Towards a Hybrid Approach Based on Elliptic Curves and Cellular Automata to Encrypt Images

Hichem Bouchakour Errahmani
Djillali Liabes University Of Sidi-bel-abbes
Algeria
B_Hichem@Hotmail.fr

Kamel Mohamed Faraoun
Udl-sba
Algeria
B_Hichem@Hotmail.fr, Kamel_Mh@Yahoo.fr

ABSTRACT: *In this paper, we present a new approach of image encryption using a stream ciphering model. The proposed pseudo-random number generator is based on combination of elementary one-dimensional cellular automata (CA)with elliptic curves. Indeed, we explored the transitions of the CA with the coordinates of a point belonging to an elliptic curve. The outcome from another point was multiplied by a scalar. CA offers the qualities of ambiguity and chaos, so by combining the two concepts, we have constructed a PRNG that generates a key stream, used by the way in our approach. This work constitutes an analogy to works that involved dynamic systems like logistics map with elliptic curves. The proposed PRNG shows good cryptographic properties,and high security level that depends in the difficulty to solve the discrete logarithm problem on elliptic curves (ECDLP). The proposition evaluated the images encryption, and it turns out that obtained results are very promising.*

Keywords: Elliptic Curve Cryptography, Cellular Automata, Pseudo-Random Number Generator, Image Encryption

Received: 1 June 2017, Revised 12 July 2017, Accepted 25 July 2017

1. Introduction

Communication between peoples has always existed, and was developed over the centuries in its content; that was only writings, images, videos, and multimedia today, or in its means of transport. And with the tensions that prevailed among the peoples, which led to conflicts, competitions, and even wars, all of which pushed people to protect their communications and data, and hence the appearance is the cryptography. The science makes possible to encrypt data, and makes it incomprehensible for a person not authorized to access it. Thus the technology has continued to grow in performance, in miniaturization, and in techniques used; one of them is the elliptic curves, which have proved recently the efficiency in cryptography, especially by reducing the size of encryption keys, and their capabilities to be used in embedded systems, such as mobile phones, tablets, sensors, etc. Since we are in a century of multimedia, images and videos as well as their communications requires strong

protection with encryption tools, it is easy to implement and less expensive. The solution was to use a stream encryption system that is considered as the safest and the easiest to implement, especially on embedded systems, where processor and memory performance is limited.

In this paper, we propose a new stream encryption approach, whose pseudo-random number generator is based on the use of point coordinates generated randomly from an elliptic curve, and their mapping in a grid constructed by the transitions of an elementary cellular automata. The paper is organized as follows: the following section presents the two key concepts of this approach, namely elliptic curves and cellular automata. Section 3 reports some works using elliptic curves for the generation of pseudo-random numbers or sequences and their applications with dynamic systems such as logistic sequences in image encryption. In the next section, we present our construction approach. Section 5 shows the results obtained. Then we make a small comparison of our approach with another existing approach in Section 6. Finally, we end with a conclusion.

2. Preliminaries

2.1 Elliptic curve

Let F_p be a finite field. An elliptic curve is the set of points E satisfying the cubic equation [1]: $y^2 = x^3 + ax + bx, y, a, b \in F_p$

With $4a^3 + 27b^2 \neq 0$ since it is not singular.

This curve has a special point called the point at infinity noted *O*.

The set of points of the curve *E*, form an Abelian group $E(F_p)$ with the geometric addition +, with *O* as neutral element. The addition of the points is defined as follow:

Let $P(x_1, y_1)$ be a point of $E(F_p)$, if $Q(x_2, y_2) \in E(fp)$, such that $Q \neq -P$.

Then $(\boldsymbol{P + Q})(\boldsymbol{x_3}, \boldsymbol{y_3})$ is given as follow:

$$x_3 = \lambda^2 - x_1 - x_2 \quad \text{and} \quad y_3 = \lambda(x_1 - x_3) - y_1$$

Where :

$$\lambda = \begin{cases} (y_2 - y_1)(x_2 - x_1)^{-1}, siP \neq Q \\ (3x_1^2 + a)(2y_1)^{-1}, siP = Q \end{cases}$$

All of the calculations are made modulo p. To calculate the scalar multiplication of a point like $k.P$, we need two elementary operations; addition and doubling to accelerate the process, for example to calculate 7*P*, one must first, do the double of *P* to obtain 2*P*, then add a *P* to get3*P*, then do the double of it to get 6*P*, and finally add a *P*.

The Elliptic Curve Discrete Logarithm Problem (ECDLP) [1] is similar to its counterpart of the Discrete Logarithm Problem over integers (DLP), let us recall that the latter asserts that knowing the value of $y = x^k$ it is difficult to find in a finite time, the value of *k* which constitutes the discrete logarithm of *y*. Thus, given a point $P = m.G$ with *G* a point on the elliptic curve, then it is difficult to find the integer $m \in F_p$ in a finite time, *m* is called the elliptic discrete logarithm.

The group of points on an elliptic curve is an interesting group in cryptography insofar as there is no known algorithm sub exponential for its Discrete Logarithmic Problem. The advantage here, is that smaller exponents are used.

Also, we add some indispensable notions on the elliptic curves concerning our paper, the number of points of an elliptic curve *M* is called the order of the curve, the smallest integer for which *N*. *P* is equal to the Infinite point *O* is called the order of the point *P*, so $N \leq M$, if $N = M$ then *P* is said a generator point, if *P* exists in such a curve, this curve is called cyclic elliptic curve[2].

The origin of elliptic curves, dates back in 1985 through the work of the mathematician Koblitz[3]. It has been shown that elliptic curves are mathematical objects that have improved cryptographic features of many cryptosystems such as RSA, El Gamal, moreover just the key is 40 times smaller than that used in RSA, also the speed they offer is outstanding. These two characteris-

