Commun* 282(11):2123–2127
- Kaur M, Kumar V (2018a) Adaptive differential evolution-based lorenz chaotic system for image encryption. *Arab J Sci Eng* 43(12):8127–8144
- Kaur M, Kumar V (2018b) Beta chaotic map based image encryption using genetic algorithm. *Int J Bifurcat Chaos* 28(11):1850132
- Kaur M, Kumar V (2018c) Colour image encryption technique using differential evolution in non-sampled contourlet transform domain. *IET Image Proc* 12(7):1273–1283
- Kaur M, Kumar V (2018d) Efficient image encryption method based on improved lorenz chaotic system. *Electron Lett* 54(9):562–564
- Kaur M, Kumar V (2018e) Fourier-mellin moment-based intertwining map for image encryption. *Mod Phys Lett B* 32(09):1850115
- Kaur M, Kumar V (2018f) Parallel non-dominated sorting genetic algorithm-II-based image encryption technique. *Image Sci J* 66(8):453–462

- Kaur M, Kumar V (2020) A comprehensive review on image encryption techniques. *Arch Comput Methods Eng* 27:15–43. <https://doi.org/10.1007/s11831-018-9298-8>
- Kaur M, Kumar V, Li L (2019) Color image encryption approach based on memetic differential evolution. *Neural Comput Appl* 31(11):7975–7987
- Li W, Packard N (1990) The structure of the elementary cellular automata rule space. *Complex Syst* 4(3):281–297
- Li W, Packard NH, Langton CG (1990) Transition phenomena in cellular automata rule space. *Physica D* 45(1–3):77–94
- Li X, Zhang G, Zhang X (2015) Image encryption algorithm with compound chaotic maps. *J Ambient Intell Hum Comput* 6(5):563–570
- Nandi S, Kar B, Chaudhuri PP (1994) Theory and applications of cellular automata in cryptography. *IEEE Trans Comput* 43(12):1346–1357
- Nayak P, Nayak SK, Das S (2018) A secure and efficient color image encryption scheme based on two chaotic systems and advanced encryption standard. In: 2018 international conference on advances in computing, communications and informatics (ICACCI), IEEE, pp 412–418
- Neebel DJ, Kime CR (1997) Cellular automata for weighted random pattern generation. *IEEE Trans Comput* 46(11):1219–1229
- von Neumann J (1951) The general and logical theory of automata. In: Jeffress LA (ed) *Cerebral mechanisms in behaviour*. Wiley, New Jersey
- Neumann J, Burks AW et al (1966) *Theory of self-reproducing automata*, vol 1102024. University of Illinois Press, Urbana
- Petrica L (2018) Fpga optimized cellular automaton random number generator. *J Parallel Distrib Comput* 111:251–259
- Ross SM (2017) *Introductory statistics*. Academic Press, Cambridge
- Roy S, Rawat U, Karjee J (2019) A lightweight cellular automata based encryption technique for iot applications. *IEEE Access* 7:39782–39793
- Shaheen AM, Sheltami TR, Al-Kharoubi TM, Shakshuki E (2019) Digital image encryption techniques for wireless sensor networks using image transformation methods: DCT and DWT. *J Ambient Intell Hum Comput* 10(12):4733–4750
- Singh D, Tripathi G, Jara AJ (2014) A survey of internet-of-things: future vision, architecture, challenges and services. In: 2014 IEEE world forum on internet of things (WF-IoT), IEEE, pp 287–292
- Sirakoulis GC (2012) Hybrid DNA cellular automata for pseudorandom number generation. In: 2012 International conference on high performance computing and simulation (HPCS), IEEE, pp 238–244
- Suri S, Vijay R (2019) A synchronous intertwining logistic MAP-DNA approach for color image encryption. *J Ambient Intell Hum Comput* 10(6):2277–2290
- Toffoli T, Margolus NH (1990) Invertible cellular automata: a review. *Physica D* 45(1–3):229–253
- Tsalides P, York T, Thanailakis A (1991) Pseudorandom number generators for VLSI systems based on linear cellular automata. *IEE Proc E Comput Digit Techniq* 138(4):241–249
- Ulam S (1952) *Random processes and transformations*. Proc Int Congress Math Citeseer 2:264–275
- Wolfram S (1984) Universality and complexity in cellular automata. *Physica D* 10(1–2):1–35
- Wolfram S (1985) *Cryptography with cellular automata*. In: Conference on the theory and application of cryptographic techniques, Springer, New York, pp 429–432
- Wolfram S (1986a) *Random sequence generation by cellular automata*. *Adv Appl Math* 7(2):123–169
- Wolfram S (1986b) *Theory and applications of cellular automata*. World Scientific, Singapore
- Wolfram S (2002) *A new kind of science*, vol 5. Wolfram Media, Champaign
- Wu H, Wang H, Zhao H, Yu X (2015) Multi-layer assignment steganography using graph-theoretic approach. *Multimed Tools Appl* 74(18):8171–8196
- Wu X, Wang K, Wang X, Kan H (2017) Lossless chaotic color image cryptosystem based on DNA encryption and entropy. *Nonlinear Dyn* 90(2):855–875
- Wuensche A (2011) *Exploring discrete dynamics*. Luniver Press, Florida
- Zhu C (2012) A novel image encryption scheme based on improved hyperchaotic sequences. *Opt Commun* 285(1):29–37

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.