Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090105

On Pseudo-Random Number Generators Using Elliptic Curves and Chaotic Systems

Omar Reyad[1,2,*] *and Zbigniew Kotulski*[1]

[1] Faculty of Electronics and Information Technology, Warsaw University of Technology, Poland
[2] Faculty of Science, Sohag University, Egypt

Received: 7 Mar. 2014, Revised: 7 Jun. 2014, Accepted: 8 Jun. 2014
Published online: 1 Jan. 2015

Abstract: Elliptic Curve Cryptography (ECC) is a relatively recent branch of cryptography which is based on the arithmetic on elliptic curves and security of the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP). Elliptic curve cryptographic schemes are public-key mechanisms that provide encryption, digital signature and key exchange capabilities. Elliptic curve algorithms are also applied to generation of sequences of pseudo-random numbers. Another recent branch of cryptography is chaotic dynamical systems where security is based on high sensitivity of iterations of maps to initial conditions and parameters. In the present work, we give a short survey describing state-of-the-art of several suggested constructions for generating sequences of pseudorandom number generators based on elliptic curves (ECPRNG) over finite fields of prime order. In the second part of the paper we propose a method of generating sequences of pseudorandom points on elliptic curves over finite fields which is driven by a chaotic map. Such a construction improves randomness of the sequence generated since it combines good statistical properties of an ECPRNG and a CPRNG (Chaotic Pseudo-Random Number Generator). The algorithm proposed in this work is of interest for both classical and elliptic curve cryptography.

Keywords: Elliptic Curve Cryptography, Random Number Generator, Chaotic Maps

1 Introduction

Recently, elliptic curve cryptography (ECC) has received great interest from cryptographers, mathematicians, and computer scientists around the world [1,2]. The primary reason for this is its high security over existing public key cryptographic algorithms. The best algorithm known for solving the underlying mathematical problem of ECC, referred to as the elliptic curve discrete logarithm problem (ECDLP), takes full exponential time. On the contrary, sub-exponential time algorithms are known for tackling the integer factorization and the discrete logarithm problems that RSA and DSA are relied on [3,4]. This implies that the algorithms for solving the ECDLP become infeasible much more rapidly as the problem size increases than those algorithms for tackling the integer factorization and the discrete logarithm problems. For this reason, ECC offers a security level equivalent to RSA and DSA while using a far smaller key size [1].

On the other hand, the security of most cryptographic systems depends upon the generation of unpredictable quantities that must be of sufficient size and randomness. Taking ECC as an example, we need to generate random bits in order to create random curves and the large secret integer [1,2]. This implies that we usually need to implement a random number generator in a cryptographic system.

However, sources of truly random integers are hard to use in practice. It is therefore common to search for pseudo-random number generators (PRNG). Roughly speaking, a pseudo-random source may not be distinguished from a truly random source by any polynomial time algorithm. Several PRNG have been proposed which are using the form of elliptic curves such as [5]. Since [6] methods, different approaches for extracting pseudo randomness from elliptic curves (ECPRNG) have been proposed by [7,8,9].

As we already remarked, the great advantage of elliptic curve cryptography is operating over small-size finite fields (comparing other public-key cryptosystems). However, in case of PRNG small finite fields imply short period of a generator. Therefore, to increase the period of a generator working on an elliptic curve (EC) we propose to combine it with a chaotic dynamical system.

* Corresponding author e-mail: ormak4@yahoo.com

Chaotic dynamical systems are another recent branch of cryptography. Its security is based on high sensitivity of iterations of maps to initial conditions and parameters. The idea of application of discrete dynamical systems for constructing cryptosystems have been presented in [10] where the authors proposed using chaotic maps' parameters as a secret key. Their system was instantaneously broken [11] but an improved cryptosystem [12] with an initial condition of the chaotic dynamical system playing the role of a secret key still remains secure. Recent years such cryptosystems were extensively studied [13] with large variety of particular algorithms and applications. Among them Chaotic Pseudo-Random Number Generators (CPRNG) initiated in [14] found many effective implementations [15] since their period is (by theory) infinite.

In this paper we propose a new method of generating sequences of pseudorandom points on elliptic curves over finite fields which is driven by a chaotic map. Such a construction increases randomness of the sequence generated and makes its period (theoretically) infinite since it combines positive properties of an ECPRNG and a CPRNG. After transformation of the points into binary numbers it can be used for any cryptographic applications.