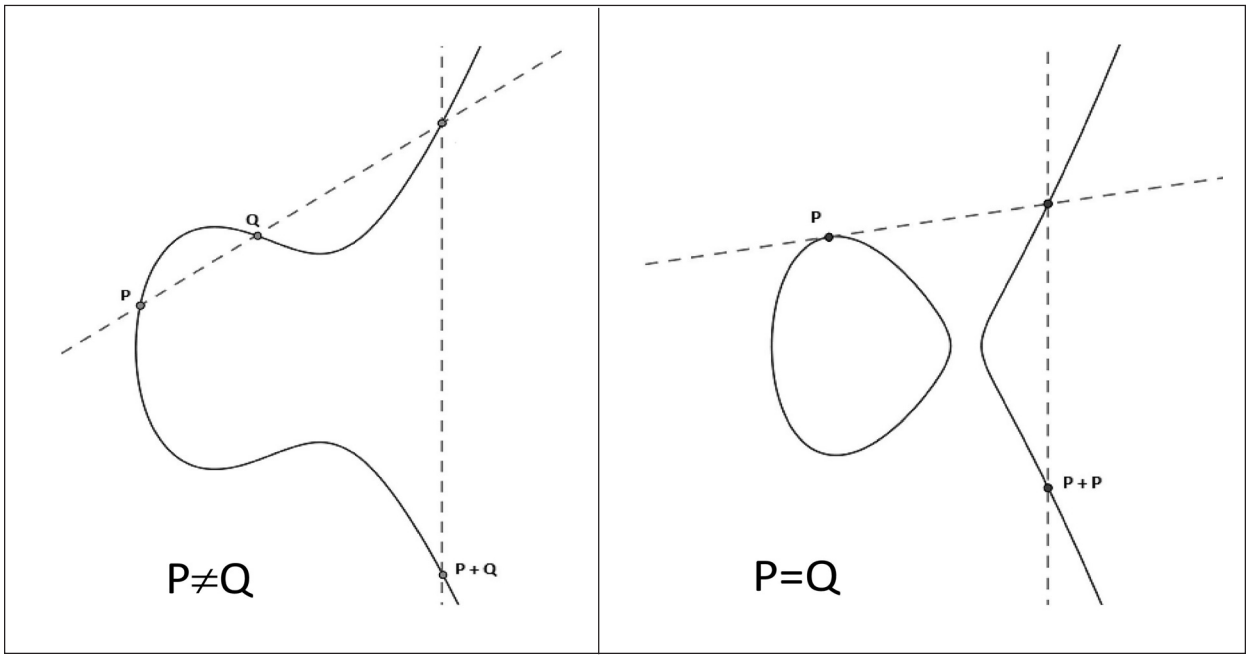


Figure 1. P+Q Addition

tics: the miniaturization of the key and speed of calculation in a finite field, have led to the use of cryptosystems based on elliptic curves in embedded systems in particular sensor networks, FPGAs, …etc.

2.2 Cellular Automata

A Cellular Automata (CA) is the dynamic evolution of a one-dimensional array into a grid, made up of cells, each having a well-defined state (1 or 0); this evolution is subject to a rule which provides the next state of the current cell, based on its current state and its neighboring cells.The CAs are part of the dynamic systems family, they evolve over time, progressively as the current status of the cell changes. They were invented by Von Neumann and developed by Wolfram in the 70 years [4].

Usually the most used rule in elementary CA is 30.According to Wolfram, it has a chaotic and unpredictable behavior, that is required in the construction of pseudorandom number generator, also CA were used in the construction of cryptographic systems, secret sharing schemes, etc. [5]. The choice of cellular automata in this paper is due to lack of works involving these with elliptic curves.

Initially CA is a one-dimensional array, where each cell indexed by *i,* at time *t,* has a state *s* which is 0 or 1. In a transition following a well-defined rule, the state of cell *i* at time *t* +1 depend on the old state and the neighboring cells statements:

$$s_{t+1}(i) = f(s_t(i-1), s_t(i), s_t(i+1))$$

So a *CA* containing two states and a neighborhood of space equal to 3, has 2^3 configurations, each configuration is named as a rule governing the *CA*, and each possible configuration leads to a state, so all possible rules are equal to 2^{2^3} or 256 possible rules [6].

Initial configuration of the neighborhood	111	110	101	100	011	010	001	000
The state of cell *i* at time $t+1$ according to rule 30 written in binary (00011110)	0	0	0	1	1	1	1	0

This type of cellular automata is named: elementary cellular automata. We can represent a *CA* with a drawing, where the black dots denote 0 and the white dots denote 1 (Figure 2).

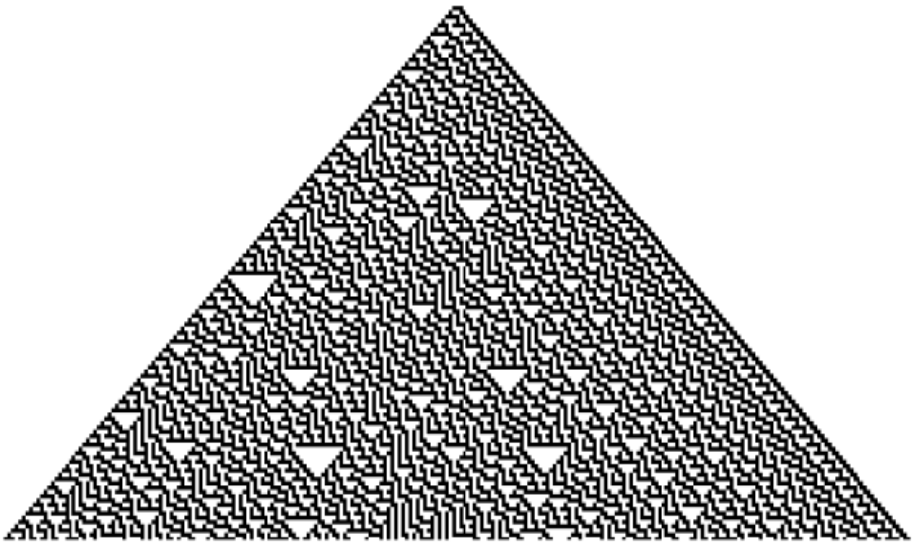


Figure 2. Transition of the CA rule 30 starting from a one black cell

3. Related Works

We will see some works based on elliptic curves and which have allowed to construct generators of pseudo random numbers.

Gong and others [7] have constructed a generator of pseudo-random sequences based on the multiplication of a scalar by a point and the trace function. Indeed, they began by generating a random point, multiplying it by $i = 1, \ldots, 2^n$ to have a point of coordinates x_i and y_i, then they calculated $a_i = trace(x_i)$ and $b_i = trace(y_i)$. Thus, the generated sequence is the concatenation a_i and b_i.

Beelen and others have used elliptic curves in a different way for the generation of pseudo-random sequences, based on additive and multiplicative characters on elliptic curves, and also, using linear recurrence relations [8].

