Cryptography and Communications (2022) 14:505–525
https://doi.org/10.1007/s12095-021-00535-6

Attacking the linear congruential generator on elliptic curves via lattice techniques

Jaime Gutierrez[1]

Received: 19 June 2021 / Accepted: 26 August 2021 / Published online: 12 September 2021

Abstract
In this paper we study the linear congruential generator on elliptic curves from the cryptographic point of view. We show that if sufficiently many of the most significant bits of the *composer* and of three consecutive values of the sequence are given, then one can recover the *seed* and the *composer* (even in the case where the elliptic curve is private). The results are based on lattice reduction techniques and improve some recent approaches of the same security problem. We also estimate limits of some heuristic approaches, which still remain much weaker than those known for nonlinear congruential generators. Several examples are tested using implementations of ours algorithms.

Keywords PseudoRandom Bit Generator - Elliptic Curves- Lattice based attack

Mathematics Subject Classification (2010) 94A60 - 11T71 - 94A55

1 Introduction

A PseudoRandom Bit Generator(PRBG) is a deterministic algorithm that, once initialized with some random value (called the seed), outputs a sequence that appears random, in the sense that an observer who does not know the value of the seed cannot distinguish the output from that of a (true) random bit generator. PRBG's have important applications on simulations (e.g. for the Monte Carlo method), electronic games (e.g. for procedural generation), and cryptography. Good statistical properties are a vital requirement for the output of a PRBG. Cryptographic applications require the output not to be predictable from earlier outputs, and more elaborate algorithms, which do not inherit the linearity of simpler PRBGs, are needed.

There is a vast literature devoted to generating pseudorandom numbers using arithmetic of finite field and residue rings, see [33, 37, 38, 45]. In 1994, Hallgreen [21] proposed a pseudorandom number generator based on the group of points of an elliptic curve defined over a prime finite field.

✉ Jaime Gutierrez
jaime.gutierrez@unican.es

[1] University of Cantabria, E-39005 Santander, Spain

For a prime p, denote by $\mathbb{F}_p$ the field of p elements and always assume that it is represented by the set $\{0, 1, \ldots, p-1\}$. Accordingly, sometimes, where obvious, we treat elements of $\mathbb{F}_p$ as integer numbers in the above range.

Let E be an elliptic curve defined over $\mathbb{F}_p$ given by an *affine Weierstrass equation*, which for $\gcd(p, 6) = 1$ takes form

$$Y^2 = X^3 + aX + b, \tag{1}$$

for some $a, b \in \mathbb{F}_p$ with $4a^3 + 27b^2 \neq 0$.

We recall that the set $E(\mathbb{F}_p)$ of $\mathbb{F}_p$-rational points forms an abelian group, with the *point at infinity* $\mathcal{O}$ as the neutral element of this group (which does not have affine coordinates).

For a given point $G \in E(\mathbb{F}_p)$ the **Linear Congruential Generator on Elliptic Curves, EC-LCG** is a sequence U_n of pseudorandom numbers defined by the relation

$$U_n = U_{n-1} \oplus G = nG \oplus U_0, \quad n = 1, 2, \ldots, \tag{2}$$

where $\oplus$ denotes the group operation in $E(\mathbb{F}_p)$ and $U_0 \in E(\mathbb{F}_p)$ is the *initial value* or *seed*. We refer to G as the *composer* of the EC-LCG.

It is clear that the period of the sequence (2) is equal to the order of G. The EC-LCG provides a very attractive alternative to linear and non-linear congruential generators with many applications to cryptography and it has been extensively studied in the literature, see [3, 12, 17, 18, 21, 22, 34, 35, 39, 40].

In the cryptographic setting, the initial value $U_0 = (x_0, y_0)$ and the constants G, a, and b are assumed to be the secret key, and we want to use the output of the generator as ephemeral key of a stream cipher. Of course, if two consecutive values U_n are revealed, it is almost always easy to find U_0 and G. So, we output only the most significant bits of each U_n in the hope that this makes the resulting output sequence difficult to predict.

It is known that not too many bits can be output at each stage: the EC-LCG is unfortunately predictable if sufficiently many bits of its consecutive elements are revealed, see [20, 31, 32].

Now, we are formalising the results. Assume that the sequence (U_n) is not known, but for some n, approximations W_j of two consecutive values U_{n+j}, $j = 0, 1$ are given. The results involve another parameter Δ which measures how well the values W_j approximate the terms U_{n+j}. This parameter is assumed to vary independently of p subject to satisfying the inequality $\Delta < p$ (and is not involved in the complexity estimates of our algorithms). More precisely, we say that $W = (x_W, y_W) \in \mathbb{F}_p^2$ is a Δ-*approximation* to $U = (x_U, y_U) \in \mathbb{F}_p^2$ if there exist integers e, f satisfying:

$$|e|, |f| \leq \Delta, \ x_W + e = x_U, \ y_W + f = y_U.$$

In general, we say that $W = (\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathbb{F}_p^n$ is a Δ-*approximation* to $U = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_p^n$ if there exist integers ϵ_i, $(i = 1, \ldots, n)$ satisfying:

$$|\epsilon_i| \leq \Delta, \ \alpha_i + \epsilon_i = x_i, \quad i = 1, \ldots, n.$$

The case where Δ grows like a fixed power p^δ where $0 < \delta < 1$ corresponds to the situation where a positive proportion δ of the least significant bits of terms of the output sequence remain hidden. The goal is to get δ as much larger as possible, recovering the rest of the bits in polynomial time.

The problem is a particular case of the following computational problem: given $f_1(X_1, \ldots, X_n), \ldots, f_s(X_1, \ldots, X_n)$ irreducible multivariate polynomials defined over the integer ring $\mathbb{Z}$, having a common root $(x_1, \ldots, x_n)$ modulo a known integer N, namely,

$$f_i(x_1, \ldots, x_n) \equiv 0 \bmod N, \quad i = 1, \ldots, s.$$

The root should be *small* root, in the sense that each x_i is bounded by a known value Δ. We require to bound the sizes of Δ allowing to recover the desired root in polynomial time. For polynomial in one variable an algorithm has been given by Coppersmith in [9]. For bivariate polynomials does exist different methods in [6, 10, 11, 16] and, for general multivariate polynomials in [16, 24]. All of them are based on the so called lattice reduction techniques, also called the LLL techniques, because the celebrated LLL algorithm of Lenstra, Lenstra and Lovász [30]. However in the general case only heuristic results are known, which are just generalization of the original result by Coppersmith.

An algorithm to recover the seed U_0 in deterministic polynomial time if $\Delta < p^{1/6}$, requiring compute a closest vector of a lattice of dimension 8 and coefficients size $\log p$ is presented in [20]. The recent results [31, 32] recover 'heuristically' the seed U_0 if $\Delta < p^{1/5}$. The heuristic method, since there is no guarantee of success, may fail by several reason, among them the difficulty of finding a short vector in a high dimensional lattice. Since the number of the monomials is quite large; those results does not imply practical attacks, since the naive application of Coppersmith method is impractical for high dimensional lattice.

The computation which is theoretically polynomial-time becomes in practice prohibitive, for instance and according to [32], if the quality is $0.187 < 0.2 = 1/5$ requires 188 polynomials and 314 monomials, so the lattice dimension is 502.

In this paper, we prove a deterministic algorithm to recover the seed U_0 in polynomial time if $\Delta < p^{1/6}$, requiring compute a closest vector of a lattice of dimension 5. Previous result in [20] required a lattice of dimension 8.

We also provide an heuristic method to recovering U_0 if $\Delta < p^{\frac{k-1}{4k-2}}$, requiring compute a closest vector for a lattice of dimension $3k-1$, when $k > 2$ consecutive $\Delta-$approximations to points $U_i, (0, \ldots, k-1)$ of the curve E are given. A similar result is also presented in [31, 32] with a theoretical better bound $\Delta < p^{\frac{3k}{11k+4}}$, but again requiring computing LLL's algorithm for a lattice of huge dimension. In fact, there is no practical way of testing their methods, not only because the lattice large dimension used, but the size of the prime p should be several hundreds of bits. On the other hand, for instance, we can recover the sequence produced by EC-LCG if only three consecutive $\Delta-$approximations are given as soon as $\Delta < p^{1/5}$ requiring, the most time consuming, to find a closest vector for a lattice of dimension 7, and it matched by primes p of only 1000 bits.

In principle, we cannot obtain any approximation to composer G from any approximations to two consecutive values U_n, U_{n+1} of the EC-LCG, because the elliptic curve group operation. We also rigorously demonstrate our approach in the special case when we have an approximation to composer G; we show that given Δ if sufficiently many of the most significant bits of G and of three consecutive values U_n, U_{n+1}, U_{n+2} of the EC-LCG are given, one can recover the seed U_0 and the composer G as soon as $O(\Delta) < p^{1/6}$ requiring compute two closest vector for two lattices of dimension 7. And heuristic algorithm if $O(\Delta) < p^{5/12}$ by computing a short vector for a lattice of dimension 9. Finally, we obtain an heuristic method to recovering the whole sequence if $\Delta < p^{\frac{k-1}{5k-4}}$, by computing a closest vector of a certain lattice of dimension $4k-3$ when $k > 2$ consecutive $\Delta-$approximations to points $U_i, (0, \ldots, k-1)$ of the curve E and an $\Delta-$approximation to composer G are given.

This suggests that for cryptographic applications EC-LCG should be used with great care. For the *linear congruential generator* similar problems have been introduced by Knuth [26] and then considered in [7, 13, 23, 27]; see also the surveys [8, 28]. The *quadratic congruential generator* and the *inverse congruential generator* have been studied in [4] and [15], see also the recent paper [44] for a more general problem

On the other hand, our results are substantially weaker than those known for the linear and nonlinear congruential generators.