The organization of the rest of the paper is as follows. In Section 2, the background of discrete dynamical systems and the construction of CPRNG are discussed in Subsections 2.1, 2.2 respectively. In Subsections 2.3 and 2.4 we discuss EC over finite fields and describe its ECPRNG construction. The proposed random number generator will be described in Section 3. Periods of the proposed generator are analyzed in Section 4, while the test results are reported in Section 5. In Section 6, discussions and conclusions are made.

2 Preliminaries

2.1 Discrete Dynamical Systems

A discrete dynamical system is a pair (S,Φ), where S is the state space (usually metric space) and $(\Phi : S \to S)$ is a measurable map which is the generator of the semigroup of iterations [16]. The trajectory starting from the initial state s_0 is the sequence $(s_0)_{i=0}^{\infty}$ of elements of S obtained by iteration

$$s_{i+1} = \Phi(s_i), i = 0,1,2,.... \tag{1}$$

For our purpose of generating pseudo-random sequences we assume several properties of discrete dynamical systems. The most important in our construct is chaos which means strong (exponential) sensitivity of trajectories to changes of initial state and/or system's parameters. Among many known formal definitions of chaos [17] the most popular is that using Lyapunov exponents. Thus, a nonlinear dynamical system is chaotic in some region if for almost all points s in this region (with respect to some Lebesgue invariant measure μ) it has positive Lyapunov exponents. Chaos in a dynamical system makes the trajectories very unstable; starting from two very close initial points, after several iterations we come to quite different final states, what makes the system unpredictable.

Two other properties of the dynamical system make distribution of its iterated states uniform over the state space. We say that the dynamical system (S,Φ) is ergodic [16] if and only if it has only trivial invariant sets, i.e., either $\mu(B) = 0$ or $\mu(S\backslash B) = 0$, whenever the subset B of the space S is measurable and Φ- invariant (the invariance of B means that $\Phi(B) \subset B$).

Due to ergodicity property, the space S cannot be divided into invariant disjoint parts, nontrivial with respect to the measure μ. Thus, a trajectory starting from any point $s_0 \in S$ never localizes in a smaller region and inversely, knowing the final state of the dynamical system, one cannot point out the region (smaller than S) where the trajectory started. A property stronger than ergodicity is mixing property. The dynamical system (S,Φ) is mixing [16] if for each two sets $A,B \in \sigma(S)$,

$$\lim_{i\to\infty} \mu\left(\Phi^{-i}(A) \cap B\right) = \mu(A)\,\mu(B), \tag{2}$$

where $\Phi^{-i}(A)$ is the pre-image of the set A under the i-th iteration of Φ. This formula shows that iterations of Φ make each set A (asymptotically) statistically independent from B. This means that the trajectory starting at a fixed point $s_0 \in S$, after iterations, reaches any region of the space S with the same probability. It is also useful for proving (asymptotic) statistical independence of states of trajectories of discrete dynamical systems.

2.2 Construction of CPRNG

Consider the dynamical system defined in (1) on the state space S and assume that μ is a normalized invariant measure of the system, equivalent to a Lebesgue measure. The idea of construction of CPRNG is to divide the state space S, $\mu(S) = 1$, into two disjoint parts S_0, S_1 such that $\mu(S_0) = \mu(S_1) = 1/2$. As a seed we shall consider an initial point $s \in S' \subseteq S$, where S' is the set of acceptable seeds (for most systems, $\mu(S') = 1$). To obtain a pseudorandom sequence of bits we observe the iterations of the system governed by the map Φ starting from s, i.e., the sequence $s_i := \Phi^i(s)$. Assume that the i-th bit $b_i(s)$ of the generated pseudo-random sequence is equal to "0" if $s_i \in S_0$, and is equal to "1" otherwise, so as a result of iterations we obtain the infinite sequence of bits $G(s)$. Finally, we obtain the map

$$G : S' \to \prod_{i=1}^{\infty} \{0,1\}, \tag{3}$$

such that

$$G(s) = \{b_i(s)\}_{i=1,2,\ldots} = \{b_1(s), b_2(s), \ldots\}, \qquad (4)$$

and where $\prod_{i=1}^{\infty}\{0,1\}$ is the Cartesian product of the infinite number of the two-element set $\{0,1\}$.