Bayer has built a PRNG called ECPRNG based on the work of Kaliski that uses twisted elliptic curves [9], [10]. He chooses an elliptic curve E such as $E(F_p)$ and its twist $E^{tw}(F_p)$ have a primal order, of course, he says his system is safe. This is due to the difficulty of solving the discrete logrithm problem in elliptic curves [11].

Lee and others have constructed a pseudo-random number generator based on point addition operations in an elliptic curve. Initially, they chose a finite field F, an elliptic curve E, a point of this curve denoted P, and a seed k_1, after constructing the point k_1. P with a modulus of multiplication of a point by a scalar, its abscissa x, which will represent the pseudo-random sequence, is added to the number of iterations made is 1 (at the beginning) to have the new seed k_2, thus $k_{n+1} = x_n + n$, and successively this k_{n+1} will be multiplied By the point P [12].

Barker and others have proposed a pseudo-random number generator called "The Dual Elliptic Curve Pseudorandom Generator DEC PRG" based entirely on the points of an elliptic curve, they claim that the security of this generator is safe since it is highly related to the resolution of the Elliptic Curve Discrete Logarithm Problem (ECDLP). Indeed, the pseudo-random bits are extracted from a random point of an elliptic curve, exploiting the 240 least significant bits of the x-coordinate of this point, the elliptic curve is defined over a finite field F_p with $log_2 p = 256$. They show that these bits are distinct from the 240-bit pseudo-random from a uniform distribution [13].

In his paper, Laszlo Merai provides a PRNG based on the congruent generation points of the elliptic curve, through a rational function which depends on the coordinates of the generated point, it returns a pseudo-random sequence [14].

Payingat and others have proposed a pseudo-random number generator for stream cipher based on elliptic curve. The sequence is obtained by combining the traditional secret scalar multiplication operations with a point and the transition of an LFSR register, and therefore the sequence represents a set of bits sliced from the generated point [15].

Reyad and others proposed a new construction based on chaotic systems, and characteristic elliptic curves 2 i.e., $F(2^m)$. The addition of chaotic systems has made the cryptosystem faster and semantically safer. The pseudo-random sequence constitutes a point U (x, y) generated by the following relation: $U_i = G + U_{i-1} = [i]G + U_0$ where G is a given point in $E(F_p)$ having a very high order and U_0 is a point chosen initially [16].

In another paper, Laszlo Merai proposed a sequence generator (x_n) based on congruences in elliptic curves, defined by the following relation: $x_n = x(W_n) = x(W_{n-1} + G) = x(nG + W_0)$ such that W_0 is an initial point of an elliptic curve $E(F_p)$[17].

Little works have been proposed for the use of elliptic curves in image encryption, and this does not prevent to cite some works that have been seen.

Sathyanarayana and others developed a stream cipher system for digital images. They began by building a cyclic elliptic curve[18]. It is a curve which has a point P whose order is equal to the number of points on the elliptic curve, also called a generator point then they generated a seed k by the LFSR generator, which subsequently will be multiplied by P and for each pseudo-random seed, we have a pseudo random point whose its coordinates will be used as a keystream cipher under different aspects in an additive and affine encryption [19].

Abdellatif and others presented an image encryption using chaotic logistics map combined with cyclic elliptic curves. Their keystream generator begins by producing a seed k over the field F_p with a conventional generator, then, it multiplies this seed by a point P on the curve, giving a k. P sequence which coordinates x, y, which will be combined in an addition with a pseudo random output of a scheme based on logistic chaotic map. This diagram combines in addition the input (the secret key of the user) with two operands: the output of the schemeafter one iteration, and the key stream generated also afterone iteration[20].

Dawahdeh and others developed a cryptosystem for images that combines elliptic curves and HILL encryption. It is a symmetric algorithm but known to be insecure, where the idea is to transform the symmetrical aspect of encryption HILL to another that is asymmetrical, and that in order to strengthen its security. The results showed a good performance especially concerning entropy [21].

4. The Proposed Image Encryption Algorithm

4.1 The Pseudo-Random Sequence Generator

The proposed generator (figure 3) combines cellular automata with elliptic curves exploiting its multiplication operation of a point by a scalar, first a seed is generated by a conventional generator, for our case we opted for the LFSR generator that is so widespread in research, we could totally use a generator of any language such as the Random class of the Java language. This seed is then multiplied by a generator point of the cyclic elliptic curve and the result which is simply a point of our elliptic curve will be used as coordinate x and y that will match respectively to the row and the column of static grid generated by cellular automata modulo p, the designated cell that contains either a 1 or a 0 will be a bit of our random sequence. The process is repeated many times until to obtain a random sequence of "k" bits.

The algorithm used for multiplication of a point by a scalar is the famous Double-and-Add because its simplicity, the size of the cellular automata is 1024, also the number of transition is large enough for our case, we chose 1024, then the grid generated by the automata is a square matrix. Therefore the cell indicated by the coordinates (x, y) of the point $n_i.P$ correspond to the bit of the random sequence.

INPUT: P a point of the elliptic curve, n as a seed

OUTPUT: $b_1 b_2 \ldots b_i \ldots b_N$

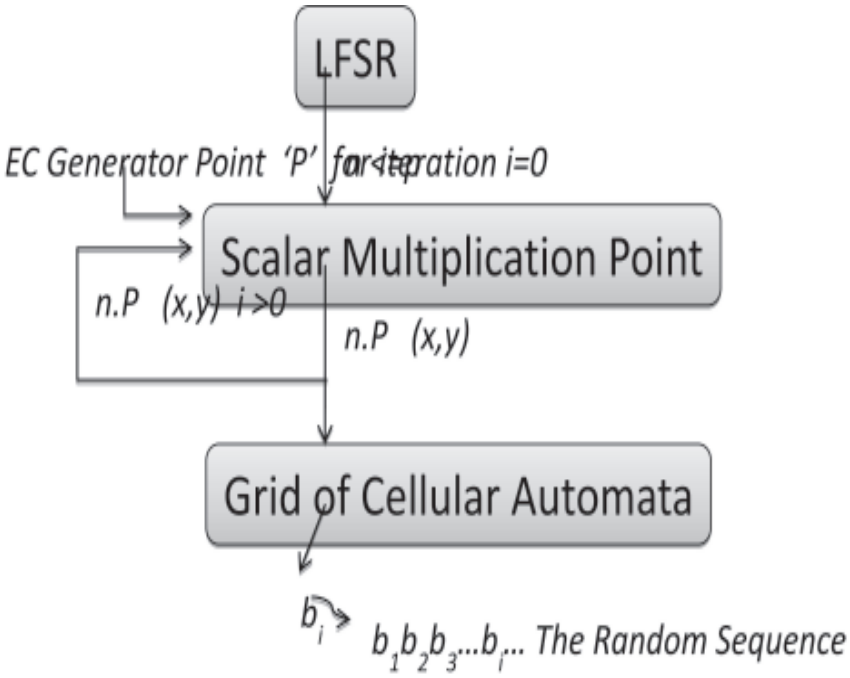


Figure 3. Pseudo-Random Sequence Generator Scheme

$For i = 1\ to N do$
  Compute $P \leftarrow n.P$ // (x,y) are the coordinate of the point $n.P$
  $b_i \leftarrow Grid(x, y)$