The remainder of the paper is structured as follows. We start with a very short outline of some basic facts about the Closest Vector Problem (CVP), and the polynomial equations associated to the elliptic curve abelian group in Section 2. In Section 3 we attack the EC-LCG when the composer G is known. Section 4 is dedicated to study the case when the composer G is private. Then in Section 5 we discuss the results of numerical tests of pour approaches. We conclude with Section 6, which makes some final comments and poses open questions.

Throughout the paper, we use the convention that the parameters on which the implied constant in a Landau symbol O are written in the subscript of O. A symbol O without a subscript indicates and absolute implied constant.

2 Preliminaries

2.1 Closest vector problem in lattices

Here we review some results and definitions concerning the Closest Vector Problem, all of which can be found in [19]. For more details and more recent references, we recommend consulting [23].

Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_s\}$ be a set of linearly independent vectors in $\mathbb{R}^r$. The set

$$\mathcal{L} = \{c_1\mathbf{b}_1 + \cdots + c_s\mathbf{b}_s \mid c_1, \ldots, c_s \in \mathbb{Z}\}$$

is an *s-dimensional lattice* with *basis* $\{\mathbf{b}_1, \ldots, \mathbf{b}_s\}$. If $s = r$, the lattice $\mathcal{L}$ is of *full rank*.

One basic lattice problem is the *Closest Vector Problem (CVP)*: given a basis of a lattice $\mathcal{L}$ in $\mathbb{R}^s$ and a shift vector $\mathbf{t}$ in $\mathbb{R}^s$, the goal is finding a vector in the lattice $\mathcal{L}$ closest to the target vector $\mathbf{t}$. It is well known that this problem is **NP**-hard when the dimension grows. However, it is solvable in deterministic polynomial time provided that the dimension of $\mathcal{L}$ is fixed (see [25], for example).

In fact, lattices in this paper consist of integer solutions:
$\mathbf{x} = (x_0, \ldots, x_{s-1}) \in \mathbb{Z}^s$ of a system of congruences

$$\sum_{i=0}^{s-1} a_{ij}x_i \equiv 0 \bmod q_j, \qquad j = 1, \ldots, m,$$

modulo some positive integers $q_1, \ldots, q_m$. Typically (although not always) the volume of such a lattice is the product $Q = q_1 \cdots q_m$. Moreover, all the aforementioned algorithms, when applied to such a lattice, become polynomial in $\log Q$. If
$textbfb_1, \ldots, \mathbf{b}_s\}$ is a basis of the above lattice, by the Hadamard inequality we have:

$$\prod_{i=1}^{s} \|\mathbf{b}_i\| \geq \mathrm{vol}(\mathcal{L}). \tag{3}$$

For a slightly weaker task of finding a sufficiently close vector, the celebrated *LLL algorithm* of Lenstra, Lenstra and Lovász [30] provides a desirable solution, as noticed by [2], that is, a polynomial time algorithm in the bit size coefficients of the lattice basis and also of the lattice dimension. Here, we state this result as Lemma 1.

Lemma 1 *There exists a deterministic polynomial time algorithm which, when given an s-dimensional full rank lattice $\mathcal{L}$ and a shift vector $\boldsymbol{t}$, finds a lattice vector $\boldsymbol{u} \in \mathcal{L}$ satisfying the inequality*

$$\|\boldsymbol{t} - \boldsymbol{u}\| \leq 2^{s/2} \min\{\|\boldsymbol{t} - \boldsymbol{v}\| : \boldsymbol{v} \in \mathcal{L}\}.$$

An outline of the algorithms presented in this paper goes as follows. They are divided into two stages.

- **Stage 1**: We construct a certain lattice $\mathcal{L}$ of dimension s; this lattice depends on the given approximations. We also show that a certain vector $\mathbf{E}$ directly related to missing information is a very short vector. A closest vector $\mathbf{F}$ is found; see [25] for fixed dimension and Lemma 1 the approximation solution for arbitrary dimension.
- **Stage 2**: We show that $\mathbf{F}$ provides the required information about $\mathbf{E}$ if the approximation is good enough.

Many other results on both exact and approximate finding of a closest vector in a lattice are discussed in [19, 23].

2.2 The polynomial equation of the group associated to an elliptic curve

The operation $\oplus$ acts over the points $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ of $E(\mathbb{F}_p)$ with $P, Q \neq \mathcal{O}$ as follows:

$$P \oplus Q = R = (x_R, y_R)$$

- If $x_P \neq x_Q$, then

$$x_R = m^2 - x_P - x_Q, \;\; y_R = m(x_P - x_R) - y_P, \quad \text{where} \quad m = \frac{y_Q - y_P}{x_Q - x_P}. \tag{4}$$

- If $x_P = x_Q$ but $y_P \neq y_Q$, then $P \oplus Q = \mathcal{O}$.
- If $P = Q$ and $y_P \neq 0$, then

$$x_R = m^2 - 2x_P, \;\; y_R = m(x_P - x_R) - y_P, \quad \text{where} \quad m = \frac{3x_P^2 + a}{2y_P}. \tag{5}$$

- If $P = Q$ and $y_P = 0$, then $P \oplus Q = \mathcal{O}$.

See [1, 5, 43] for these and other general properties of elliptic curves.

Our context is a pseudorandom bit generator which outputs affine points in an elliptic curve. One obtains recursively them by operating a fixed composer G to the previous value. So, almost always, the above equations in the first case (4) will determine the process.

If P is not Q or $-Q$, then, clearing denominators in (4), we can translate $P \oplus Q = R$ into the following identities in the field $\mathbb{F}_p$:

$$L_1 = L_1(x_Q, y_Q, x_P, y_P, x_R) \equiv 0 \bmod p$$

and

$$L_2 = L_2(x_Q, y_Q, x_P, y_P, x_R, y_R) \equiv 0 \bmod p,$$

where

$$\begin{aligned} L_1 &= {x_Q}^3 + x_R {x_Q}^2 - x_P {x_Q}^2 - 2\, x_R x_Q x_P - x_Q {x_P}^2 \\ &\quad + {x_P}^3 + 2 y_Q y_P + x_R {x_P}^2 - {y_Q}^2 - {y_P}^2, \\ L_2 &= y_R x_Q - y_R x_P - y_Q x_P + y_Q x_R - y_P x_R + y_P x_Q. \end{aligned} \tag{6}$$

Lemma 2 *Let $L_1(X_Q, Y_Q, X_P, Y_P, X_R), L_2(X_Q, Y_Q, X_P, Y_P, X_R, Y_R)$ be elements of the polynomial ring $\mathbb{F}_p[X_Q, Y_Q, X_P, Y_P, X_R, Y_R]$ and let U, V, W be algebraically independent variables.*

1. *Let* Φ *be the linear transformation:*

$$(X_P \to X_P, Y_P \to Y_P, X_Q \to X_P + U, Y_Q \to Y_P + V, X_R \to X_P + W)$$

we have $\Phi(L_1) = 3X_P U^2 + U^3 + U^2 W - V^2$.

2.

$$L_1(X_Q, Y_Q + U, X_P, Y_P + U, X_R) = L_1(X_Q, Y_Q, X_P, Y_P, X_R)$$
$$L_2(X_Q + U, Y_Q, X_P + U, Y_P, X_R + U, Y_R) = L_2(X_Q, Y_Q, X_P, Y_P, X_R, Y_R)$$

Proof It is straightforward. □

On the other hand, since $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$ and $R = (x_R, y_R)$ are points of the elliptic curve E, we have:

$$\begin{aligned} y_P^2 &= x_P^3 + ax_P + b, \\ y_Q^2 &= x_Q^3 + ax_Q + b, \\ y_R^2 &= x_R^3 + ax_R + b. \end{aligned}$$

Eliminating the curve parameters a, b and assuming that $x_P \neq x_R$, we obtain the following polynomial $L_3 \in \mathbb{F}_p[X_Q, Y_Q, X_P, Y_P, X_R, Y_R]$

$$\begin{aligned} L_3 = &-X_R^3 X_P + X_R^3 X_Q + X_R X_P^3 - X_R Y_P^2 - X_R X_Q^3 + X_R Y_Q^2 \\ &+ Y_R^2 X_P - Y_R^2 X_Q - X_P^3 X_Q + X_P X_Q^3 - X_P Y_Q^2 + Y_P^2 X_Q \end{aligned} \tag{7}$$

verifying $L_3(x_Q, y_Q, x_P, y_Q, x_R, y_R) \equiv 0 \bmod p$. Now, we consider the linear map Θ :

$$(X_Q \to X_Q, Y_Q \to Y_P + U, X_P \to X_P, Y_P \to Y_P, X_R \to X_R, Y_R \to -Y_P + V)$$
$$\Theta(L_3) = [2U(X_Q - X_P) + 2V(X_R - X_P)]Y_P + A \tag{8}$$

where degree of $A \in \mathbb{F}_p[X_Q, Y_Q, X_P, Y_P, X_R, Y_R, U, V]$ with respect the variable Y_P is zero.

3 Predicting EC-LCG for Known composer

Assume that a, b are unknown, but the prime p is given to us. In [20] shows that when we are given Δ-approximations W_n, W_{n+1} to (respectively) two consecutive affine values U_n, U_{n+1} produced by the EC-LCG; we can recover the exact values, provided that x_0 does not lie in a certain set, whose size is bounded by $O(\Delta^6)$. Note that once two affine points in a curve as (1) are given, such that their first component is different, the curve (the parameters a and b) are determined. Then, after discovering the values U_n and U_{n+1}, we can reproduce (backwards and forwards) the whole sequence. To simplify the notation, we assume that $n = 0$ from now on.

