Inferring Sequences Produced by a Linear Congruential Generator on Elliptic Curves using Coppersmith's Methods

Thierry Mefenza

Cascade-DIENS, CNRS, INRIA, PSL, Paris, France, and
Department of Mathematics, University of Yaounde 1, Cameroon
thierrymefenza@yahoo.fr

Abstract. We analyze the security of the Elliptic Curve Linear Congruential Generator (EC-LCG). We show that this generator is insecure if sufficiently many bits are output at each iteration. In 2007, Gutierrez and Ibeas showed that this generator is insecure given a certain amount of most significant bits of some consecutive values of the sequence. Using the Coppersmith's methods, we are able to improve their security bounds.

Keywords. Elliptic Curve Linear Congruential Generator, Lattice reduction, Coppersmith's methods, Elliptic Curves.

1 Introduction

In cryptography, a pseudo-random number generator is a deterministic algorithm which takes as input a short random seed and outputs a long sequence which is indistinguishable in polynomial time from a truly random sequence. Pseudo-random numbers have found a number of applications in the literature. For instance they are useful in cryptography for key generation, encryption and signature. In 1994, Hallgren [Hal94] proposed a pseudo-random number generator based on a subgroup of points of an elliptic curve defined over a prime finite field. This generator is known as the Linear Congruential Generator on Elliptic Curves (EC-LCG). Let E be an elliptic curve defined over a prime finite field $\mathbb{F}_p$, that is a rational curve given by the following Weierstrass equation

$$E : y^2 = x^3 + ax + b$$

for some $a, b \in \mathbb{F}_p$ with $4a^3 + 27b^2 \neq 0$. It is well known that the set $E(\mathbb{F}_p)$ of $\mathbb{F}_p$-rational points (including the special point O at infinity) forms an Abelian group with an appropriate composition rule (denoted $\oplus$) where O is the neutral element. For a given point $G \in E(\mathbb{F}_p)$, the EC-LCG is a sequence U_n of points defined by the relation:

$$U_n = U_{n-1} \oplus G = nG \oplus U_0, \quad n \in \mathbb{N}$$

where $U_0 \in E(\mathbb{F}_p)$ is the initial value or seed. We refer to G as the *composer* of the generator. The EC-LCG provides a very attractive alternative to linear

and non-linear congruential generators and it has been extensively studied in the literature [Shp05,HS02,GL01,GBS00,MS02,BD02]. In cryptography, we want to use the output of the generator as a stream cipher. One can notice that if two consecutive values U_n, U_{n+1} of the generator are revealed, it is easy to find U_0 and G. So, we output only the most significant bits of each coordinate of U_n, $n \in \mathbb{N}$ in the hope that this makes the resulting output sequence difficult to predict. In this paper, we show that the EC-LCG is insecure if sufficiently many bits are output at each stage. Therefore a secure use of this generator requires to output fewer bits at each iteration and the efficiency of the schemes is thus degraded. Our attacks used the well-known Coppersmith's methods for finding small roots on polynomial equations. These methods have been introduced in 1996 by Coppersmith for polynomial of one or two variables [Cop96a,Cop96b] and have been generalized to many variables. These methods have been used to infer many pseudorandom generators and to cryptanalyze many schemes in cryptography (see [BCTV16,BVZ12] and the references therein). In this paper we used such techniques to improve the previous bounds known on the security of the EC-LCG in the literature. Our improvements are theoretical since in practice, the performance of Coppersmith's method in our case is bad because of large dimension of the lattice.