Algorithm: Generating the Pseudo Random Sequence

Where Grid represents the transition of the cellular automata obtained like a matrice.

			y							
	0	0	1	0	1	1	0	1	1	1
	1	0	1	0	0	0	0	1	0	1
	0	0	0	0	1	1	0	0	0	1
	1	0	0	1	1	1	0	0	0	0
x	0	0	0	0	1	0	1	0	1	1
	1	0	1	1	1	0	1	0	1	1
	0	0	0	0	0	0	0	0	1	0
	0	0	0	1	1	0	1	0	1	0

Example of Grid $(x, y) = 1$

We can propose several schemes for this algorithm to calculate b_i like:

$b_i \leftarrow Grid(x, x)$ or;
$b_i \leftarrow Grid(y, y)$…etc.

4.2 Encryption Scheme

The utility of this generator is to involve it,in the stream cipher of images, this encryption is considered as perfect and safe, also it is simple to implement, so once the random sequence is generated, itis used as a key in the Xor of the pixel (fig. 4), thus :

$$c_i = k_i \otimes p_i$$

Where:

p_i is a clear pixel

c_i is an encrypted pixel

k_i is a keystream encryption

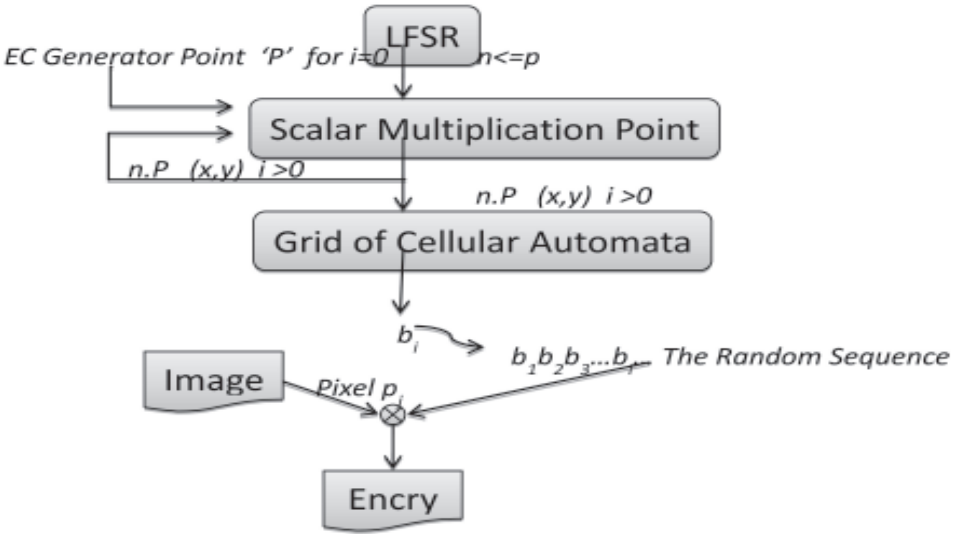


Figure 4. Proposed Construction

5. Experimental Results and Analysis

5.1 Security Aspects of the Proposed System

5.1.1 Key Analysis

Our PRNG is unpredictable, indeed, if we have the first $b_1, b_2, \ldots, b_i$ from the output of the PRNG then, one cannot compute the remainder of the bits $b_{i+1}, b_{i+2}, \ldots, b_n$, because b_{i+1} depends on the precedent state's grid such that $b_{i+1} = Grid(n.P)$ and it is provably secure, since one has to resolve the Elliptic Curve Discrete Logarithm Problem (ECDLP) of $n.P$ to get n.

5.2.2 ***Key space***

To have a secure cryptosystem, which relies on the secret of the key, the space of the latter must be large enough to prevent the effect of the brute force attack within a reasonable time. Our PRNG has a large and flexible key space, in fact, the key is constructed from a pair (n, P) and its dimension is as follows:

- The number of distinct elliptic curves on $GF(2^m)$ is $2(2^m - 1)$
- The possible number of points P is 2^{2m} .
- The number of possible values for n depends on the size of LFSR d, ie 2^d (for our case $d = 32$)

Therefore, the total number of possible keys is equal to the product of what has been stated: $2(2^m - 1) \times 2^{2m} \times 2^d$

3) DataSet Used

For our experimentation, we used a prime number p, an elliptic curve definer over F_p, and we choose a point P such as:

p = 14626471 is a prime number

a = 3692319

b = 415404 (a and b to define the elliptic curve)