We write $W_j = (\alpha_j, \beta_j)$, $U_j = (x_j, y_j)$, for $j = 0, 1$; so there exist integers e_j, f_j for j=0, 1 with:

$$\begin{aligned} x_j &= \alpha_j + e_j, \quad y_j = \beta_j + f_j \\ |e_j|, |f_j| &\leq \Delta, \quad j = 0, 1. \end{aligned} \tag{9}$$

Theorem 1 [20] *With the above notations and definitions, there exists a set* $\mathcal{U}(\Delta; a, x_G, y_G) \subseteq \mathbb{F}_p$ *of cardinality* $\#\mathcal{U}(\Delta; a, x_G, y_G) = O(\Delta^6)$ *with the following property: whenever* $x_0 \notin \mathcal{U}(\Delta; a, x_G, y_G)$ *then, given* $\Delta-$*approximations* W_0, W_1 *to two consecutive affine values* U_0, U_1 *produced by linear congruential generator on elliptic curves (2), and given the prime* p *one can recover the seed* U_0 *in deterministic polynomial time.*

The proof of this result in [20], the 'bad' set of values $\mathcal{U}(\Delta; a, x_G, y_G)$ for the components x_0 is described, proving whenever that value lies outside the set, the algorithm works correctly. Furthermore, the size of the set is asymptotically bounded with Δ^6. This means that if $\Delta = o(p^{1/6})$ and p is large enough, assuming an uniform distribution of probabilities for $x_0 \in \mathbb{F}_p$, the method is unlikely to fail.

The proof in [20] requires the two polynomials L_1 and L_2 of (6) and 8 monomials, so the involved lattice has dimension 8. Here, we use only the polynomial L_2 of (6), then the corresponding lattice dimension is only 5. The present proof is a simple observation of the same strategy, we have included here a significant part of the details for the reader convenience.

Proof We assume that $x_0 \in \mathbb{F}_p$ is chosen so as not to lie in a certain subset $\mathcal{U}(\Delta; a, x_G, y_G)$ of $\mathbb{F}_p$ We place the value $x_G \in \mathcal{U}(\Delta; a, x_G, y_G)$, so that U_0 is not G or $-G$. Then, clearing denominators in (4),

$$U_1 = U_0 \oplus G \tag{10}$$

we obtain

$$L_2(x_G, y_G, x_0, y_0, x_1, y_1) \equiv 0 \bmod p,$$

Using the equalities $x_j = \alpha_j + e_j$ and $y_j = \beta_j + f_j$ for $j = 0, 1$, the polynomial L_2 of (6) become:

$$(-\beta_1 - y_G)\, e_0 + (x_G - \alpha_1)\, f_0 + (y_G - \beta_0)\, e_1 + (x_G - \alpha_0)\, f_1 - [e_0 f_1 + e_1 f_0] =$$

$$\beta_1\alpha_0 - x_G\beta_1 + y_G\alpha_0 - y_G\alpha_1 + \beta_0\alpha_1 - x_G\beta_0.$$

Now, we linearize this polynomial system. Writing

$$\begin{array}{l} B_0 \equiv \beta_1\alpha_0 - x_G\beta_1 + y_G\alpha_0 - y_G\alpha_1 + \beta_0\alpha_1 - x_G\beta_0 \bmod p, \\ B_1 \equiv -\beta_1 - y_G \bmod p, \quad B_2 \equiv x_G - \alpha_1 \bmod p, \\ B_3 \equiv y_G - \beta_0 \bmod p, \quad B_4 \equiv x_G - \alpha_0 \bmod p, \\ B_5 \equiv -1 \bmod p. \end{array} \tag{11}$$

is it easy to check that the vector

$$E = (\Delta e_0, \Delta f_0, \Delta e_1, \Delta f_1, e_1 f_0 + e_0 f_1) =$$

$$(\Delta E_1, \Delta E_2, \Delta E_3, \Delta E_4, E_5)$$

is a solution to the following linear system of congruences:

$$\begin{array}{ll} \sum_{i=1}^{4} B_i X_i + \Delta B_5 \equiv \Delta B_0 \bmod p, & \\ X_1 \equiv X_2 \equiv X_3 \equiv X_4 & \equiv 0 \bmod \Delta \end{array} \tag{12}$$

Moreover, bounds (9) implies $\mathbf{E}$ is a relatively short vector. We have:

$$|E_i| \leq \Delta, i = 1, 2, 3, 4, \ |E_5| \leq 2\Delta^2; \ \|\mathbf{E}\| \leq \sqrt{8}\Delta^2. \tag{13}$$

Let $\mathcal{L}$ be the lattice consisting of integer solutions $\mathbf{X} = (X_1, X_2, \ldots, X_5) \in \mathbb{Z}^5$ of the system of congruences:

$$\begin{array}{l} \sum_{i=1}^{4} B_i X_i + B_5 X_5 \equiv 0 \bmod p, \\ X_1 \equiv X_2 \equiv X_3 \equiv X_4 \equiv 0 \bmod \Delta. \end{array} \tag{14}$$

We compute a solution $\mathbf{T}$ of the system of congruences (12), using linear diophantine equations methods. Applying an algorithm solving the CVP for the shift vector $\mathbf{T}$ and the lattice (14), we obtain a vector

$$\mathbf{F} = (\Delta F_1, \Delta F_2, \Delta F_3, \Delta F_4, F_5).$$

We have $\mathbf{F} = \mathbf{v} + \mathbf{T}$ (where $\mathbf{v}$ is the lattice vector returned by the CVP algorithm) is the vector of minimal norm satisfying (12), hence $\mathbf{F}$ must have norm at most equal to the norm of the solution $\mathbf{E}$. Using the bounds (13), we get:

$$\|\mathbf{F}\| \leq \sqrt{8}\Delta^2. \tag{15}$$

Note that we can compute $\mathbf{F}$ in polynomial time from the information we are given. We might hope that $\mathbf{E}$ and $\mathbf{F}$ are the same, or at least, that we can recover the approximations errors from $\mathbf{F}$. If not, we will show that x_0 belongs to a subset $\mathcal{U}(\Delta; a, x_G, y_G) \subseteq \mathbb{F}_p$ of cardinality $\#\mathcal{U}(\Delta; a, x_G, y_G) = O(\Delta^6)$. Vector $\mathbf{D} = \mathbf{E} - \mathbf{F}$ lies in $\mathcal{L}$:

$$\mathbf{D} = (\Delta D_1, \Delta D_2, \Delta D_3, \Delta D_4, D_5), \quad D_i = E_i - F_i, i = 1, \ldots, 5.$$

Bounds (13) and (15) imply $\|\mathbf{D}\| \leq 4\Delta^2$ and

$$|D_i| \leq 4\Delta, i = 1, 2, 3, 4, \quad |D_5| \leq 4\Delta^2.$$

From here, by closely following the proof in [20], since only depends on L_2 and D_i, we bound the "bad" possibilities for which this process does not succeed. □

We now present some heuristic arguments showing that Theorem 1 could possibly be strengthened so that it becomes nontrivial when the precision Δ is of the order of $p^{1/4}$ rather than of order $p^{1/6}$ as currently.

The heuristic that we use is of a totally different nature than that used in the so called Coppersmith's method, where the heuristic assumption is that all created polynomials define a zero dimension algebraic variety. Here, we use the so-called "Gaussian heuristic" that suggests that and s-dimensional lattice $\mathcal{L}$ with volume $vol(\mathcal{L})$ is unlikely to have a nonzero vector which is substantially shorter than $vol(\mathcal{L})^{1/s}$. Moreover, if it is known that such a very short vector does exist, then up to a scalar factor it is likely to be the only vector with this property.

Let us formalise the problem. Again, we assume that a, b are unknown, but the prime p is given to us. Suppose that we are given $k \geq 2$ consecutive $\Delta-$approximations $W_j = (\alpha_j, \beta_j)$ to $U_j = (x_j, y_j) \in E(\mathbb{F}_p)$ $(j = 0, \ldots, k-1)$, produced by the EC-LCG, so there exist integers e_j, f_j with:

$$\begin{array}{l} x_j = \alpha_j + e_j, \quad y_j = \beta_j + f_j \\ |e_j|, |f_j| \leq \Delta, \quad j = 0, \ldots, k-1 \end{array} \tag{16}$$

Theorem 2 *With above notation, under the 'Gaussian heuristic' we can recovering the seed U_0 in polynomial time in* $\log p$ *as soon as $\Delta < p^{\frac{k-1}{4k-2}}$ by computing a closest vector of a certain lattice of dimension* $3k - 1$.

Proof As in the previous Theorem 1 we use only the polynomial L_2 in (6) for a pair of points (U_j, U_{j+1}) for $j = 0, \ldots, k-2$. Again, we translate $U_{j+1} = U_j \oplus G$ into a polynomial system of equation:

$$\begin{array}{l} L_2^{(0)}(x_G, y_G, x_0, y_0, x_1, y_1) = L_2(x_G, y_G, x_0, y_0, x_1, y_1) \equiv 0 \bmod p, \\ L_2^{(1)}(x_G, y_G, x_1, y_1, x_2, y_2) = L_2(x_G, y_G, x_1, y_1, x_2, y_2) \equiv 0 \bmod p, \\ \ldots \\ \ldots \\ L_2^{(k-2)}(x_G, y_G, x_{k-2}, y_{k-2}, x_{k-1}, y_{k-1}) = L_2(x_G, y_G, x_{k-2}, y_{k-2}, x_{k-1}, y_{k-1}) \equiv 0 \bmod p. \end{array}$$

Using the equalities (16) the above polynomial $L_2^{(j)}$, for $j = 0, \ldots, k-2$, becomes:

$$\left(-\beta_{j+1}-y_G\right)e_j+\left(x_G-\alpha_{j+1}\right)f_j+\left(y_G-\beta_j\right)e_{j+1}+\left(x_G-\alpha_j\right)f_{j+1}-[e_jf_{j+1}+e_{j+1}f_j]=$$
$$\beta_{j+1}\alpha_j-x_G\beta_{j+1}+y_G\alpha_j-y_G\alpha_{j+1}+\beta_j\alpha_{j+1}-x_G\beta_j.$$