In the paper [18] it was proven that if the discrete dynamical system (1) is chaotic, ergodic and it satisfies the mixing property (2) (which is stronger than ergodicity), then the CPRNG defined in (3) and (4) has the fundamental required properties of PRNG:

– unique dependence of the sequence (4) from the seed s,
– equiprobable occurrence of "0" and "1" in the sequence (4),
– asymptotic statistical independence of bits.

Moreover, theoretically the period of such a CPRNG is infinite, since it is iterated over the infinite state space S.

In many practical applications for constructing CPRNG we assume that $S = [0, 1]$ is the interval, $S_0 = [0\,,\,0.5]$, $S_1 = (0.5\,,\,1]$ are two subsets of the measure equal 0.5 and $\Phi : [0, 1] \to [0, 1]$ is a chaotic map with positive Lyapunov exponent λ. However, in concrete implementations of such a CPRNG we must check properties of the particular chaotic map chosen, especially if its invariant measure is really symmetric over the two sub-intervals S_0 and S_1.

2.3 Elliptic Curves over Finite Fields

For a prime p let us denote by F_p is the finite field of p elements. Let E be an elliptic curve over F_p, $p > 3$, given by an affine Weierstrass equation of the form

$$E : y^2 = x^3 + ax + b \qquad (5)$$

with coefficients $a, b \in F_p$, such that $4a^3 + 27b^2 \neq 0$. We recall that the set $E(F_p)$ of F_p-rational points on any elliptic curve E forms an Abelian group (with a point at infinity denoted by O as the neutral element) and the cardinality of this group satisfies the Hasse-Weil bound

$$\left|\#E(F_p) - p - 1\right| \leq 2\sqrt{p} \qquad (6)$$

Point addition and point doubling are the basic EC operations. Point multiplication on EC requires scalar multiplication operation. Let P be a point with the coordinates x, y on an EC, and one needs to compute kP, where k is a positive integer. This scalar multiplication can be done by a series of doubling and addition of P. For example, given $k = 13$ entails the following sequence of operations, by which the efficiency of the scalar multiplication of the points is improved, see Table 1.

Table 1: Scalar multiplication of points of EC

P	$2P$	$3P$	$6P$	$12P$	$13P$
	Doubling	Addition	Doubling	Doubling	Addition

Let us start with $P = (x_1, y_1)$ where $P \neq -P$. To determine $2P = (x_3, y_3)$, P is doubled, use the following equation, which is a tangent to the curve at point P.

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1 \qquad (7)$$

and

$$y_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3) - y_1 \qquad (8)$$

To determine $3P$, addition of points P and $2P$ is used, treating $2P = Q$. Here, P has coordinates $P = (x_1, y_1)$. $Q = 2P$ has coordinates $Q = (x_2, y_2)$, where $P \neq \pm Q$. Then $P + Q = (x_3, y_3)$, where

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 \qquad (9)$$

and

$$y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1 \qquad (10)$$

Therefore, doubling and addition are applied depending on a sequence of operations determined for k. Every point (x_3, y_3) evaluated by doubling or addition is an affine point (points on the EC). Observe that dividing one element by another is multiplication by the inverse of that element in F_p . For this and some other general properties of elliptic curves see [22,23].

2.4 Construction of ECPRNG

2.4.1 Linear Congruential Generator on EC, EC-LCG

For a given point $G \in E(F_p)$, the EC-LCG is defined as the sequence:

$$U_i = G \oplus U_{i-1} = iG \oplus U_0, i = 1, 2, \ldots \qquad (11)$$

where $U_0 \in E(F_p)$ is the "initial value". The EC-LCG generator has been suggested in [24] and then studied in a number of papers [9,25,26].

2.4.2 Power Generator on EC, EC-PG

For a given point $G \in E(F_p)$ and an integer $e \geq 2$, the EC-LCG is defined as the sequence:

$$U_i = eU_{i-1} = e^i G, i = 1,2,... \tag{12}$$

where $U_0 \in E(F_p)$ is the "initial value".

The EC-PG generator has been introduced and studied in [27], see also [28].

2.4.3 Some Other Constructions

We note that after [6], there have been several other suggestions and approaches to extracting pseudo randomness from elliptic curves, see also [7,8,29,30]. However, these methods and results have a slightly different focus and we do not discuss them in this paper.