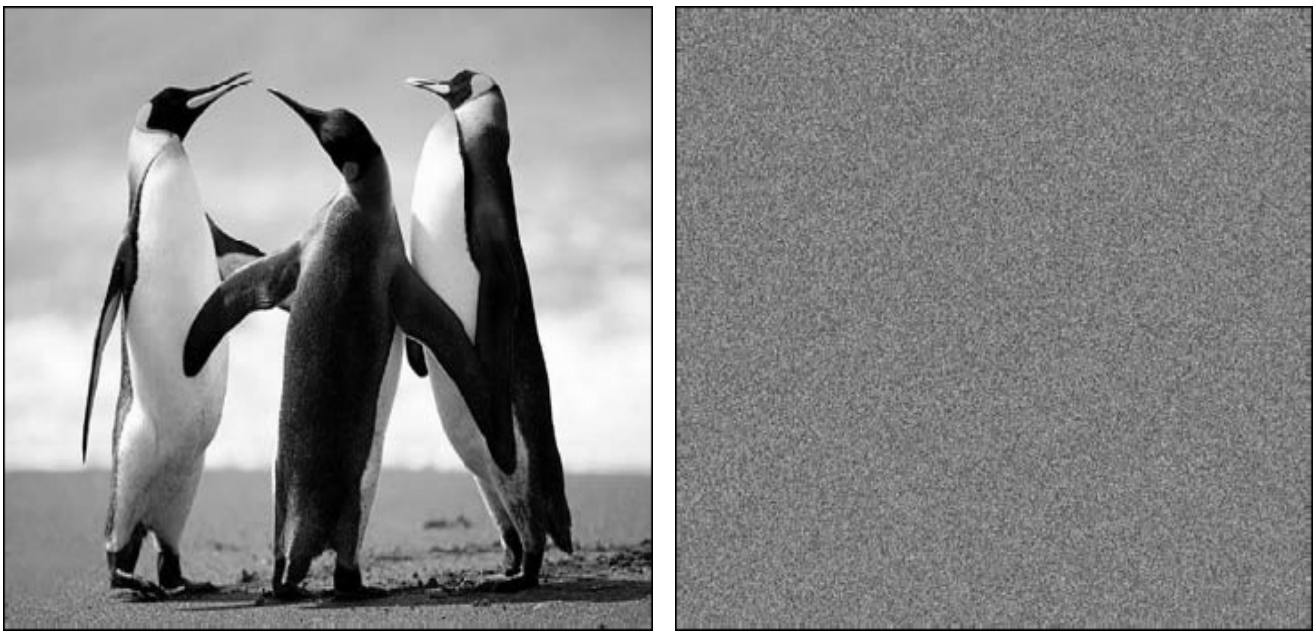
The point P: (311826,785) is the initial point.

The seed n = 2450236483

5.2 Results of Image Encryption

5.2.1 Image Encryption Illustration

We tested our system on a digital image which is most used in experiments, and it represent penguins, with resolution 1024×768, and size of 759 Ko, the figure 5 Shows the plaintext and its encrypted image, by comparing the two images, it is found that there is no visible information in the encrypted image on the plaintext image. However, the visual check is not convincing enough to judge the performance of a system. For this, we will introduce some statistical tests to prove the quality of our encrypted image.

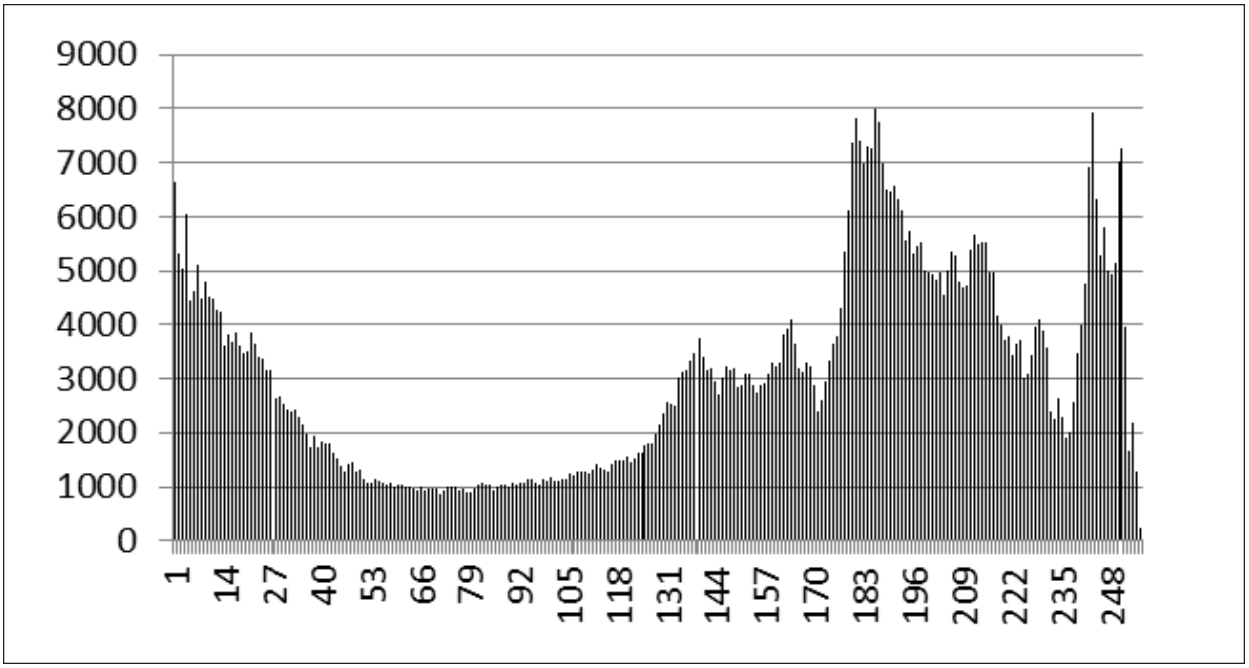


(a) (b)

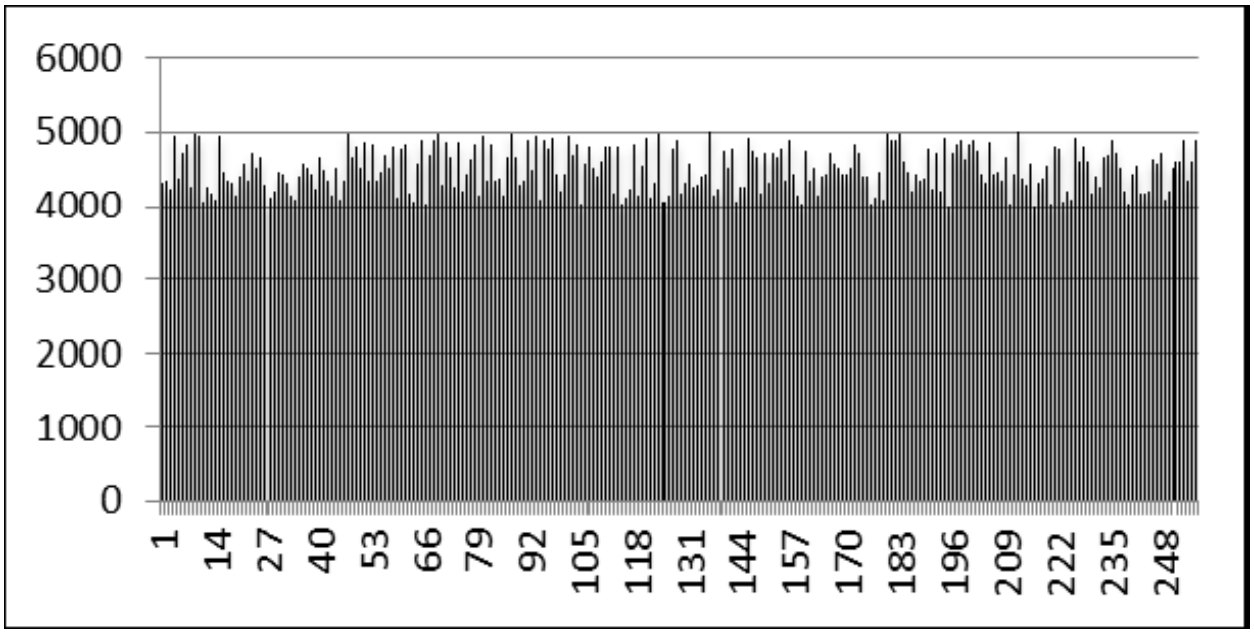
Figure 5. (a) Original image of penguins. (b) Encrypted Image