Now, we linearize this polynomial system. Writing, for $j=0,\ldots,k-2$ and for $i=0,\ldots,k-1$:

$$\begin{aligned}
&\Sigma^{(j)}\equiv\beta_{j+1}\alpha_j-x_G\beta_{j+1}+y_G\alpha_j-y_G\alpha_{j+1}+\beta_j\alpha_{j+1}-x_G\beta_j \bmod p,\\
&A_j^{(j)}\equiv-\beta_{j+1}-y_G \bmod p,\ A_{j+1}^{(j)}\equiv y_G-\beta_j \bmod p,\\
&B_{j+1}^{(j)}\equiv x_G-\alpha_{j+1} \bmod p,\ B_j^{(j)}\equiv x_G-\alpha_j \bmod p,\\
&A_i^{(j)}\equiv 0 \bmod p,\ B_i^{(j)}\equiv 0 \bmod p,\quad i\neq j\vee i\neq j+1,\\
&C_j^{(j)}\equiv -1 \bmod p,\ C_i^{(j)}\equiv 0 \bmod p,\ j\neq i.
\end{aligned}$$

we obtain that vector $\mathbf{E}=$

$$\begin{aligned}
&(\Delta e_0,\Delta e_1,\ldots,\Delta e_{k-1},\Delta f_0,\Delta f_1,\ldots,\Delta f_{k-1},\\
&e_1f_0+e_0f_1,e_2f_1+e_1f_2,\ldots,e_{k-1}f_{k-2}+e_{k-2}f_{k-1})\\
&=(\Delta E_1,\ldots,\Delta E_k,\Delta E_{k+1},\ldots,\Delta E_{2k},E_{2k+1},\ldots,E_{3k-1})
\end{aligned}$$

is a solution to the following linear system of congruences ($j=0,\ldots,k-2$):

$$\begin{aligned}
&\textstyle\sum_{i=0}^{k-1}A_i^{(j)}X_i+\sum_{i=0}^{k-1}B_i^{(j)}Y_i+\sum_{i=0}^{k-1}\Delta C_i^{(j)}Z_i\equiv\Delta\Sigma^j \bmod p,\\
&X_0\equiv X_1\equiv\ldots\equiv X_{k-1} &&\equiv 0 \bmod \Delta\\
&Y_0\equiv Y_1\equiv\ldots\equiv Y_{k-1} &&\equiv 0 \bmod \Delta.
\end{aligned}\tag{17}$$

Moreover, bounds (16) imply $\mathbf{E}$ is a relatively short vector. We have:

$$\|\mathbf{E}\|\leq\sqrt{6k-6}\Delta^2.\tag{18}$$

Let $\mathcal{L}_k$ be the lattice consisting of integer solutions:

$$(X_0,\ldots,X_{k-1},Y_0,\ldots,Y_{k-1},Z_0,\ldots,Z_{k-1})\in\mathbb{Z}^{3k-1}$$

of the system of congruences, ($j=0,\ldots,k-2$):

$$\begin{aligned}
&\sum_{i=0}^{k-1}A_i^{(j)}X_i+\sum_{i=0}^{k-1}B_i^{(j)}Y_i+\sum_{i=0}^{k-1}\Delta C_i^{(j)}Z_i\equiv 0 \bmod p,\\
&X_0\equiv X_1\equiv\ldots\equiv X_{k-1}\equiv 0 \bmod\Delta\\
&Y_0\equiv Y_1\equiv\ldots\equiv Y_{k-1}\equiv 0 \bmod\Delta.
\end{aligned}\tag{19}$$

We compute a solution $\mathbf{T}$ of the system of congruences (17), using linear diophantine equations methods. Applying an algorithm solving the CVP for the shift vector $\mathbf{T}$ and the lattice (19) we obtain a vector

$$\mathbf{F}=(\Delta F_1,\ldots,\Delta F_{2k},F_{2k+1},\ldots,F_{3k-1}).$$

We have $\mathbf{F}=\mathbf{v}+\mathbf{T}$ (where $\mathbf{v}$ is the lattice vector returned by the CVP algorithm) is the vector of minimal norm satisfying (17), hence $\mathbf{F}$ must have norm at most equal to the norm of the solution $\mathbf{E}$. Using the bounds (18), we get:

$$\|\mathbf{F}\|\leq\sqrt{6k-6}\Delta^2.\tag{20}$$

Note that we can compute $\mathbf{F}$ in polynomial time from the information we are given, see Lemma 1. We might hope that $\mathbf{E}$ and $\mathbf{F}$ are the same, or at least, that we can recover the approximations errors from $\mathbf{F}$. The volume of the lattice (19) is $p^{k-1}\Delta^{2k}$ (see Section 2.1)

Then, using (20) and *Gaussian heuristic* vector $\mathbf{E}$ is likely to be the one founded whenever $\Delta^2 < p^{\frac{k-1}{3k-1}} \Delta^{\frac{2k}{3k-1}}$, this is:

$$\Delta < p^{\frac{k-1}{4k-2}}.$$

This finishes the proof. □

This time, we did not provide a rigorous proof to bound the number of possibilities for which this method could fail. We will see in Section 5 that our SAGEMATH implementation certifies the above bound.

The following illustrates the importance of knowledge parameter a of the elliptic curve (1).

Remark 1 If three consecutive values of the EC-LCG are given, then can eliminated G from $U_0 \oplus G = U_1$ and $U_1 \oplus G = U_2$:

$$U_2 \oplus U_0 = 2U_1$$

So, given three $\Delta-$approximations to U_0, U_1, U_2 and, assuming that U_0 and U_1 are not G or $-G$ and $y_0 y_1 \neq 0$, clearing denominators in (4) and (5) we can translate equation $U_2 \oplus U_0 = 2U_1$ into two polynomial identities in the field $\mathbb{F}_p$, but involving the unknown parameter a.

On the other hand, given the parameters a and b of the elliptic curve (1) and $\Delta-$approximation (α_0, β_0) to point $P = (x_0, y_0)$ of the curve we can recover P as soon as $\Delta < p^{1/7}$, see [14, 16].

4 Predicting EC-LCG for unknown composer

In previous section, it has been assumed that the cryptanalyst has access to the composer G, which places his task in a quite optimistic frame. This section is devoted to the case that the parameter G is also private. This case is studied in [20] and also [32] requiring three approximations, no necessarily consecutive, instead of two. They consider the information given as approximations to arbitrary points in the same elliptic curve, in such a way that they are not taking advantage from the knowledge of the procedure which has generated them. In other words, they provide a method to recover three points lying in an elliptic curve in the form (1), given corresponding approximations. And then use that method in the frame of an EC-LCG and three values partially revealed. Both methods are heuristics. In [20] requieres a $\Delta-$approximations such that $\Delta < p^{1/46}$, a better result is presented in [31, 32], which requieres $\Delta < p^{1/24}$. Both methods are looking for small roots of polynomial (7) $L_3 \in \mathbb{F}_p[X_Q, Y_Q, X_P, Y_P, X_R, Y_R]$.

We assume that approximations to the coordinates of $G = (x_G, y_G) \in E(\mathbb{F}_p)$ and also W_n, W_{n+1} to (respectively) two consecutive affine values U_n, U_{n+1} produced by the EC-LCG; we are trying to recover the exact values, provided that the approximations are good enough.

We write $\bar{G} = (\gamma_x, \gamma_y)$ and $W_j = (\alpha_j, \beta_j)$, $U_j = (x_j, y_j)$, for $j = 0, 1$; so there exist integers h_x, h_y and e_j, f_j for j=0, 1 with:

$$\begin{aligned} x_G = \gamma_x + h_x, \quad y_G = \gamma_y + h_y, \quad \& \quad |h_x|, |h_y| \leq \Delta \\ x_j = \alpha_j + e_j, \quad y_j = \beta_j + f_j \\ |e_j|, |f_j| \leq \Delta, \quad j = 0, 1. \end{aligned} \tag{21}$$

The first attempt to design a such procedure would be, as in the previous Theorem 1, translate (10) into the identity in the fielp $\mathbb{F}_p$ getting

$$L_2(x_G, y_G, x_0, y_0, x_1, y_1) \equiv 0 \bmod p,$$

and looking for small roots of L_2. However, Lemma 2-(2) shows it is no possible recovering the seed. Neither from $L_1 \equiv 0 \bmod p$, again by Lemma 2-(2). So, we have to involve both polynomials:

$$L_1 \equiv 0 \bmod p, \quad L_2 \equiv 0 \bmod p$$

First, we rigorously demonstrate how recovering only abscissa coordinates if $\Delta < p^{1/6}$.

Lemma 3 *With the above notations and definitions, there exists a set $\mathcal{U}(\Delta) \subseteq \mathbb{F}_p^6$ of cardinality $\#\mathcal{U}(\Delta) = O(p^5\Delta^6)$ with the following property: whenever $P = (x_G, y_G, x_0, y_0, x_1, y_1) \notin \mathcal{U}(\Delta)$ then, given $\Delta-$approximation*

to point P and given the prime p, one can recover x_G, x_0, x_1 in deterministic polynomial time by computing a closest vector of a certain lattice of dimension 5.