Prior work. In the cryptography setting, the initial value U_0 and the constants G, a and b may be kept secret. Gutierrez and Ibeas [GI07] consider two cases: the case where the *composer* G is known and a, b are kept secret and the case where the *composer* G is unknown and a, b are kept secret. In the first case, they showed that the EC-LCG is insecure if a proportion of at most $1/6$ of the least significant bits of two consecutive values of the sequence is hidden. When the *composer* is unknown, they showed heuristically that the EC-LCG is insecure if a proportion of at most $1/46$ of the least significant bits of three consecutive values of the sequence is hidden. Their result is based on a lattice basis reduction attack, using a certain linearization technique. In some sense, their technique can be seen as a special case of the problem of finding small solutions of multivariate polynomial congruences. The Coppersmith's methods also tackle the problem of finding small solutions of multivariate polynomial congruences. Gutierrez and Ibeas due to the special structure of the polynomials involved claimed that "the Coppersmith's methods does not seem to provide any advantages", and that "It may be very hard to give any precise rigorous or even convincing heuristic analysis of this approach". Our purpose in this paper is to tackle this issue.

Our contributions. We infer the EC-LCG sequence using Coppersmith's method for calculating the small roots of multivariate polynomials modulo an integer. The method for multivariate polynomials is heuristic since it is not proven and may fail (but in practice it works most of the time). At the end of the Coppersmith's methods we use the methods from [BCTV16] to analyze the success condition. In the case where the *composer* is known, we showed that the EC-LCG is insecure if a proportion of at most $1/5$ of the least significant bits of two consecutive values U_0 and U_1 of the sequence is hidden. This improves the previous bound $1/6$ of Gutierrez and Ibeas. We further improve this result by

considering several consecutive values of the sequence. We showed that the EC-LCG is insecure if a proportion of at most 3/11 of the least significant bits of these values is hidden. In the case where the *composer* is unknown, we showed that the EC-LCG is insecure if a proportion of at most 1/24 of the least significant bits of two consecutive values U_0 and U_1 of the sequence is hidden. This improves the previous bound 1/46 of Gutierrez and Ibeas. We further improve this result by considering sufficiently many consecutive values of the sequence. We showed that the EC-LCG is insecure if a proportion of at most 1/8 of the least significant bits of these values is hidden.

The table below gives a comparison between our results and those of Gutierrez and Ibeas. It gives the bound of the proportion of least significant bits hidden from each consecutive values necessary to break the EC-LCG in (heuristic) polynomial time. The basic proportion corresponds to the case where the adversary knows bits coming from the minimum number of intermediate values leading to a feasible attack; while the asymptotic proportion corresponds to the case when the bits known by the adversary knows bits coming from arbitrary number of values.

	Basic proportion		Asymptotic proportion	
	Prior result	Our result	Prior result	Our result
known *composer*	1/6	1/5	None	3/11
unknown *composer*	1/46	1/24	None	1/8

2 Preliminaries

For some $\Delta > 0$, 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 exists integers e, f satisfying:

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

Throughout the paper, $\Delta < p^\delta$, with $0 < \delta < 1$, corresponds to the situation where a proportion of at most δ of the least significant bits of the output sequence remain hidden.

2.1 The group law on elliptic curves

In this subsection, we recall the group law $\oplus$ on elliptic curves defined by the Weierstrass equation (for more details on elliptic curves, see [BSS99,Was08]), since our pseudorandom generator is defined recursively by adding a fixed composer G to the previous value. Let $E/\mathbb{F}_p : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbb{F}_p$. For two points $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$, with $P, Q \neq O$ the addition law $\oplus$ is defined as follows:

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

– If $x_P \neq x_Q$, then

$$x_R = m^2 - x_P - x_Q, \quad y_R = m(x_P - x_R) - y_P, \quad \text{where, } m = \frac{y_Q - y_P}{x_Q - x_P} \quad (1)$$

– If $x_P = x_Q$ but ($y_P \neq y_Q$ or $y_P = y_Q = 0$), then $R = O$
– If $P = Q$ and $y_P \neq 0$, then

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

2.2 Coppersmith's methods

In this section, we give a short description of Coppersmith's method for solving a multivariate modular polynomial system of equations modulo an integer N. We refer the reader to [JM06] for details and proofs.