3 The Proposed Random Number Generator

For a given point $G \in E(F_p)$, we can define the sequence:

$$U_i = i(1+b_i)G \oplus U_0 = \begin{cases} iG \oplus U_0 \text{ if } b_i = 0 \\ 2iG \oplus U_0 \text{ if } b_i = 1 \end{cases}, i = 1,2,... \tag{13}$$

where $U_0 \in E(F_p)$ is the "initial value" and b_i is a binary sequence generated by the chaotic map Φ

$$b_i = \begin{cases} 0 \text{ if } \Phi^i(s) \in S_0 \\ 1 \text{ if } \Phi^i(s) \in S_1 \end{cases}, i = 1,2,... \tag{14}$$

Using EC point sequence U_i and by converting the x,y coordinates of each point $U_i(x,y)$ into binary format we can obtain the bit sequence B_i by applying the following map

$$B_i = U_i(x,y) = \begin{cases} U_{2x2}(x,y) \\ U_{3x3}(x,y) \end{cases}$$

This map takes the two right-most bits from x coordinate and the two right-most bits from y coordinate which denoted $U_{2x2}(x,y)$. Analogously, by taking the three right-most bits from x coordinate and y coordinate which denoted $U_{3x3}(x,y)$ we can obtain another bit sequence. This generalization can also be used in the case of the EC-PG generator mentioned in Subsection 2.4.2.

Example

Consider the curve $E : y^2 = x^3 + x + 4$ over F_{11} . This curve has order 9 and is cyclic. Here $p = 11$. Let $G = (2,5)$ be a point on E and choose $U_0 = (0,2)$ as the initial value. The EC points U, together with the bit sequence B in the two cases, are listed in Table 2.

Table 2: An example of transforming EC points into binary sequences

i	$U_i(x,y)$	$U_i(x,y)_2$	$B_i(U_i)_{2x2}$	$B_i(U_i)_{3x3}$
1	(9,4)	(1001,0100)	(01,00)	(001,100)
2	(3,1)	(0011,0001)	(11,01)	(011,001)
3	(9,7)	(1001,0111)	(01,11)	(001,111)
4	(2,5)	(0010,0101)	(10,01)	(010,101)
5	(0,9)	(0000,1001)	(00,01)	(000,001)
6	(2,6)	(0010,0110)	(10,10)	(010,110)
7	(0,9)	(0000,1001)	(00,01)	(000,001)
8	(2,5)	(0010,0101)	(10,01)	(010,101)
9	(0,2)	(0000,0010)	(00,10)	(000,010)

4 Period Analysis

Define the set τ_p of all triples (a,b,G), where a and b are the parameters of the EC (5) and G is some point of this EC. For any prime $p \geq 5$ and for any $\delta > 0$ and $\varepsilon > 0$, the number of triples $(a,b,G) \in \tau_p$, such that the period T of the sequence generated in equation (15),

$$U_i = G \oplus U_{i-1} = iG \oplus U_0, \tag{15}$$

satisfies the inequality (16),

$$T < p^{1-\delta} \tag{16}$$

is at most

$$O(\#\tau_p p^{-2\delta/3+\varepsilon}), \tag{17}$$

where

$$\#\tau_p = (p^2 + O(p))(p + O(p^{1/2})) \sim p^3. \tag{18}$$

The result presented in Eqs. (17 - 18), showing that typically the period of the sequence (15) is large, has been introduced in [31].

5 Test Results

To ensure good statistical properties (which determine the quality of a generator) of the proposed ECPRNG we assume that the dynamical systems used are also ergodic or preferably mixing. This allows us to use of the well-developed theory of dynamical systems to prove the required statistical properties. Traditionally, extensive statistical testing was used to assess or estimate this quality. Test suites developed for this purpose may be found in [19,20,21]. From these tests we selected 5 which taken together verify random properties of sequences generated. They are:

1. The monobit test (in Tables 3 - 8 named *Frequency Test*), which verifies if the number of "1" bits in the sequence lies within specified limits.