Proof First, we place all points of the form $(x_0, y_G, x_0, x_1, y_0, y_1) \in \mathcal{U}(\Delta)$, so that U_0 is not G or $-G$. We write L_2 as

$$L_2 = (Y_P + Y_R)(X_Q - X_P) + (Y_P - Y_Q)(X_P - X_R).$$

Taking $W_0 = Y_P + Y_R$, $V_0 = X_Q - X_P$, $W_1 = Y_P - Y_Q$, $V_1 = X_P - X_R$, we consider polynomial $\bar{L}_2 \in \mathbb{F}_p[W_0, W_1, V_0, V_1]$:

$$\bar{L}_2 = W_0V_0 + W_1V_1$$

We write

$$w_0 = y_0 + y_1, \quad v_0 = x_G - x_0, \quad w_1 = y_0 - y_G, \quad v_1 = x_0 - x_1 \tag{22}$$

From (21), we obtain $(\beta_0 + \beta_1, \gamma_x - \alpha_1, \beta_1 - \gamma_y, \alpha_0 - \alpha_1) = (b_0, a_0, b_1, a_1)$ is 2Δ-approximation to the root $(w_0, v_0, w_1, v_1) \in \mathbb{F}_p^4$ of $\bar{L}_2$. So, rewriting the (21) and (22):

$$w_0 = b_0 + f_0,\ v_0 = a_0 + e_0,\ w_1 = b_1 + f_1,\ v_1 = a_1 + e_1, \quad |e_j|, |f_j| \le 2\Delta,\ j = 0, 1. \tag{23}$$

The polynomial $\bar{L}_2 \in \mathbb{F}_p[W_0, W_1, V_0, V_1]$ becomes:

$$b_0e_0 + a_0f_0 + b_1e_0 + a_1f_1 + [e_0f_0 + e_1f_1] = -(a_0b_0 + a_1b_1)$$

Writing:

$$C_0 \equiv -(a_0b_0 + a_1b_1) \bmod p, \quad C_1 \equiv b_0 \bmod p, \quad C_2 \equiv a_0 \bmod p,$$
$$C_3 \equiv b_1 \bmod p, \quad C_4 \equiv a_1 \bmod p, \quad C_5 \equiv 1 \bmod p.$$

we obtain that vector

$$\mathbf{E} = (\Delta e_0, \Delta f_0, \Delta e_1, \Delta f_1, e_0f_0 + e_1f_1) =$$

$$(\Delta E_1, \Delta E_2, \Delta E_3, \Delta E_4, E_5)$$

is a solution to the following linear system of congruences:

$$\sum_{i=1}^{4} C_iX_i + \Delta C_5X_5 \equiv \Delta C_0 \bmod p, \tag{24}$$

$$X_1 \equiv X_2 \equiv X_3 \equiv X_4 \equiv 0 \bmod \Delta.$$

Moreover, bounds (23) imply $\mathbf{E}$ is a relatively short vector. We have:

$$|E_i| \leq 2\Delta, i = 1, \ldots, 4, \ |E_5| \leq 4\Delta^2; \ \|\mathbf{E}\| \leq \sqrt{32}\Delta^2. \tag{25}$$

Let $\mathcal{L}$ be the lattice consisting of integer solutions $\mathbf{X} = (X_1, X_2, \ldots, X_5) \in \mathbb{Z}^5$ of the system of congruences:

$$\sum_{i=1}^{4} C_i X_i + \Delta C_5 X_5 \equiv 0 \bmod p, \tag{26}$$
$$X_1 \equiv X_2 \equiv X_3 \equiv X_4 \equiv 0 \bmod \Delta.$$

We compute a solution $\mathbf{T}$ of the system of congruences (24), using linear diophantine equations methods. Applying an algorithm solving the CVP for the shift vector $\mathbf{T}$ and the lattice (26), we obtain a vector

$\mathbf{F} = (\Delta F_1, \Delta F_2, \Delta F_3, \Delta F_4, F_5)$

We have $\mathbf{F} = \mathbf{v} + \mathbf{T}$ (where $\mathbf{v}$ is the lattice vector returned by the CVP algorithm) is the vector of minimal norm satisfying (24), hence $\mathbf{F}$ must have norm at most equal to the norm of the solution $\mathbf{E}$. Using the bounds (25), we get:

$$\|\mathbf{F}\| \leq \sqrt{32}\Delta^2. \tag{27}$$

Note that we can compute $\mathbf{F}$ in polynomial time from the information we are given. We might hope that $\mathbf{E}$ and $\mathbf{F}$ are the same, or at least, that we can recover the approximations errors from $\mathbf{F}$. If not, we will show that $(x_G, y_G, x_0, y_0, x_1, y_1)$ belongs to a subset $\mathcal{U}(\Delta) \subseteq \mathbb{F}_p^6$ of cardinality $\#\mathcal{U}(\Delta) = O(p^5\Delta^6)$.

Vector $\mathbf{D} = \mathbf{E} - \mathbf{F}$ lies in $\mathcal{L}$:

$$\mathbf{D} = (\Delta D_1, \Delta D_2, \Delta D_3, \Delta D_4, D_5), \quad D_i = E_i - F_i, i = 1, \ldots, 5.$$

Bounds (25) and (27) imply $\|\mathbf{D}\| \leq 8\sqrt{2}\Delta^2$ and

$$|D_i| \leq 8\sqrt{2}\Delta, i = 1 \ldots, 4, \quad |D_5| \leq 8\sqrt{2}\Delta^2. \tag{28}$$

Now, we distinguish two cases:

1. $D_i \equiv 0 \bmod p, \ i = 1, \ldots, 4,$
2. D_i is nonzero for some $i, i = 1, \ldots, 4$

In the first case, we can recover the root (w_0, v_0, w_1, v_1) of the polynomial $\bar{L}_2(W_0, V_0, W_1, V_1)$, then by (22), we have

$$x_G = x_0 + v_0, \quad x_1 = x_0 - v_1, \quad y_G = y_0 - w_1.$$

Substituting those equalities into (6) polynomial L_1 and since $x_G \neq x_0$, that is $v_0 \neq 0$, then by Lemma 2-(1) we can recover x_G, x_0, x_1.

Hence, we may assume that D_i is nonzero for some $i, i = 1, \ldots, 4$. Substituting $b_j = W_j - e_j, a_j = V_j - f_j, \ j = 0, 1$ in the first equation of lattice (26), we obtain a **nonzero** polynomial: $G = G(W_0, V_0, W_1, V_1) =$

$$(W_0 - e_0)D_1 + (V_0 - f_0)D_2 + (W_1 - e_1)D_3 + (V_1 - f_1)D_4 + D_5$$

whose coefficients are in $\mathbb{Z}[D_i, e_j, f_j]$, for $i = 0, \ldots, 5$ and $j = 0, 1$ and such that:

$$G(w_0, v_0, w_1, v_1) \equiv 0 \bmod p.$$

For every choice of D_i, e_j, f_j, for $i = 0, \ldots, 5$ and for $j = 0, 1$, the specialised $\bar{G}(W_0, W_0, W_1, V_1) \in \mathbb{F}_p[(W_0, W_0, W_1, V_1)]$ is a linear polynomial, so the number of solutions in $\mathbb{F}_p^4$ is exactly p^3. Now, for every zero (w_0, v_0, w_1, v_1) of $\bar{G}$ we have exactly

p^2 points of the form $(y_0 + y_1, x_G - x_0, y_0 - y_G, x_0 - x_1)$. In total we have p^5 points $(x_G, y_G, x_0, x_1, y_0, y_1) \in \mathbb{F}_p^6$ such that $L_2(x_G, y_G, x_0, x_1, y_0, y_1) \equiv 0 \bmod p$, we place all of then into the set $\mathcal{U}(\Delta) \subseteq \mathbb{F}_p^6$. We need to show that the cardinality of $\mathcal{U}(\Delta)$ is as claimed in the statement of the theorem. In other words, we need to prove that for every choice of D_i, e_j, f_j, $(i = 0, \ldots, 5$ and $j = 0, 1)$, the number of non zero specialized polynomials $\bar{G}(W_0, W_0, W_1, V_1) \in \mathbb{F}_p[(W_0, W_0, W_1, V_1)]$ are bounded by $O(\Delta^6)$.

We write

$$G = W_0 D_1 + V_0 D_2 + W_1 D_3 + V_1 D_4 + C,$$

where $C \equiv D_5 - (e_0 D_1 + f_0 D_2 + e_1 D_3 + f_1 D_4) \bmod p$. By (28) the number of possible choices for D_1, D_2, D_3, D_4 is $O(\Delta^4)$. On the other hand, C can take $O(\Delta^2)$ distinct values, because from bounds (23) and (28) we obtain that $D_5 - (e_0 D_1 + f_0 D_2 + e_1 D_3 + f_1 D_4) = O(\Delta^2)$. So, the number of nonzero polynomials G are bound by $O(\Delta^6)$, which finishes the proof. □

The above result also finds the values $U = y_0 + y_1$ and $V = y_0 - y_G$. Plugging $y_1 = -y_0 + U$ and $y_G = y_0 + V$ to polynomial L_3 defined in (7), then from (8) we have

$$L_3(x_G, y_0 + V, x_0, y_0, x_1, -y_0 + U) \equiv 0 \bmod p$$

So, we cannot recover y_G, y_0, y_1 from L_3.

On the other hand, substituting $y_1 = -y_0 + U$ and $y_G = y_0 + V$ in the elliptic curve (1):

$$\begin{aligned} y_0^2 &= x_0^3 + a x_0 + b, \\ (y_0 + V)^2 &= x_G^3 + a x_G + b, \\ (-y_0 + U)^2 &= x_1^3 + a x_1 + b. \end{aligned}$$

We can derive a linear equation in a and y_0:

$$a(x_0 - x_G) + 2V y_0 - x_G^3 + x_0^3 + V^2$$

Now, we can recover a and y_0 if a_0 is ϵ-approximation to a with $\epsilon < p^{1/2}$ using the same lattice technique's.

As previous remark, we can also obtain from (4) a linear equation in b and quadratic in y_0

$$b(x_G - x_0) + A y_0 + A y_0^2 + C.$$

where $A, B, C \in \mathbb{Z}[x_0, x_1, x_G, U, V]$ and $A, B, C \not\equiv 0 \bmod p$. Again, we can recover b and y_0 if b_0 is ϵ-approximation to b as soon as $\epsilon < p^{1/3}$.

