A Random Number Generator Based on Isogenies Operations

He Debiao*, Chen Jianhua, Hu Jin

School of Mathematics and Statistics,Wuhan University, Wuhan 430072, Hubei, China

Abstract*:* A random number generator based on the operation of isogenies between elliptic curves over finite fields *Fp* is proposed. By using the proposed generator together with the isogeny cryptography algorithm, which is against the attack of quantum computer, we can save hardware and software components. Theoretical analyses show that periods of the proposed random number generator are sufficiently long. Moreover, the generated sequences have passed the U.S. NIST statistical test.

Key words: Random Number Generator; Public-key Cryptosystem; Quantum Computer; Isogeny; Elliptic Curve

1. Introduction

Security of the known public public-key cryptosystems is based on two general mathematical problems: determination of order and structure of a finite Abelian group, and discrete logarithm computation in a cyclic group with computable order. Both of the problems can be solved in polynomial time using Shor's algorithm for a quantum computer [1]. Thus, most of the current public-key cryptosystems will become insecure when size of a quantum register is sufficient. Development of key agreement protocols, which would be strong against a quantum computer, is necessary.

A mathematical problem, which is hypothetically strong against the attack of quantum computer, has been proposed [2]. It consists of searching for an isogeny (an algebraic homomorphism) between elliptic curves over a finite field. The problem is a special case of morphism computation in an Abelian groups category.

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 elliptic curve cryptosystem [3] and isogeny cryptosystem [2] as examples, we need to generate random bits in order to create random curves and the large secret random number. This implies that we usually need to implement a random number generator in a cryptographic system. A number of random number generators have been proposed [4-7]. However, they are usually not designed together with the cryptographic system and so extra design and implementation effort are required. If both the tasks of random number generation and encryption can be done by using the same software or hardware module, we can save hardware cost, memory space, and design time.

The organization of the rest of the paper is as follows. In Section 2, the background of isogeny will be described. In Section 3, the proposed random number generator will be proposed. Periods of the proposed generator are analyzed in Section 4. The test results are reported in Section 5. In section 6, choice of parameter and implementation are discussed. Conclusions will be made in Sections 7.

*Corresponding author.

E-mail:hedebiao@163.com, *Tel*:+0086015307184927, *Fax*: +008602787817667

2. Elliptic Curves over Fp and Isogeny Star

In this section, we will give some background about prime field F_p and isogeny star. We refer reader to [2] for more knowledge about isogeny star.

Let *E* be a elliptic curve, defined on the finite fields F_p, and it's equation is

$$y^2 = x^3 + ax + b\ ,\quad a,b \in F_p\ . \tag{1}$$

Then the map

$$\pi : (x,y) \to (x^p, y^p) \tag{2}$$

specifies the Frobenius endomorphism of the curve E. A Frobenius map satisfies its characteristic equation

$$\pi^2 - T\pi + p = 0\ , \tag{3}$$

Where $T = p - a - \# E(F_p)$ is the Frobenius trace. Through the Hasse's theorem, we know

$$\text{that } |T| < 2\sqrt{p}\ . \tag{4}$$

So the discriminant D_π of the Frobenius equation (3) satisfies

$$D_\pi = T^2 - 4p < 0\ . \tag{5}$$

Theorem 1. Elliptic curves are isogenous over F_p if and only if they have equal number of points.

Proof. See[2].

Theorem 2. Let an elliptic curve $E(F_p)$ have the Frobenius discriminant D_π and $\left(\frac{D_\pi}{l}\right)$ be a Kronecker symbol for some l-degree isogeny. If $\left(\frac{D_\pi}{l}\right) = -1$, then there are no l-degree isogenies; if $\left(\frac{D_\pi}{l}\right) = 1$, then two l-degree isogenies exist; if $\left(\frac{D_\pi}{l}\right) = 0$, then 2 or *l* + 1 *l*-degree isogenies exist and *l* is called Elkies prime number.

Proof. See[2].

Let $U = \{E_i(F_p)\}$ be a set of elliptic curves with equal number of points, so that each element of U is uniquely determined by a *j*-invariant of an elliptic curve. According to the theorem

1 and the equation (4), we can consider *U* as a category, and the set of isogenies between elements of U as a set of morphisms of this category. We can compute $\#U = h_{D_\pi}$, where h_{D_π} is the degree of Hilbert polynomial [2]. According to [7], we can get $h_{D_\pi} = O(\sqrt{D_\pi})$.