2.The cumulative sums test, which determines whether the cumulative sum of the partial sequences occurring in the tested sequence is too large or too small relative to the expected behavior of that cumulative sum for random sequences. The test has two modes, which are either forward through the sequence or backward through the sequence, named in the Tables *C.Sum (forward)* and *C.Sum (reverse)*, respectively.
3.The runs test (*Runs Test* in the Tables) checking whether the number of runs (the test is carried out for runs of zeros and runs of ones) of length 1, 2, 3, 4 and 5 as well as the number of runs which are longer than 5, each lies within specified limits.
4.The long run test (*Longest Runs Test*) confirming that in the tested sequence there must be no run of length equal to or greater than 34 bits.
5.The discrete Fourier transform test (*DFT*) detecting the periodic features in the tested sequence that would indicate a deviation from the assumption of randomness.

Thus, in these 5 tests, the monobit test verifies if globally the binary distribution is symmetric, the cumulative sums tests check if the sequence is symmetrically growing during bits generation, the runs test and the long run test confirm bits independence and the discrete Fourier transform test allows detecting periodic behavior of the binary sequence generated. Additional motivation for such a choice of such a set of 5 tests (from all 15 tests proposed in the document SP800-22b [20]) is that they can be applied for binary sequences of different size, also very short ones. In our investigations we used sequences of 100, 200, 500, 1000, 2000 bits for the generators constructed on EC over the very small finite field and additionally the sequences of 5000, 10000, 20000 bits for generators on EC over a larger finite field.

The statistical tests made in this paper were on the significance level α equal to 0.01, so the tests are passing if P-value ≥ 0.01,. Moreover, the larger the P-value is, the better the pseudorandom property the generator is.

To investigate the effect of chaotic modulation of the additive ECPRNG we considered three examples of chaotic dynamical systems and two elliptic curves over different-size finite fields. First, we tested random properties of the binary sequences generated by three discrete dynamical systems governed by the following maps:

the Tent Map [32]:

$$s_{i+1} = \Phi(s_i) = \begin{cases} 2s_i \text{ if } s_i < \frac{1}{2} \\ 2(1-s_i) \text{ if } s_i \geq \frac{1}{2} \end{cases}, \qquad (19)$$

the Logistic Map [33]:

$$s_{i+1} = \Phi(s_i) = 4 \cdot s_i (1-s_i), \qquad (20)$$

both for the state space $S = [0,1]$ and $S_0 = [0,0.5]$, $S_1 = (0.5,1]$, and the Chebyshev Map [34]:

$$s_{i+1} = \Phi(s_i) = \cos\left(4\cos^{-1}(s_i)\right), \qquad (21)$$

for the state space $S = [-1,1]$ and $S_0 = [-1,0]$, $S_1 = (0,1]$.

For the tent map (19) we had problems with finding a right initial condition which leads to the pseudorandom sequence with good statistical properties. For the Logistic Map (20) and the Chebyshev Map (21) the statistical tests confirmed good randomness of the binary sequences generated. Therefore we decided to use only the chaotic maps (20) and (21) for our PRNG, testing the sequences generated for certain initial conditions before engaging them into ECPRNG.

In the experiments we used two elliptic curves:

$$E_1 : y^2 = x^3 + x + 4 \qquad (22)$$

over F_{29} and the elliptic curve described by formally the same equation

$$E_2 : y^2 = x^3 + x + 4 \qquad (23)$$

but now over F_{5501}. Results of testing the sequences generated are presented in Tables 3 - 8.

In Table 3 are presented results for the additive ECPRNG on the curve E_1 without chaotic modulation. As it is expected, the generator works correctly for very short binary sequences (200 bits) due to its periodicity, what is indicated by the DFT Test. Including the CPRNG enables generating correctly longer sequences: 2000 bits for the Logistic Map (Table 4) or 1000 for the Chebyshev Map (Table 5). Analogously, for the larger elliptic curve E_2 the non-disturbed ECPRNG gave a correct result till 5000 bits generated, as it is seen from Table 6. The generators driven by the two chaotic maps (20) and (21) give twice as much correct pseudo-random bits, see Tables 7 and 8. For 20000 and more bits the DFT test indicates the generators' periodicity.