Then, as consequence of Lemma 3 we have the following:

Corollary 1 *With the above notations and definitions, there exists a set $\mathcal{U}(\Delta) \subseteq \mathbb{F}_p^6$ of cardinality $\#\mathcal{U}(\Delta) = O(p^5 \Delta^6)$ with the following property: whenever $P = (x_G, y_G, x_0, y_0, x_1, y_1) \notin \mathcal{U}(\Delta)$ then, given $\Delta-$approximation to $U_0 = (x_0, y_0), U_1(x_1, y_1)$ and to $G = (x_G, y_G)$ and $\epsilon-$approximation to a with $\epsilon < p^{1/2}$ or an $\epsilon-$approximation to b with $\epsilon < p^{1/3}$ one can recover the sequence produced EC-LCG in deterministic polynomial time.*

Previous result it has been assumed that we have access to approximations to elliptic curve (1) parameters a or b, which again places this task in a quite optimistic frame.

Then suppose Δ-approximations to $U_0 = (x_0, y_0)$, $U_1 = (x_1, y_1)$ and $U_2 = (x_2, y_2)$ points generated by the EC-LCG and to $G = (x_G, y_G)$ then applying Lemma 3 to equation $U_0 \oplus G = U_1$ we recovering (x_G, x_0, x_1) and also

$$Z_1 = y_0 + y_1, \quad Z_2 = y_0 - y_G$$

Again applying the same Lemma 3 but in this case to equation $U_1 \oplus G = U_2$, we are able to recovering x_G, x_1, x_2 and

$$Z_3 = y_1 + y_2, \quad Z_4 = y_1 - y_G$$

as soon as $\Delta < p^{1/6}$. We obtain the following linear system of equation in $\mathbb{F}_p$:

$$\begin{pmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ -1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} y_G \\ y_0 \\ y_1 \\ y_2 \end{pmatrix}$$

Since $\det \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ -1 & 0 & 1 & 0 \end{pmatrix} = 1$ it has a unique solution and we can recovering all the missing information.

Theorem 3 *With the above notations and definitions, there exists two sets $\mathcal{U}(\Delta) \subseteq \mathbb{F}_p^6$ and $\mathcal{V}(\Delta) \subseteq \mathbb{F}_p^6$ both the cardinality $O(p^5\Delta^6)$ with the following property: whenever $P = (x_G, y_G, x_0, y_0, x_1, y_1) \notin \mathcal{U}(\Delta)$ and $Q = (x_G, y_G, x_1, y_1, x_2, y_2) \notin \mathcal{V}(\Delta)$ then, given $\Delta-$approximation to $U_0 = (x_0, y_0)$, $U_1(x_1, y_1)$, $U_2 = (x_2, y_2)$ and to $G = (x_G, y_G)$, one can recover the whole sequence produced by EC-LCG in deterministic polynomial time.*

Notice that we can not applies Remark 1 because the parameter a of the elliptic curve (1) is unknown.

On one hand the previous result requires computing a closest vector of two distinct lattices. On the other hand, it would be interesting attacking EC-LCG when we have access to several consecutive approximations and having an approximation to composer G. The following result try to answer those interesting computational problems.

Let us formalise the problem. Again, we assume that a, b and the composer G are unknown, but we have a $\Delta-$approximation $\bar{G} = (\gamma_x, \gamma_y)$ to $G = (x_G, y_G)$. Suppose that we are given $k + 1 \geq 2$ consecutive $\Delta-$approximations $W_j = (\alpha_j, \beta_j)$ to $U_j = (x_j, y_j) \in E(\mathbb{F}_p)$ $(j = 0, \ldots, k-1)$, produced by the EC-LCG, so there exist integers h_x, h_y and e_j, f_j with:

$$\begin{gathered} x_G = \gamma_x + h_x, \quad y_G = \gamma_y + h_y, \quad \text{with} \quad |h_x|, |h_y| \leq \Delta \\ x_j = \alpha_j + e_j, \quad y_j = \beta_j + f_j \\ |e_j|, |f_j| \leq \Delta, \quad j = 0, \ldots, k-1 \end{gathered} \tag{29}$$

Theorem 4 *With above notation, under the 'Gaussian heuristic' we can recovering the whole sequence (2) in polynomial time in* $\log p$ *as soon as $\Delta < p^{\frac{k-1}{5k-4}}$ by computing the closest vector of a certain lattice of dimension $4k-3$.*

Proof For every pair of points of the form $U_{i-1} = (x_{i-1}, y_{i-1}), U_i = (x_i, y_i), i = 1, \ldots, k-1$, we denote by

$$\begin{gathered} w_i = y_{i-1} + y_i,\ s_i = x_G - x_{i-1},\ v_i = x_{i-1} - x_i,\ t_i = y_{i-1} - y_G, \\ \bar{L}_2^{(i)} = W_i S_i + V_i T_i \in \mathbb{F}_p[W_i, S_i, V_i, T_i] \end{gathered}$$

By $U_{i-1} \oplus G = U_i$, we have $\bar{L}_2^{(i)}(w_i, s_i, v_i, t_i) \equiv 0 \bmod p$ and we obtain the following system of congruences:

$$\begin{aligned}
&\bar{L}_2^{(1)}(w_1, s_1, v_1, t_1) = L_2(x_G, y_G, x_0, y_0, x_1, y_1) \equiv 0 \bmod p,\\
&\bar{L}_2^{(2)}(w_2, s_2, v_2, t_2) = L_2(x_G, y_G, x_1, y_1, x_2, y_2) \equiv 0 \bmod p,\\
&\cdots\\
&\cdots\\
&\bar{L}_2^{(k-1)}(w_{k-1}, s_{k-1}, v_{k-1}, t_{k-1}) = L_2(x_G, y_G, x_{k-2}, y_{k-2}, x_{k-1}, y_{k-1}) \equiv 0 \bmod p.
\end{aligned}$$

From (29), we have

$$(\beta_{i-1} + \beta_i,\ \gamma_x - \alpha_{i-1},\ \beta_i - \gamma_y,\ \alpha_{i-1} - \alpha_i) = (b_i,\ p_i,\ a_i,\ q_i)$$

is 2Δ-approximation to the root (w_i, s_i, v_i, t_i) of $\bar{L}_2^i$. So, rewriting the (29)

$$\begin{aligned}
&\bar{f}_i = f_{i-1} + f_i,\ \bar{m}_i = h_x - e_{i-1}, \bar{e}_i = e_{i-1} - e_i,\ \bar{n}_i = f_{i-1} - h_y\\
&w_i = b_i + \bar{f}_i,\ s_i = p_i + \bar{m}_i,\ v_i = a_i + \bar{e}_i,\ t_i = q_i + \bar{n}_i,\\
&|\bar{f}_j|, |\bar{m}_i|, |\bar{n}_i|, |\bar{e}_j| \le 2\Delta.
\end{aligned} \tag{30}$$

Using the equalities (30) the polynomial $\bar{L}_2^{(i)}$ becomes:

$$\begin{aligned}
&(b_i + \bar{f}_i)(p_i + \bar{m}_i) + (a_i + \bar{e}_i)(q_i + \bar{n}_i)\\
&= q_i\bar{e}_i + p_i\bar{f}_i + a_i\bar{n}_i + b_i\bar{m}_i + [\bar{m}_i\bar{f}_i + \bar{e}_i\bar{n}_i]\\
&= -(b_i p_i + a_i q_i)
\end{aligned}$$

Since $\bar{m}_i = h_x - e_{i-1} = \bar{m}_{i-1} + \bar{e}_{i-1}$, writing $\bar{e}_0 = 0$ and $\bar{m}_1 = \bar{m}$ then polynomial $\bar{L}_2^{(i)}$ for $i = 1, \ldots, k-1$ becomes:

$$q_i\bar{e}_i + p_i\bar{f}_i + a_i\bar{n}_i + b_i(\bar{m} + \sum_{j=0}^{i-1}\bar{e}_i) + [(\bar{m} + \sum_{j=0}^{i-1}\bar{e}_j)\bar{f}_i + \bar{e}_i\bar{n}_i] = -(b_i p_i + a_i q_i)$$

Now, we linearize this polynomial system. Writing, for $i = 1, \ldots, k-1$ and for $j = 1, \ldots, k-1$:

$$\begin{aligned}
&C^{(i)} \equiv -(b_i p_i + a_i q_i) \bmod p, \quad B_0^{(i)} \equiv b_i \bmod p,\\
&Q_j^{(i)} \equiv b_i \bmod p,\ (0 < j < i),\ Q_i^{(i)} \equiv q_i \bmod p,\ Q_j^{(i)} \equiv 0 \bmod p,\ (j > i)\\
&P_i^{(i)} \equiv p_i \bmod p,\ P_j^{(i)} \equiv 0 \bmod p,\ i \ne j\\
&A_i^{(i)} \equiv a_i \bmod p,\ A_j^{(i)} \equiv 0 \bmod p,\ i \ne j\\
&B_i^{(i)} \equiv 1 \bmod p,\ B_j^{(i)} \equiv 0 \bmod p,\ i \ne j
\end{aligned}$$

we obtain that vector $\mathbf{E} =$

$$\begin{aligned}
&(\Delta\bar{m}, \Delta\bar{e}_1, \ldots, \Delta\bar{e}_{k-1}, \Delta\bar{f}_1, \ldots, \Delta\bar{f}_{k-1}, \Delta\bar{n}_1, \ldots, \Delta\bar{n}_{k-1},\\
&\bar{m}\bar{f}_1 + \bar{e}_1\bar{n}_1, (\bar{m} + \bar{e}_1)\bar{f}_2 + \bar{e}_2\bar{n}_2, \ldots, (\bar{m} + \sum_{j=0}^{k-2}\bar{e}_j)\bar{f}_{k-1} + \bar{e}_{k-1}\bar{n}_{k-1})\\
&= (\Delta E_1, \Delta E_2, \ldots, \Delta E_{3k-2}, E_{3k-1}, \ldots, E_{4k-3})
\end{aligned}$$

is a solution to the following linear system of congruences ($i = 1, \ldots, k-1$):

$$\begin{aligned}
&B_0^{(i)}X_0 + \sum_{j=1}^{k-1}Q_j^{(i)}X_j + \sum_{j=1}^{k-1}P_j^{(i)}Y_j + \sum_{j=1}^{k-1}A_j^{(i)}Z_j + \sum_{j=1}^{k-1}\Delta B_j^{(i)}\Sigma_j \equiv \Delta C^{(i)} \bmod p,\\
&X_j \equiv 0 \bmod \Delta \quad (j = 0, \ldots, k-1)\\
&Y_j \equiv 0 \bmod \Delta, \quad Z_j \equiv 0 \bmod \Delta \quad (j = 1, \ldots, k-1).
\end{aligned} \tag{31}$$