5.2.2 Histograms

The histogram of an image shows the distribution of gray level pixels, fig. 6 (a) illustrates the pixels distribution of the plain image, this distribution depends of course on the content of the image, figure 6 (b) shows the distribution of the gray level pixels of the encrypted image, it is found that the distribution is uniform and far from being similar to the histogram (a). This shows a good statistical property which stipulates that all pixels of the encrypted image appear with almost the same probability, and also shows that an attacker cannot have information about the plain image (Cipher Image Only Attack), and this implies security against statistical analysis.



(a)



(b)

Figure 6 (a). Histogram of the plain image. (b) Histogram of the cipher image

5.2.3 Correlation

In any normal image, each pixel is highly correlated with its adjacent pixels; either vertically, horizontally, or diagonally. However, the encrypted images must not have this property, because their coefficient of correlation is less than that of the plain image, for this we have calculated [22] the coefficient of correlation of the three levels (horizontal, vertical, and Diagonal) for the three colors

(red, green, and blue). Fig. 7 shows the graph of correlations of the plain image, it can be seen that the coefficient of correlation is regular in the three color levels and varies between 0.97 and 0.98, very close to 1. As for Fig. 8, which shows the correlation graphs of the encrypted image, it can be seen that the correlation coefficients are less inferior to the preceding ones.

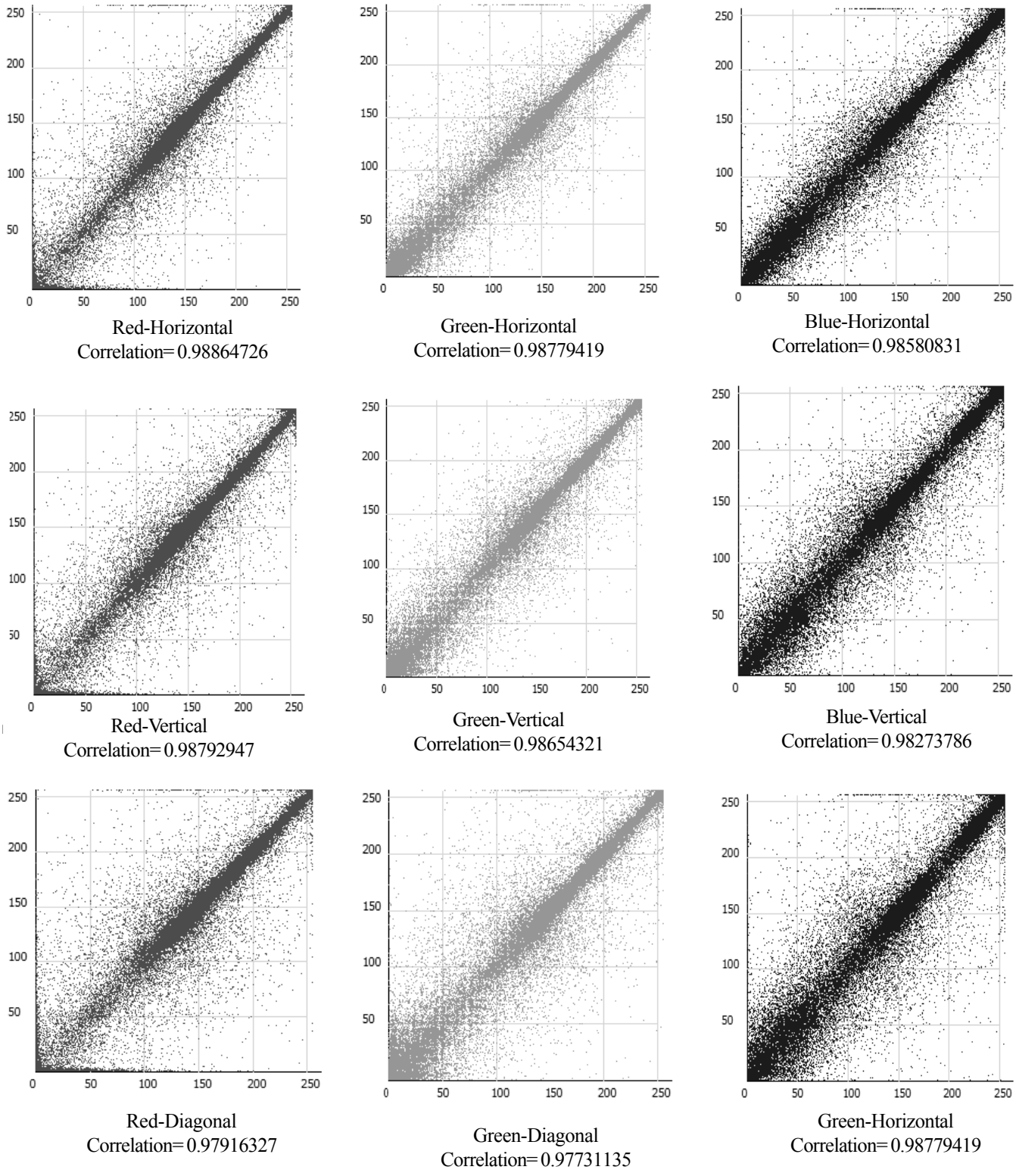
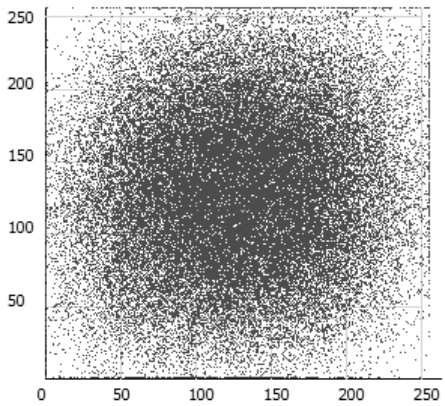
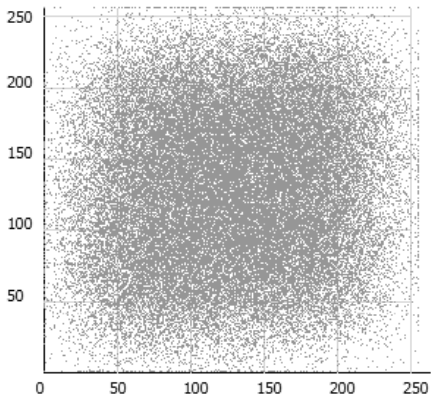