6 Conclusions

In this paper we proposed a new construction of a pseudorandom number generator which uses both elliptic curves and discrete dynamical systems for bitstreams generation. As our experiments presented in Section 5 shown such a combination gave us the construction with positive properties being resultant properties of the two components. Comparing purely EC-based pseudorandom number generator, our construction has longer period for a fixed size of the finite field F_p where the EC lives. Thus, we can use smaller fields (with less computational complexity of arithmetic calculations) to obtain a bitstream of a fixed length. Relating the generator proposed to a purely chaotic pseudorandom number generator, now we can obtain more bits in one iteration: instead 1 bit, as it is in the chaotic case, we can have the

Table 3: ECPRNG without chaotic modulation. *P*-values for the case $EC_{2x2}, p = 29$

Test name	Case 1 100 bits	Case 2 200 bits	Case 3 500 bits	Case 4 1000 bits	Case 5 2000 bits
Frequency	1.000000	0.671373	0.371093	0.100097	0.066717
C. Sum (forward)	0.722386	0.939470	0.727622	0.163980	0.133434
C. Sum (reverse)	0.722386	0.574764	0.389118	0.107464	0.083738
Runs	0.317311	0.877383	0.399727	0.397138	0.232001
Longest Runs	1.000000	1.000000	1.000000	1.000000	1.000000
DFT	0.745603	0.168669	**0.000089**	**0.000221**	**0.005837**

Table 4: ECPRNG modulated with the Logistic map. *P*-values for the case $EC_{2x2}, p = 29$

Test name	Case 1 100 bits	Case 2 200 bits	Case 3 500 bits	Case 4 1000 bits	Case 5 2000 bits
Frequency	0.841481	0.887537	0.591505	0.113846	0.152406
C. Sum (forward)	0.958638	0.973049	0.917914	0.227688	0.245709
C. Sum (reverse)	0.999430	0.998656	0.727622	0.133272	0.214807
Runs	0.315185	0.202545	0.277117	0.055114	0.010298
Longest Runs	1.000000	1.000000	1.000000	1.000000	1.000000
DFT	0.745603	0.358795	0.110478	0.411770	0.110478

Table 5: ECPRNG modulated with the Chebyshev map. *P*-values for the case $EC_{2x2}, p = 29$

Test name	Case 1 100 bits	Case 2 200 bits	Case 3 500 bits	Case 4 1000 bits	Case 5 2000 bits
Frequency	0.548506	0.322199	0.210498	0.036879	0.089242
C. Sum (forward)	0.540731	0.638440	0.279973	0.049508	0.103459
C. Sum (reverse)	0.897326	0.574764	0.389118	0.068227	0.170221
Runs	0.333303	0.521555	0.802748	0.158895	**0.001870**
Longest Runs	1.000000	1.000000	1.000000	1.000000	1.000000
DFT	0.104757	0.646355	0.110478	0.150897	0.110478

Table 6: ECPRNG without chaotic modulation. *P*-values for the case $EC_{3x3}, p = 5501$

Test name	Case 1 100 bits	Case 2 200 bits	Case 3 500 bits	Case 4 1000 bits	Case 5 2000 bits	Case 6 5000 bits	Case 7 10000 bits	Case 8 20000 bits
Frequency	1.000000	0.479500	0.371093	0.026857	0.591505	0.977435	0.289145	0.276178
C. Sum (forward)	0.814758	0.704309	0.359311	0.029787	0.075135	0.376714	0.189838	0.176714
C. Sum (reverse)	0.814758	0.405915	0.420651	0.027282	0.245709	0.358107	0.436946	0.452636
Runs	0.045500	0.095740	0.027537	0.134454	0.055256	0.497257	**0.008072**	**0.006788**
Longest Runs	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.031775	**0.000110**
DFT	0.330390	1.000000	0.663355	1.000000	0.468160	0.713570	0.363646	**0.000000**

Table 7: ECPRNG modulated with the Logistic map. *P*-values for the case $EC_{3x3}, p = 5501$

Test name	Case 1 100 bits	Case 2 200 bits	Case 3 500 bits	Case 4 1000 bits	Case 5 2000 bits	Case 6 5000 bits	Case 7 10000 bits	Case 8 20000 bits
Frequency	0.689157	0.571608	0.474274	0.849515	0.152406	0.350623	0.952156	0.630635
C. Sum (forward)	0.814758	0.892023	0.256747	0.458362	0.292175	0.491372	0.602017	0.697670
C. Sum (reverse)	0.722386	0.704309	0.685633	0.328147	0.045122	0.282680	0.549275	0.452636
Runs	0.987214	0.150110	0.171959	0.612070	0.999143	0.765048	0.262729	0.010401
Longest Runs	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.653236	0.213793
DFT	0.104757	0.646355	0.884636	0.837419	0.081659	0.358795	0.398920	**0.000000**