Moreover, from (30) we have $\mathbf{E}$ is a relatively short vector. We have:

$$\|\mathbf{E}\| \leq 2(k+1)\Delta^2. \tag{32}$$

Let $\mathcal{L}_k$ be the lattice consisting of integer solutions

$$\mathbf{X} = (X_0, X_1, \ldots, X_{k-1}, Y_1, \ldots, Y_{k-1}, Z_1, \ldots, Z_{k-1}, \Sigma_1, \ldots, \Sigma_{k-1}) \in \mathbb{Z}^{4k-3}$$

of the system of congruences, $(i = 0, \ldots, k-1)$:

$$\begin{aligned} B_0^{(i)} X_0 + \sum_{j=1}^{k-1} Q_j^{(i)} X_j + \sum_{j=1}^{k-1} P_j^{(i)} Y_j + \sum_{j=1}^{k-1} A_j^{(i)} Z_j + \sum_{j=1}^{k-1} \Delta B_j^{(i)} \Sigma_j \equiv 0 \bmod p, \\ X_j \equiv 0 \bmod \Delta \quad (j = 0, \ldots, k-1) \\ Y_j \equiv 0 \bmod \Delta, \quad Z_j \equiv 0 \bmod \Delta \quad (j = 1, \ldots, k-1). \end{aligned} \tag{33}$$

We compute a solution $\mathbf{T}$ of the system of congruences (31), using linear diophantine equations methods. Applying an algorithm solving the CVP for the shift vector $\mathbf{T}$ and the lattice (33), we obtain a vector

$$\mathbf{F} = (\Delta F_1, \Delta F_2, \ldots, \Delta F_{3k-2}, F_{3k-1}, \ldots, F_{4k-3})$$

We have $\mathbf{F} = \mathbf{v} + \mathbf{T}$ (where $\mathbf{v}$ is the lattice vector returned by the CVP algorithm) is the vector of minimal norm satisfying (31), hence $\mathbf{F}$ must have norm at most equal to the norm of the solution $\mathbf{E}$. Using the bounds (32), we get:

$$\|\mathbf{F}\| \leq 2(k+1)\Delta^2. \tag{34}$$

Note that we can compute $\mathbf{F}$ in polynomial time from the information we are given, see Lemma 1. We might hope that $\mathbf{E}$ and $\mathbf{F}$ are the same, or at least, that we can recover the approximations errors from $\mathbf{F}$. This time, we are not giving a rigorous proof to bound the number of possibilities for which this method could fail. The volume of the lattice (33) is $p^{k-1}\Delta^{3k-2}$ (see Section 2.1) Then, by *Gaussian heuristic* and (34) vector $\mathbf{E}$ is likely to be the one founded whenever $\Delta^2 < p^{\frac{k-1}{4k-3}} \Delta^{\frac{3k-2}{4k-3}}$, this is:

$$\Delta < p^{\frac{k-1}{5k-4}}.$$

□

As we can see, if $k = 3$, we obtain from Theorem 4 that $O(\Delta) = p^{2/11}$ which is an improvement of Theorem 3.

5 Numerical results

We have proposed algorithms to recover a sequence of pseudorandom numbers produced by EC-LCG. The input required by all of them include approximations to some pseudorandom values. The first Theorem 1 and the second one Theorem 2 requires additionally precise knowledge of the parameter G. The rest require an approximation to the composer G. The quality of those approximations is the measure used to characterise when the algorithms output the expected sequence.

In Theorem 1 a "bad" set of values for the components x_0 is described, proving that whenever that value lies outside the set, the algorithm works correctly. Furthermore, the size of the set is asymptotically bounded with $O(\Delta^6)$. This means that if $\Delta < p^{1/6}$ and p is large enough, assuming a uniform distribution of probabilities for $x_0 \in \mathbb{F}_p$, the method is unlikely to fail. The same applies in Theorem 3 where the two "bad" subsets of $\mathbb{F}_p^6$ are

asymptotically bounded with $O(p^5\Delta^6)$, again it means that if $\Delta < p^{1/6}$ and p is large enough the method is unlikely to fail.

In Theorem 2 the heuristic algorithm requires k consecutive $\Delta-$approximations with $\Delta < p^{\frac{k-1}{4k-2}}$ when G is public. Finally, the heuristic method described in Theorem 4 when an approximation to composer G and k consecutive $\Delta-$approximations are given recovering the whole sequence if $\Delta < p^{\frac{k-1}{5k-4}}$

However, two aspects must be taken into account before considering those bounds as the threshold for the error tolerance upon which the algorithms fail. On the one side, the constants hidden in the asymptotic reasoning (namely, the size of the prime p). On the other one, the threshold could be higher, as the "bad" set does not guarantee that the methods indeed fails.

We have performed some numerical tests with a SageMath implementation of all methods. Firstly, we generate an elliptic curve over a prime finite field of a desired size by chossing randomly in $\mathbb{F}_p$ parameters a, b to fix (1). Then, we generate randomly points in the curve (1). For several pairs of points, an EC-LCG is simulated, and approximations to some consecutive values are given as input to our algorithms.

```
size_prime = 1024
p=next_prime(ZZ.random_element(2**size_prime))
a=ZZ.random_element(p); b=ZZ.random_element(p)
if (4*a**3+27*b**2)%p != 0:
    C =EllipticCurve(GF(p),[a,b])
G=C.random_element(); U_0=C.random_element()
U_1= U_0 + G
d=int(p**(0.14))
# We use the ZZ.random_element SageMath method
ZZ.random_element(-d+int(U_1[0]), d+int(U_1[0]))
```

And it is certified that both heuristic and deterministic methods confirm the obtained bounds. We show the numerical results of the heuristic algorithms, which, on the other hand, also include the deterministic ones. We summarize its results in the following tables. We have selected primes of several sizes, and note the obtained success threshold.

- Theorem 2: $\Delta-$approximations to k consecutive values and G public.
  - $k = 2$, $\Delta = O(p^{1/6})$, $\frac{1}{6} = 0.16666$. Lattice dimension: 5.

$\log_2(p)$	50	100	500	1000
$\log_p(\Delta)$	0.15	0.156	0.164	0.165

  - $k = 3$, $\Delta = O(p^{1/5})$, $\frac{1}{5} = 0.2$. Lattice dimension: 8

$\log_2(p)$	50	100	500	1000
$\log_p(\Delta)$	0.183	0.189	0.191	0.192

  - $k = 13$, $\Delta = O(p^{6/25})$, $\frac{6}{25} = 0.24$. Lattice dimension: 38

$\log_2(p)$	50	100	500	1000
$\log_p(\Delta)$	0.205	0.213	0.223	0.231

- Theorem 4: $\Delta-$approximation to G and to k consecutive values.
  - $k = 3,\ \Delta = O(p^{2/11}),\ \frac{2}{11} = 0.1818$. Lattice dimension: 9

$\log_2(p)$	50	100	500	1000
$\log_p(\Delta)$	0.161	0.169	0.171	0.179

  - $k = 13,\ \Delta = O(p^{12/61}),\ \frac{12}{61} = 0.1967$. Lattice dimension: 49

$\log_2(p)$	50	100	500	1000
$\log_p(\Delta)$	0.151	0.171	0.185	0.191

We have implemented the attack in SageMath-8.1 on a MacBook Pro laptop computer (3,3 GHz Intel Core i7, 16 GB RAM 2133 MHz LPDDR3 Mac OSX 10.12.6). Since the lattice dimension is very low -the biggest one is 49 which correspond the case $k = 13$ in Theorem 4- the time consuming in any trail is, as maximum, a couple of seconds.

6 Remarks and open questions

We have presented efficient algorithms for predicting the sequence produced by the linear congruential generator on elliptic curves. In fact, they only require computing a closest vector for a lattice of very low dimension, and for practical purposes can be used Babai's Nearest Plane algorithm, see [2].

Following the ideas in [9] by generating more non-linear equations by multiplication of several non-linear equations before the linearization step, papers [31] and [32] present theoretical better bound $O(p^{1/5})$ for Theorem 1 and $O(p^{\frac{3k}{11k+4}})$ for Theorem 2, under the heuristic assumption that the created polynomials define a zero dimensional ideal. Their algorithm need to compute the LLL algorithm of a certain lattice of huge dimension and, after that, it also requires a Groebner Basis computation or alternatively any other elimination polynomial method. In practice the performance of the so called of Coppersmith's methods in those cases are very bad because of large dimension of the lattice as it is shown in that papers. In fact, they can not test the bound $O(p^{\frac{3k}{11k+4}})$ not only because the large dimension but the size of the prime p should be several hundreds of bits. On the other hand, for instance, we can recover the sequence produced by EC-LCG if only three consecutive $\Delta-$approximations are given as soon as $\Delta < p^{1/5}$ requiring, the most time consuming, to compute a closest vector for a lattice of dimension 7 and it it matched by primes p of only 1000 bits.