Let *l* is Elkies prime number, we can get that there are two isogenous elliptic curves for any elliptic curves of U, from theorem 2. It is practically determined that, when #*U* is prime, all the elements of *U* form a single isogeny cycle.

Let $l_1 \neq l$ be one more prime isogeny degree with the property that $\left(\frac{D_\pi}{l_1}\right) = 1$. In this case, $l_i -$degree isogenies form a cycle over U as well. Then we can put the $l -$ and $l_i -$degree isogeny cycles over each other. Same can be done for other isogeny degrees of such kind.

Definition 1. A graph, consisted of prime number of elliptic curves, connected by isogenies of degrees satisfying $\left(\frac{D_\pi}{l_i}\right) = 1$, is an isogeny star.

The example of an isogeny star is shown on the figure 1. There are 7 elliptic curves over F_{83} having Frobenius trace is 9. Their j-invariants are noted in the nodes [2].

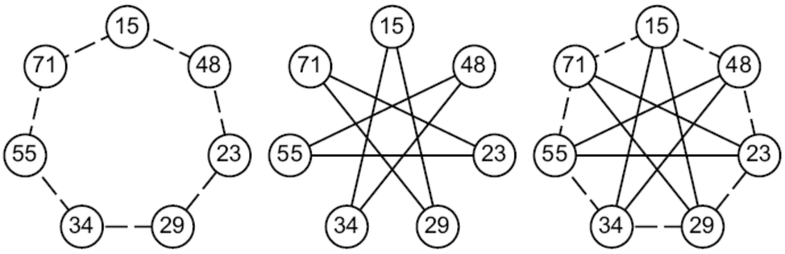

Fig. 1: 3- and 5-degree isogeny cycles, and the isogeny star

If an isogeny star is wide enough, we can use it for crypto algorithm constructing. For that purpose, it is necessary to specify a direction on a cycle and route of isogeny stat. The method for direction determination on an isogeny cycle is mentioned in [4], we don't give the detail here.. It uses impact of Frobenius endomorphism on an isogeny kernel. The definition of isogeny stat is following.

Let S be an isogeny star, d is an positive integer, $L = \{l_i\}, 0 < i \leq d$ is a set of Elkies isogeny degrees being used and $F = \{\pi_i\}, 0 < i \leq d$ is a set of Frobenius eigenvalues, which specify positive direction for every $l_i \in L, 0 < i \leq d$.

Definition 2. A set $R = \{r_i\}, 0 < i \leq d$, where r_i is number of steps by the $l_i -$isogeny in the direction π_i , is a route on the isogeny star.

For example, if we use the clockwise direction on the figure 1, then the route $R = \{2,1\}$, starts from the node 15, follows through 48, 23 and leads to 55. We will denote it by $R(15) = 55$. Obviously, it doesn't matter, in which order we do steps of a route. The latter route can be evaluated by $15 \to 48 \to 34 \to 55$ as well.

We can define the composition [2] of routes $A = \{a_i\}$ and $B = \{b_i\}$ as $AB = \{a_i + b_i\}$.

It’s easy to get that routes are commutative: $AB = BA$.

The computation of iosgeny between elliptic curves can be done using the method in [8-10], we don’t give the detail here.

3. The Proposed Random Number Generator

In this paper we use an elliptic curve E defined over a finite field F_p, whose equation is (1), the parameters is following.

1) F_p : the finite field;

2) E_{init} : an initial elliptic curve, its equation is $y^2 = x^3 + a_{init}x + b_{init}$, $a_{init}, b_{init} \in F_p$;

3) d: number of isogeny degrees being used;

4) $L = \{l_i\}, 1 \le i \le d$: a set of Elkies isogeny degrees being used;

5) $F = \{\pi_i\}$, $1 \le i \le d$:a set of Frobenius eigenvalues, which specify the positive direction for every $l_i \in L$;

6) $k = \{k_1, k_2, \cdots, k_d\}, k_i > 1, (k_i, k_j) = 1$: a set of coprime positive integer, which is the limit for number of steps by one isogeny degree in a route. For any route $\{r_i\}$, numbers of steps are selected in $-k_i \le r_i \le k_i$;

Let E be an elliptic curve defined over a finite field F_p, equation is (1). Let A_E and B_E denote the parameter a and b of the equation of E separately. A block diagram of the proposed random number generator is shown in Figure 2, where $\oplus$ denotes the operation of exclusive or.