Table 8: ECPRNG modulated with the Chebyshev map. *P*-values for the case EC_{3x3}, $p = 5501$

Test name	Case 1 100 bits	Case 2 200 bits	Case 3 500 bits	Case 4 1000 bits	Case 5 2000 bits	Case 6 5000 bits	Case 7 10000 bits	Case 8 20000 bits
Frequency	0.161513	0.571608	0.591505	0.751830	0.395489	0.671373	0.968093	0.954889
C. Sum (forward)	0.322973	0.458043	0.563698	0.850473	0.685633	0.612629	0.796727	0.828133
C. Sum (reverse)	0.032790	0.405915	0.644038	0.562079	0.374000	0.881166	0.832407	0.875593
Runs	0.065759	0.981919	0.778427	0.901837	0.741919	0.911937	0.327094	0.336229
Longest Runs	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.653236	0.213793
DFT	0.745603	1.000000	0.884636	0.837419	0.663355	0.462869	0.037854	**0.000000**

number of bits that is of the range twice as much a the binary size of the modulus p of the finite field F_p, what slightly increases the speed of generation.

The experiments presented in this paper confirm that our theoretical assumptions concerning the new construction of the PRNG are satisfied. However, to optimize the procedures of generation further extensive studies must be performed. One possible extension is generating bits using elliptic curves over binary finite fields, to omit the operation of decoding points of the elliptic curve into binary sequences. Next, we should find better method of establishing parameters of chaotic generators to avoid prior testing of their random properties, what would allow us to make all operations on-line. Such research will be the subject of our further studies.

Acknowledgment

The first author acknowledges the financial support by the Ministry of Higher Education of Egypt which made possible work on this paper.

References

[1] Certicom Corp, http: //www. certicom. com/.
[2] A. Menezes, Elliptic Curve Public Key Cryptosystems, Kluwer Academic, (1993).
[3] J. Crowie, B. Dodson, R. Elkenbracht-Huizing, A. Lenstras, P. Montgomery, J. Zayer, A world wide number field sieve factoring record: On to 512 bits, In Advances in Cryptology-ASIACRYPT '96, pp. 382-394, Springer-Verlag, (1996).
[4] D. Gordon, Discrete logarithms in GF(p) using the number field sieve, SIAM J. Discr. Math. 6, 124-138, (1993).
[5] D. Jao, D. Jetchev, R. Venkatesan, On the bits of elliptic curve diffie-hellman keys. In: Srinathan, K., Rangan, C.P., Yung, M. (eds.) INDOCRYPT 2007. LNCS, vol. 4859, pp. 33–47. Springer, Heidelberg (2007)
[6] B.S. Kaliski, One-way permutations on elliptic curves, Journal of Cryptology 3, pp. 187 - 199 (1991-1992)
[7] Farashahi, R.R., Schoenmakers, B., Sidorenko, A., Efficient pseudorandom generators based on the DDH assumption. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 426–441. Springer, Heidelberg (2007)
[8] M. Caragiu, R.A. Johns, J. Gieseler, Quasi-random structures from elliptic curves. J.Algebra, Number Theory Appl. 6, 561–571, 2006.
[9] G. Gong, T.A. Berson, D.R. Stinson. Elliptic curve pseudorandom sequence generators. Selected areas in cryptography (Kingston, ON, 1999), pages 34-48. Springer, Berlin, 2000
[10] T. Habutsu, Y. Nishio, I. Sasase, S. Mori, A secret key cryptosystem by iterating a chaotic map, Advances in Cryptology - EUROCRYPT '91, LNCS 547, pp.127-140, Springer 1991.
[11] E. Biham, Cryptoanalysis of the chaotic-map cryptosystem suggested at EUROCRYPT'91, Advances in Cryptology - EUROCRYPT '91, LNCS 547, pp.532-534, Springer 1991.
[12] Z. Kotulski, J. Szczepanski, Discrete chaotic cryptography, Annalen der Physik, vol.6, no.5, pp.381-394, 1997.
[13] L. Kocarev, Sh. Lian, (Eds.), Chaos-based Cryptography. Theory, Algorithms and Applications, Series: Studies in Computational Intelligence, Vol. 354, Springer 2011.
[14] T. Kohda, A. Tsuneda. Statistic of chaotic binary sequences. IEEE Transactions on Information Theory 43, no.1: 104-112. 1997.
[15] Z. Kotulski, J. Szczepanski, K. Gorski, A. Paszkiewicz, A. Gorska, On constructive approach to chaotic pseudorandom number generators, Proceedings RCMIS 2000, Vol.1, pp.191-203, October 4-6, Zegrze 2000.
[16] L.P. Cornfeld, S. V. Fomin, and Ya.G. Sinai, Ergodic Theory, Springer-Verlag, Berlin 1982.
[17] R. Brown, L.O.Chua, Clarifying chaos: examples and counterexamples, International Journal of Bifurcation and Chaos vol.6, no. 2, pp. 219-249, 1996.
[18] J. Szczepanski, Z. Kotulski, Pseudorandom number generators based on chaotic dynamical systems, Open Systems & Information Dynamics, Vol.8, No.2, pp.137-146, (2001).
[19] FIPS 140-2, Security Requirements for Cryptographic Modules, NIST, 2000.
[20] A. Rukhin, J. Soto, J. Nechvatal et.al., A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, NIST Special Publication 800-22 with revisions, May 15, 2001.
[21] D.E. Knuth, The Art of Computer Programming - Seminumerical Algorithms, vol. 2, Addison-Wesley, Reading, 1981.
[22] I. Blake, G. Seroussi, N. Smart, Elliptic curves in cryptography, London Math. Soc., Lecture Note Series, 265, Cambridge Univ. Press, 1999.