As papers [31] and [32] show the bound of the size of the set of exceptional values of u_0 given in Theorem 1 is not tight and might be improved by more careful examination of the structure of (4) and (5) and this applies to Theorem 2.

Obviously the result in Theorem 3 is nontrivial only for $\Delta = O(p^{1/6})$, we believe that this can be improvement to $\Delta = O(p^{2/11})$ as Theorem 4 shows.

Giving rigorous proofs of our heuristic Theorem 2 and Theorem 4 is a challenging open question as well.

Same question has been studied in [32] for the Power Generator on elliptic curves: for a positive integer $e > 1$ and a point $G \in E(\mathbb{F}_p)$ of order l with $\gcd(e, l) = 1$, the elliptic curve Power Generator, (see [29]) generate a sequence of points V_n defined by the relation

$$V_n = [e^n]G$$

Requiring the prime p, integer e, constants a and b of the elliptic curve (1), and $\Delta-$approximations W_0, W_1 to two consecutive values V_0, V_1 for $\Delta < p^{\frac{1}{2e^2}}$. An improvement is presented in [14] recovering the root $U_0 = (x_0, y_0)$ of the polynomial (1) from a $\Delta-$approximation to (x_0, y_0) as soon as $\Delta < p^{1/7}$, considering the information given as approximation to the seed, in such a way that it is not taking advantage from the knowledge of the procedure which has generated them, see also [16]. We also think that better bounds are expected.

Another open problem is to mount an attack when the modulo p is unknown. Unfortunately, we do not know how to predict the EC-LCG when the modulus p is secret.

Finally, it would be interesting to study the security other PRBG based on elliptic curves under these type of attacks. In particular, it is not clear how to mount an attack based on the lattices to the Naor-Reingold Generator on Elliptic curves, see [36, 41, 42].

Acknowledgements The author wishes to thank Arne Winterhof for reading and commenting on a draft version.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

References

1. Avanzi, R., Cohen, H., Doche, C., Frey, G., Lange, T., Nguyen, K.K.: Elliptic and hyperelliptic curve crytography: Theory and practice. CRC Press (2005)
2. Babai, L.: On lovasz lattice reduction and the nearest lattice point problem. Combinatorica **6**, 1–6 (1986)
3. Beelen, P., Doumen, J.: Pseudorandom Sequences from Elliptic Curves. Finite Fields with Applications to Coding Theory Cryptography and Related Areas, pp. 37–52. Springer, Berlin (2002)
4. Blackburn, S., Gomez-Perez, D., Gutierrez, J., Shparlinski, I.: Predicting nonlinear pseudorandom number generators. Math. Comput. **74**, 1471–1494 (2005)
5. Blake, I., Seroussi, G., Smart, N.: Elliptic curves in cryptography. London Math. Soc., Lecture Note Series, vol. 265. Cambridge Univ Press (1999)
6. Bloemer, J., May, A.: A tool kit for Finding small roots of Bivariate Polynomial over the Integers. Advances in Cryptology-Crypt. LNCS 2729, pp. 27–43. Springer (2003)
7. Boyar, J.: Inferring sequences produced by pseudo-random number generators. J. ACM **36**, 129–141 (1989)
8. Brickell, E., Odlyzko, A.M.: Cryptanalysis: A survey of recent results. Contemp. cryptology, pp. 501–540. IEEE Press, NY (1992)
9. Coppersmith, D.: Small solutions to polynomial equations and low exponent RSA vulnerabilities. J. Cryptol. **10**(4), 233–260 (1997)
10. Coppersmith, D.: Finding a Small Root of a Bivariate Integer Equations; Factoring with High Bits Known'. In: Maurer, U. (ed.) Proc. EUROCRYPT-96, LNCS 1070,155–156. Springer, Berlin (1996)
11. Coron, J.S.: Finding small roots of Bivariate Integer Polynomial Equations Revisted. Proc. Advances in Cryptology- Eurocrypt'04, LNCS, 3027, pp. 492–505. Springer (2004)
12. El Mahassni, E., Shparlinski, I.: On the Uniformity of Distribution of Congruential Generators over Elliptic Curves. Proc. Intern. Conf. on Sequences and Their Applications, 257–264 Bergen 2001. Springer, London (2002)
13. Frieze, A., Håstad, J., Kannan, R., Lagarias J. C., Shamir, A.: Reconstructing truncated integer variables satisfying linear congruences. SIAM J. Comp. **17**, 262–280 (1988)

14. Gutierrez, J.: Reconstructing Points of Superelliptic Curves over a Prime Finite Field, Preprint. University of Cantabria (2021)
15. Gomez-Perez, D., Gutierrez, J., ibeas, A.: Attacking the Pollard Generator. IEEE. Trans. Inf. Theory. **52**(12), 5518–5523 (2006)
16. Gomez-Perez, D., Gutierrez, J.: Recovering zeros of polynomials modulo a prime'. Math. Comput. **83**(290), 2953–2965 (2014)
17. Gong, G., Berson, T., Stinson, D.: Elliptic Curve Pseudorandom Sequence Generators. Lect. Notes in Comp. Sci., vol. 1758, pp. 34–49. Springer, Berlin (2000)
18. Gong, G., Lam, C.: Linear recursive sequences over elliptic curves. Proc. Intern. Conf. on Sequences and their Applications, Bergen 2001, pp. 182–196. Springer, London (2002)
19. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer, Berlin (1993)
20. Gutierrez, J., Ibeas, A.: Inferring sequences produced by a linear congruential generator on elliptic curves missing high-order bits. Des. Codes Cryptogr. **45**(2), 199–212 (2007)
21. Hallgren, S.: Linear congruential generators over elliptic curves. Preprint CS-94-143, pp. 1–10. Dept. of Comp. Sci. Cornegie Mellon University (1994)
22. Hess, F., Shparlinski, I.: On the linear complexity and multidimensional distribution of congruential generators over elliptic curves. Des. Codes Cryptogr. **35**, 111–117 (2005)
23. Joux, A., Stern, J.: Lattice reduction: a toolbox for the cryptanalyst. J. Cryptol. **11**, 161–185 (1998)
24. Jochemz, E., May, A.: A Strategy for Finding Roots of Multivariate Polynomials with New Applications in Attacking RSA Variants Advances in Cryptology (Asiacrypt 2006), Lecture Notes in Computer Science. Springer (2006)
25. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Res. **12**, 415–440 (1987)
26. Knuth, D.: Deciphering a linear congruential encryption. IEEE Trans. Inf. Theory. **31**, 49–52 (1985)
27. Krawczyk, H.: How to predict congruential generators. J. Algorithm. **13**, 527–545 (1992)
28. Lagarias, J.: Pseudorandom number generators in cryptography and number theory, vol. 42, pp. 115–143. Proc. Symp. in Appl. Math. Amer. Math. Soc., Providence (1990)
29. Lange, T., Shparlinski, I.: Certain exponential sums and random walks on elliptic curves. Canad. J. Math. **57**, 338–350 (2005)
30. Lenstra, A., Lenstra, H., Lovász, L.: Factoring polynomials with rational coefficients. Math. Annalen. **261**, 515–534 (1982)
31. Mefenza, T.: Inferring sequences produced by a linear congruential generator on elliptic curves using coppersmith's methods. COCOON 2016 LNCS, vol. 9797, pp. 293–304 (2016)
32. Mefenza, T., Vergnaud, D.: Inferring sequences produced by a elliptic Curves generators y using Coppersmith's methods. Theor. Comput. Sci. **280**, 20–42 (2020)
33. Meidl, W., Winterhof, A.: Linear Complexity of sequences and multisequences. , pp. 324–334. Handbook of Finite Fields, CRC Press, Taylor and Francis, Group. edits G. Mullen and D. Panario (2013)
34. Mérai, L.: Remarks on pseudorandom binary sequences over elliptic curves. Fund. Inf. **114**(3-4), 301–308 (2012)
35. Mérai, L.: Predicting the elliptic curve congruential generator. Applicable Algebra in Engineering. Commun. Comput. **28**(3), 193–203 (2017)
36. Naor, M., Reingold, O.: Number theoretic constructions of efficient pseudo-random functions. Proc 38th IEEE Symp. on Found. of Comp. Sci., pp. 458–467. IEEE (1997)
37. Niederreiter, H.: Design and Analysis of Nonlinear Pseudorandom Number Generators. In: Schueller, G.I., Spanos, P.D. (eds.) Monte Carlo Simulation, pp. 3–9. A. Balkema Publishers, Rotterdam (2001)
38. Shparlinski, I.: Cryptographic applications of analytic number theory. Birkhauser (2003)
39. Shparlinski, I.: Orders of points on elliptic curves. Affine Algebraic Geometry. Amer. Math. Soc., pp. 245–252 (2005)
40. Shparlinski, I.: Pseudorandom Points on Elliptic Curves over Finite Fields. Recent trends in Cryptography. Contemporary Mathematics, v.477, Amer.Math. Soc., pp. 121–141 (2009)
41. Shparlinski, I.: On the Naor-Reingold pseudo-random function from elliptic curves. Appl. Algebra Engin. Commun. Comput. **11**, 27–34 (2000)
42. Shparlinski, I., Silverman, J.: On the linear complexity of the Naor-Reingold pseudorandom function from elliptic curves. Des. Codes Cryptogr. **24**, 279–289 (2001)

43. Silverman, J.: The Arithmetic of Elliptic Curves. Springer, Berlin (1995)
44. Sun, H., Zhu, X., Zheng, Q.: Predicting truncated multiple recursive generators with unknown parameters. Des. Codes Cryptogr., pp. 1–20 (2020)
45. Winterhof, A.: Recent results on recursive nonlinear pseudorandom number generators. (Invited Paper). SETA 2010: 113–124 (2010)