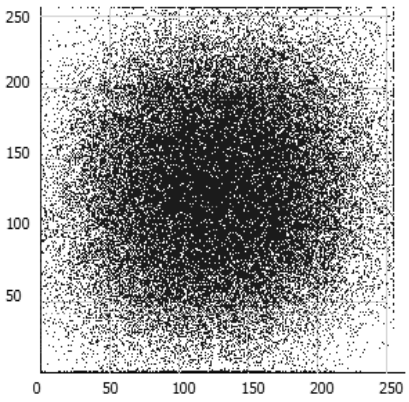
Figure 7. Correlation graph of plain image



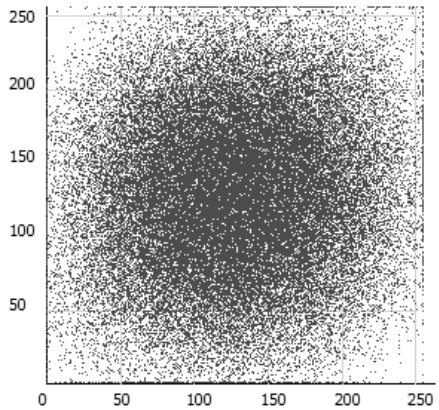
Red-Horizontal
Correlation= 0.0061599875



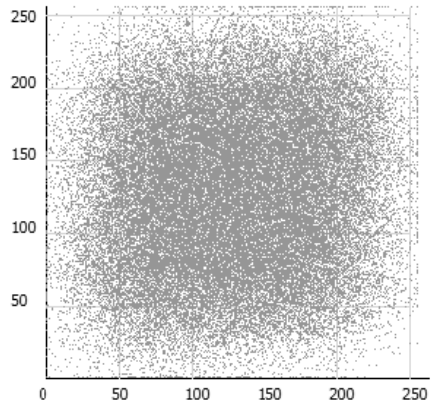
Green-Horizontal
Correlation= 0. 0061645798



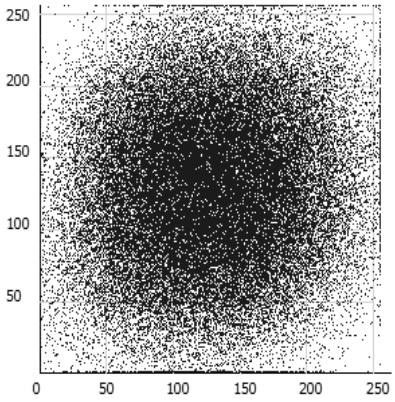
Blue-Horizontal
Correlation= 0. 0062037008



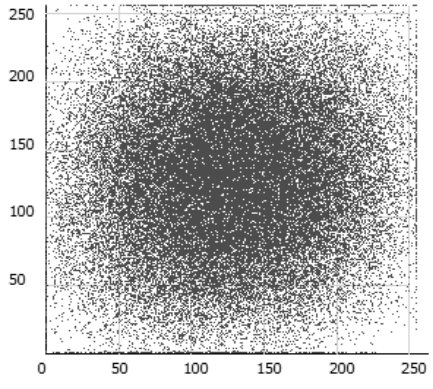
Red-Vertical
Correlation= 0. 0062078990



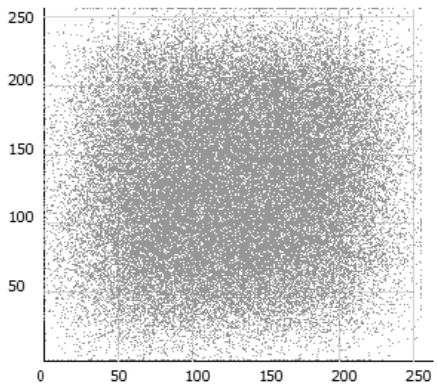
Green-Vertical
Correlation= 0. 0061635988



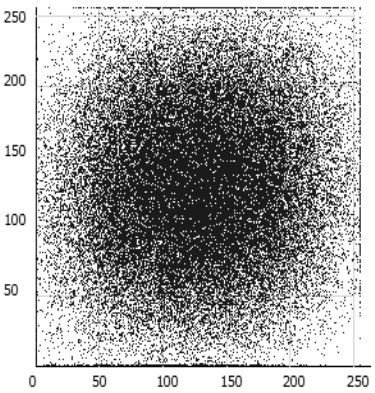
Blue-Vertical
Correlation= 0. 0062482608



Red-Diagonal
Correlation= 0. 0061829164



Green-Diagonal
Correlation= 0. 0060739116



Blue-Diagonal
Correlation= 0. 0061848903

Figure 8. Correlation graph of cipher image

5.2.4 Information Entropy

Entropy is a randomness measure in information theory; it is calculated using the following equation:

$$H = -\sum_{i=0}^{2m-1} p_i Log_2(p_i)$$

where p is the probability distribution of different gray level values of an image (from 0 to 255). High entropy indicates a high degree of randomness so the high value entropy of a message coded on m bits is m. since color is coded on 24 bits/pixel (8 bits for each color), then the optimal entropy value for each color is 8 in ciphered images. Table 1 shows the different entropy values for plain and ciphered images, it is clear that ciphered image has an entropy value close to optimal, and consequently, this implies that the encryption scheme is secure against the entropy based attack.