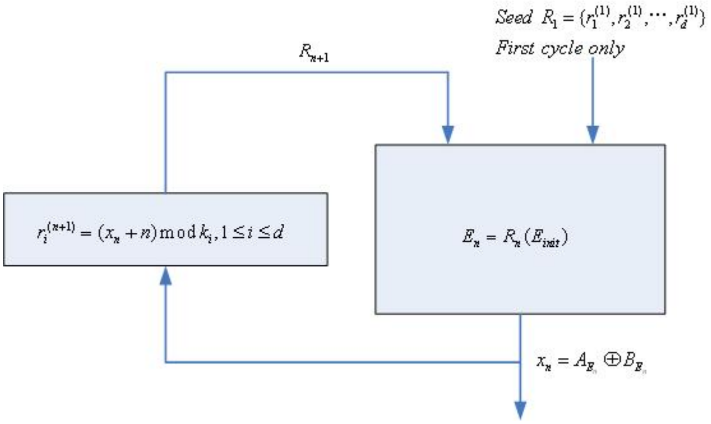
Seed $R_1 = \{r_1^{(1)}, r_2^{(1)}, \cdots, r_d^{(1)}\}$
First cycle only
R_{n+1}
$r_i^{(n+1)} = (x_n + n) \bmod k_i, 1 \le i \le d$
$E_n = R_n(E_{init})$
$x_n = A_{E_n} \oplus B_{E_n}$

Figure 2: Block diagram of the proposed random number generator.

It's easy to see that when these operations are done recursively, a sequence of bits can be obtained by collecting the x_n s.

4. Period Analysis

The purpose of setting $r_i^{(n+1)} = (x_n + n) \bmod k_i, 1 \le i \le d$ is to increase the period of the generator. If n is not added, the bit sequence depends solely on the output of the $R_n(E_{init})$ operation. It will start to repeat itself when there exists an elliptic curve E_{n-i} such that $x_n = A_{E_n} \oplus B_{E_n} = A_{E_{n-i}} \oplus B_{E_{n-i}} = x_{n-i}$.

In the s^{th} cycle,

$$r_i^{(s+1)} \equiv (x_s + s) \bmod k_i, 1 \le i \le d \tag{6}$$

where x_s denote the output of $R_n(E_{init})$, shown in figure 2.

In the t^{th} cycle,

$$r_i^{(t+1)} \equiv (x_t + t) \bmod k_i, 1 \le i \le d \tag{7}$$

where x_t denote the output of $R_n(E_{init})$, shown in figure 2.

Suppose that the output of the $E_n = R_n(E_{init})$ module in the t^{th} cycle is the same as that of the s^{th} cycle.

That is,

$$x_t = x_s \tag{8}$$

If the output of the module of the $(t+1)^{th}$ cycle is also equal to that of the $(s+1)^{th}$ cycle, then $r_i^{(t+1)} \equiv r_i^{(s+1)} \bmod k_i, 1 \le i \le d$. By (7) and (8), we have

$$x_s + s \equiv x_t + t \bmod k_i, 1 \le i \le d \tag{9}$$

By the equations (8) and (9), we can get that

$$s \equiv t \bmod k_i, 1 \le i \le d \tag{10}$$

Then $k_i \mid (s-t), k_i (1 \le i \le d)$.

Because k_i and k_j are coprime when $i \ne j$, we can get that $k_1 \times k_2 \times \cdots \times k_d \mid (s-t)$. Hence the output will repeat after $k_1 \times k_2 \times \cdots \times k_d$ cycles. In order increase the period of the random number generator, we just need to increase the value $k_1 \times k_2 \times \cdots \times k_d$ by increasing the number of k_i .

5. Test Result

The U.S. NIST statistical test suite is used to test the randomness of the generated bits. It includes 15 statistical tests and each of them is formulated to test a null hypothesis that the sequence being tested is random. There is also an alternative hypothesis which states that the sequence is not random. For each test, there is an associated reference distribution (typically normal distribution or χ^2 distribution), based on which a P_values is computed from the binary sequence. If this value is greater than a pre-defined threshold α (0.01 in default), the sequence passes the test. The two approaches that NIST has adopted include the examination of (1) proportion of sequences that pass a statistical test, and (2) uniformity of the distribution of those P_values .

According to [19], if m sequences were tested, the proportion of sequences that passed a specific statistical test should lie above p_α :

$$p_\alpha = (1-\alpha) - 3\sqrt{\frac{\alpha(1-\alpha)}{m}} \tag{11}$$

In our experiment, m = 1000, α = 0.01, and p_α = 98.05%.