[23] J.H. Silverman, The arithmetic of elliptic curves, Springer-Verlag, Berlin, 1995.
[24] S. Hallgren, Linear congruential generators over elliptic curves, Preprint CS-94-143, Dept. of Comp. Sci., Cornegie Mellon Univ., 1994, 1 - 10.
[25] P. Beelen, J. Doumen, Pseudorandom sequences from elliptic curves, Finite Fields with Applications to Coding Theory, Cryptography and Related Areas, Springer-Verlag, Berlin, 2002, 37 - 52.
[26] G. Gong, C. C. Y. Lam, Linear recursive sequences over elliptic curves, Proc. Intern. Conf. on Sequences and their Applications, Bergen 2001, Springer-Verlag, London, 2002, 182 - 196.
[27] T. Lange, I.E. Shparlinski, Certain exponential sums and random walks on elliptic curves, Canad. J. Math., 57 (2005), 338–350.
[28] E. El Mahassni and I.E. Shparlinski, On the distribution of the elliptic curve power generator, Proc. 8th Conf. on Finite Fields and Appl., Contemp. Math., vol. 461, Amer. Math. Soc., Providence, RI, 2008, 111 - 119.
[29] H. Hu, L. Hu, D. Feng, On a class of pseudorandom sequences from elliptic curves over finite fields IEEE Trans. Inform. Theory, 53 (2007), 2598 - 2605.
[30] I.E. Shparlinski, Pseudorandom number generators from elliptic curves, Affine Algebraic Geometry, Amer. Math. Soc., 2009, 121 - 142
[31] I.E. Shparlinski, Orders of points on elliptic curves, Affine Algebraic Geometry, Amer. Math. Soc., 2005, 245–252
[32] J.M. Amigo, L. Kocarev, J. Szczepanski, Theory and Practice of Chaotic Cryptography, Physics Letters A, vol. 366, no.3, pp. 211 - 216, 2007
[33] S.C. Phatak, S.S. Rao, Logistic map: A possible random-number generator, Physical Review E, vol.51, no.4, pp.3670-3678, 1995
[34] X.F. Liao, X.M. Li, J. Peng, et al, A digital secure image communication scheme based on the chaotic Chebyshev map, Int. J. Commun. Syst., vol.17, no.5, pp.437-445, 2004

Omar Reyad is a PhD student at the Faculty of Electronics and Information Technology, Warsaw University of Technology, Poland. He received his MSc in Computer Science from Sohag University. His main research interests are in Elliptic curve cryptography and cryptographic protocols.

Zbigniew Kotulski is a Professor and Head of the Security Research Group at the Faculty of Electronics and Information Technology, Warsaw University of Technology, Poland. He received his MSc in Applied Mathematics from Warsaw University of Technology and PhD and DSc degrees from the Institute of Fundamental Technological Research of the Polish Academy of Sciences. He is the author and co-author of five books and over 150 research papers on applied probability, cryptography, cryptographic protocols and network security.