Plain image			Ciphered image		
Red	Green	Blue	Red	Green	blue
7.1326	7.0915	7.0726	7.9907	7.9890	7.9800

Table 1. Entropy of plain/cipher image for the proposed cryptosystem

6. Comparison of Our Approach With Some of the Existing Approaches

The cryptosystem that we have just proposed is original in its conception, as long as it combines two completely different objects in the construction of the pseudo-random number generator, namely elliptic curves and cellular automata, and other approaches have been proposed in this sense as [20]. As our cryptosystem proved its security against statistical cryptanalysis, and showed

Approaches	**Entropy**	**Horizontal**	**Vertical Diagonal**		
			Correlation Coefficient	**Correlation Coefficient**	**Correlation Coefficient**
Proposed approach	7.9865		0.0061	0.0062	0.0061
[20]	7.9973		0.0010	0.0017	0.0125
AES	7.91		0.046	0.066	0.056
AES + W7	8		0.02	0.03	0.025
AES + A5/1	7.96		0.056	0.077	0.067
Chaos Based	7.92		0.0308	0.0304	0.0317
[19] proposed scheme 1	7.9331		-0.0037	0.0032	0.0055
[19] proposed scheme 2	7.9520		0.0013	0.0044	0.0080
[19] proposed scheme 3	7.9718		0.0031	0.0029	7.59e-005
[19] proposed scheme 4	7.9966		7.6340e-004	0.0045	-4.24e-005
[19] proposed scheme 5	7.9915		4.9460e-005	-7.20e-004	-0.0011
[19] proposed scheme 6	7.9964		-7.98e-004	-0.0013	-0.0046
[19] proposed scheme 7	7.9997		0.0030	0.0030	0.0027
[19] proposed scheme 8	7.9996		-0.0027	-0.0028	0.0026

Table 2. Comparison of results in terms of correlation and entropy

good encryption results. The comparative study deals with the values of the correlation coefficients and the entropy to compare the performances of our proposed approach with those proposed and compared in [20], [19]. Table 2 shows the comparison results in terms of correlation and entropy.

From the values shown in the table 2, it appears that our approach presents a concurrent performance to the other approaches.

7. Conclusion

In this article, a new encryption system has been proposed, which is based on the combination of elliptic curves. The objects that are purely mathematical and cellular automata. They are dynamic systems that have proven their performance in generating pseudorandom numbers. Indeed, the designed generator uses the coordinates of a point randomly generated, and matches them to amatrix built by the transitions of cellular automata. The efficiency of the generator is proven, since one has to resolve the ECDLP. Encryption was tested, and the results were performing well. In the future, we propose to use of programmable cellular automata, and they have recently shown remarkable efficacy in the field of PRNG. Also we propose the use of our pseudo-random numbers in the construction of hash functions schemes.

References

[1] Koblitz, N. (1987). *A Course in Number Theory and Cryptography*, Springer-Verlag.

[2] Morain, F. (1990). *Building cyclic elliptic curves modulo large primes*. LNCS. 547. 328–36.

[3] Koblitz, N. (1987). *Elliptic curve cryptosystems*, Mathematics of Computation 48, 203-209.

[4] Wolfram. S. (1983). *Statistical mechanics of cellular automata*. *Reviews of modern physics*.

[5] John Von Neumann. (1987). *The World of Physics: A small library of the literature of Physics from antiquity to the present,* The General and Logical Theory of Automata, New York.

[6] Stephen Wolfram. (2002). *A new kind of science*. Wolfram Media.

[7] Gong, G., Berson, T. A., Stinson, D. R. (1999). *Elliptic curve pseudorandom sequence generators*, Proceedings of the 6th Annual International Workshop on Selected Areas in Cryptography, LNCS 1758, p. 34-48.

[8] Beelen, P., Doumen, J. M. (2002) *Pseudorandom sequences from elliptic curves.* In: 6th International Conference on Finite Fields with Applications to Coding Theory, Cryptography and Related Areas, Oaxaca, Mexico p. 37-52.

[9] Kaliski, B. (1986). *A pseudorandom bit generator based on elliptic logarithms*. *In*: Advances in Cryptology-CRYPTO'86, LNCS 293, *p* 84-103. Springer-Verlag, 1986.

[10] Kaliski, B. (1988). *Elliptic curves and cryptography*. PhD thesis, M.I.T., 1988.

[11] Bayer, H. (2003). *A fast java implementation of a provably secure pseudo random bit generator based on the elliptic curve discrete logarithm problem*, Inst. für Theoretische Informatik.

[12] Lap-Piu Lee., Kwok-Wo Wong. (2004). *A Random Number Generator Based on Elliptic Curve Operations*, *Computers and Mathematics with Applications,* 47, 217-226.

[13] Barker, E., Kelsey, J. (2005). *Recommendation for random number generation using deterministic random bit generators*, December, NIST Special Publication (SP) 800-90.

[14] Laszlo Merai. (2012). *Remarks on pseudorandom binary sequences over elliptic curves*, August 2012, Fundamenta Informaticae - Cryptology, *In:* Progress: 10th Central European Conference on Cryptology, 114 (3-4) 301-308.

[15] Jilna Payingat., Deepthi, P., Pattathil. (2015). *Pseudorandom Bit Sequence Generator for Stream Cipher Based on Elliptic Curves*, Mathematical Problems in Engineering, vol. 2015, Article ID 257904, 16 p.

[16] Reyad, O., Kotulski, Z. (2016). *Pseudo-random sequence generation from elliptic curves over a finite field of characteristic 2*, Federated Conference on Computer Science and Information Systems (FedCSIS), Gdansk, 2016, p. 991-998.

[17] Laszlo Merai, (2017). Predicting the elliptic curve congruential generator*, Appl. Algebra Eng. Commun. Comput*. 28, (3),

193-203.

[18] Morain, F. (1990). *Building cyclic elliptic curves modulo large primes*, LNCS 547, pp. 328-336.

[19] Sathyanarayana, S. V., Aswatha Kumar, M., Hari Bhat, K. N. (2011). Symmetric Key Image Encryption Scheme with Key Sequences Derived from Random Sequence of Cyclic Elliptic Curve Points, *International Journal of Network Security*, 12 (3) 137-150, May.

[20] Abdellatif, A. A., Xiamu Niu. (2013). *A hybrid chaotic system and cyclic elliptic curve for image encryption*, *Int. J. Electron. Commun.* (AEU) 67, 136-143.

[21] Ziad, E., Dawahdeh,, Shahrul, N.,Yaakob., Rozmie Razif bin Othman, A new image encryption technique combining Elliptic Curve Cryptosystem with Hill Cipher, *Journal of King Saud University* – Computer and Information Sciences (2017).

[22] Abdo, AA., Lian, S G., IsmailI, A., Amin, M., Diab, H. (2013). *A cryptosystem based on elementary cellular automata*. *Commun Non linear Sci Numer Simul.* 18 (1) 136–47.