To check the distribution of P_values, the interval between 0 and 1 is divided into 10 sub-intervals. The number of P_values in each sub-interval is counted, based on which a $P\_valuesT$ is calculated. If $P\_valuesT$ > 0.0001 holds, the sequences are considered to be uniformly distributed.

The NIST statistical test suite contains 15 tests (the Lempel-Ziv complexity test is removed from the test suite since Version 1.7). For the details of those tests, please refer to [19]. Some tests such as FT, FBT, RT, ST, AET and CST require only 100 bits for each sequence. Other tests, however, require more bits. Specially, the PTMT, LZCT, RET, REVT tests need about 1 M for each sequence.

In our experiment, 100 bit sequence are generated for each five parameters $F_p, E_{init},$ by using 100 pairs of the randomly-selected initial R_1. Because the express of the elliptic and route are very complicated, we just list one case here for reference.

The curve $y^2 = x^3 + ax + b$ is used here and the parameters are $a = -3$,
$b =$ 5AC635D8 AA3A93E7 B3EBBD55 769886BC 651D06B0 CC53B0F6 3BCE3C3E 27D2604B, and the field size is $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$[14].

We let *d* is 256, the set *k* is the first 256 prime, and the set is the first 256 Elkies prime.

R_1={1,0,3,1,8,10,14,2,6,15,16,1,37,8,28,14,34,39,12,22,40,61,69,83,30,93,26,105,98,69,17,
120,15,116,113,57,25,76,41,155,67,174,180,25,161,172,22,12,212,67,182,113,98,100,160,200,42,
16, 260,253,62,104,53,243,170,306,305,103,327,210,126,265,295,138,61,161,10,9,260,169,117,
190, 289,431,116,0,182,427,437,227,82,164,112,125,378,255,276,139,168,488,208,364,400,12,
354, 284,47,515,364,456,440,100,478,334,534,64,426,594,278,473,189,7,626,55,305,605,570,423,
46, 474,564,405,343,266,553,304,769,364,56,647,290,708,545,249,297,656,82,74,282,298,121,
446, 716,361,257,162,410,295,858,459,357,496,506,739,871,698,681,849,798,538,510,830,79,
904, 660,515,665,464,109,121,480,198,598,951,89,835,732,217,1079,1006,516,323,1158,1175,
846, 602,767,424,199,234,90,268,1169,373,701,269,509,513,1156,106,1071,789,821,1302,1046,
397, 59,674,1063,134,102,425,1279,593,980,622,179,529,780,847,1177,318,616,246,1097,1308,
1196,53,420,773,878,733,1385,758,1251,1355,125,1484,1312,686,1302,1593,204,1336,1297,
1149}.

In the test of P_values uniformity for FT and CST, each sequence is set to 1024 bits by concatenating the output of the proposed random generator. Otherwise, the number of different P_values would not be sufficient to carry out the uniformity test. As for other tests which require more than 100 bits for each sequence, we collect the first 256 bit for different initial R_1. The test results can be found in Table 1.

Table 1.Test results for the random number generator

Test name	Proportion	*P_valueT*
FT	0.9901	0.1624
FBT	0.9852	0.3781
CST*	0.9863	0.1639
RT	0.9942	0.1498
LROBT	0.9875	0.1397
AET	0.9846	0.4153
ST*	0.9811	0.5741
RBMRT	0.9973	0.1542
DFTT	0.9918	0.3681
ATMT*	0.9857	0.2456
PTMT	0.9961	0.1572
MUST	0.9921	0.3660
RET*	0.9836	0.2718
REVT*	0.9918	0.2249
LCT	0.9826	0.1152

It is observed from Table 1 that the proposed random number generator pass all the statistical tests, i.e., the passing proportions are greater than 98.05% and $P\_valuesT$ greater than 0.0001. According to [19], we can conclude that the data generated by these two approaches are random.

6. Discussions

6.1 Choice of Parameters

The test results indicate that all the random number sequences generated over the three fields have passed the SP800-22 statistical test and the FIPS 140-2 statistical tests. Therefore, the proposed generator can be accepted as a reliable random number generator for integrating with the isogeny cryptosystem to generate the dynamic private keys.

By simply changing the seed R_1 and the initial curve E_{init}, a different bit sequence can be generated. These two parameters should be kept secret for security. Basically, E_{init} can be any elliptic curve defined in finite fields F_p. However, if we integrate this generator with a isogeny cryptosystem, E_{init} should not be the point used in the isogeny cryptosystem part. This is because those points are always treated as the parameters of public keys which are made public.

If the $\#U = h_{D_\pi}$ is not lager enough, the number, generate by the proposed random number generator, may be guessed correctly. We need to select correct F_p and E_{init}, which can determine the value of h_{D_π}, in order to make h_{D_π} large enough. In practice, the value of h_{D_π} can be determined using analytical methods [18].

6.2 Implementation

Various implement of computation of isogenies between elliptic curves [2, 8, 9, 10]. These algorithms can be used to implement isogeny cryptosystem. Our proposed random number generator is based on the core operations of isogeny cryptosystem, so the proposed random generator can be designed and implemented efficiently using the existing components and the cost of the implementation can be reduced.

7. Conclusion

In this paper, a new approach for constructing a random number generator using operation of isogenies between elliptic curves is presented. Periods of the generator are analyzed theoretically. By the test results, we can draw the conclusion that the quality of random number produced by the proposed generator can meet the requirement of the high quality in cryptography. Since our proposed random number generator is based on the core operations of isogeny cryptosystem, it can be designed and implemented efficiently using the existing components.

Reference

[1]. Boneh D, Lipton R. Quantum cryptanalysis of hidden linear functions[C]. Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology (LNCS 963), 1995:424-437.

[2]. Rostovtsev A and Stolbunov A, Public-key cryptosystem based on isogenies. Cryptology ePrint Archive,http://eprint.iacr.org/.

[3]. A. Menezes, Elliptic Curve P&ic Key Cryptosystems, Kluwer Academic, 1993.

[4]. D. Knuth, The Art of Computer Programming, Volume II: Seminumerical Algorithms, Third Edition, Addison-Wesley, 1998.

[5]. M. Blum and S. Micali, How to generate cryptographically strong sequences of pseudo-random bits, SIAM J. Comput. 1984, 13 (4), 850-863.

[6]. Goldreich, S. Goldwasser and S. Micali, How to construct random functions, J. ACM 1986, 33 (4), 792-807.

[7]. J.A. Gonzalez and R. Pino, A random number generator based on unpredictable chaotic functions, Comput. Phys. Commun. 1999, 120, 109-144,.

[8]. Elkies N. Elliptic and modular curves over _nite _elds and related computational issues. Pages 21-76 in Computational Perspectives on Number Theory: Proceedings of a Conference

in Honor of A.O.L. Atkin (D.A. Buell and J.T. Teitelbaum, eds.;AMS/International Press, 1998.

[9]. Muller V. Ein Algorithmus zur Bestimmung der Punktanzahl elliptisher Kurven uber endlichen Korpern der Charakteristik groser drei. Saarbrucken, 1995.

[10]. http://www.informatik.tu-darmstadt.de/TI/Forschung/ECC.

[11]. F.Morain, E.Schost, Fast Algorithms for Computing Isogenies between Elliptic Curves. http://www.lix.polytechnique.fr/~morain/jcomp.pdf, 2006.

[12]. M. Blum and S. Micali, How to generate cryptographically strong sequences of pseudo-random bits, SIAM J. Comput. 1984, 13 (4), 850-863.

[13]. Goldreich, S. Goldwasser and S. Micali, How to construct random functions, J. ACM, 1986, 33 (4), 792-807.

[14]. Cryptographic Module Validation (CMV) Programs at NIST, http: //csrc .nist . gov/cryptval/.

[15]. M. Brown, D. Hankerson, J. López, A. Menezes. Software implementation of the NIST elliptic curves over prime fields,

[16]. http://www.dms.auburn.edu/faculty/hankerson/full-paper.pdf.

[17]. National Institute of Standard and Technology, “security requirements for cryptographic modules”, FIPS 140-1，Jan,1994.

[18]. NIST Special Publication 800-22, “A Statistical Test for Random and Pseudorandom Number Generators for Cryptographic Application”, May15, 2001.

[19]. Cohen H. A course in computational number theory. 3-rd edition, Springer-Verlag, 1996.

[20]. Shanks D., Class number, a theory of factorisation, and genera, Proceedings of the symposium on pure mathematics, vol. 20, AMS, 1971, pp. 415-440.

[21]. NIST, A Statistical Test Suite for Random and Pseudo-random Number Generators for Cryptographic Applications, http://csrc.nist.gov/rng/rng2.html,2001.