Problem definition. Let $f_1(y_1, \ldots, y_n), \ldots, f_s(y_1, \ldots, y_n)$ be irreducible multivariate polynomials defined over $\mathbb{Z}$, having a root $(x_1, \ldots, x_n)$ modulo a known integer N, namely $f_i(x_1, \ldots, x_n) \equiv 0 \mod N$. We want this root to be *small* in the sense that each of its components is bounded by a known value X_i.

Polynomials collection. In a first step, one generates a collection $\mathfrak{P}$ of polynomials $\{\tilde{f}_1, \ldots, \tilde{f}_r\}$ linearly independent having $(x_1, \ldots, x_n)$ as a root modulo powers of N. Usually, multiples and powers of products of f_i, $i \in \{1, \ldots, s\}$ are chosen , namely $\tilde{f}_\ell = y_1^{\alpha_{1,\ell}} \cdots y_n^{\alpha_{n,\ell}} f_1^{k_{1,\ell}} \cdots f_s^{k_{s,\ell}}$ for some integers $\alpha_{1,\ell}, \ldots, \alpha_{n,\ell}$, $k_{1,\ell}, \ldots, k_{s,\ell}$ for $\ell \in \{1, \ldots, r\}$. Such polynomials satisfy the relation $\tilde{f}_\ell(x_1, \ldots, x_n) \equiv 0 \mod N^{\sum_{i=1}^{s} k_{i,\ell}}$, i.e., there exists an integer c_i such that $\tilde{f}_l(x_1, \ldots, x_n) = c_i N^{k_\ell}$, $k_\ell = \sum_{j=1}^{s} k_{j,\ell}$.

Monomials. We denote $\mathfrak{M}$ the set of monomials appearing in collection of polynomials $\mathfrak{P}$. Then each polynomial $\tilde{f}_i$ can be expressed as a vector with respect to a chosen order on $\mathfrak{M}$. We construct a matrix $\mathcal{M}$ and we define $\mathcal{L}$ the lattice generated by its rows. From that point, one computes an LLL-reduction on the lattice $\mathcal{L}$ and computes the Gram-Schmidt's orthogonalized basis of the LLL output basis. Extracting the coefficients appearing in the obtained vectors, one can construct polynomials defined over $\mathbb{Z}$ such that $\{p_1(x_1, \ldots, x_n) = 0, \ldots, p_n(x_1, \ldots, x_n) = 0\}$. Under the (heuristic) assumption that all created polynomials define an algebraic variety of dimension 0, the previous system can be solved (e.g., using elimination techniques such as Groebner basis) and the desired root recovered in polynomial time.

The conditions on the bounds X_i that make this method work are given by the following (simplified) inequation (see [JM06] for details):

$$\prod_{y_1^{k_1} \ldots y_n^{k_n} \in \mathfrak{M}} X_1^{k_1} \cdots X_n^{k_n} < N^{\sum_{\ell=1}^{r} \sum_{i=1}^{s} k_{i,\ell}} \quad . \quad (2)$$

For such techniques, the most complicated part is the choice of the collection of polynomials, what could be a really intricate task when working with multiple polynomials.

2.3 Analytic Combinatorics

In the following, we recall the analytic combinatorics methods [FS09] to count the exponents of the bounds $X_1, \ldots, X_n$ and of the modulo N on the monomials and polynomials appearing in the inequality (2) in Coppersmith's methods. Those methods can be used to compute the cardinalities of the sets $\mathfrak{P}$ and $\mathfrak{M}$. We used the same notations as in [BCTV16] and for more details of the methods the reader is referred to that paper. We see $\mathfrak{P}$ (respectively $\mathfrak{M}$) as a combinatorial class with size function $S(\tilde{f}_\ell) = \deg(\tilde{f}_\ell)$ (respectively $S(y_{\mathbf{k}}) = \deg(y_{\mathbf{k}})$, where $y_{\mathbf{k}} \in \mathfrak{M}$). We recall that a combinatorial class is a finite or countable set on which a size function is defined, satisfying the following conditions: (i) the size of an element is a non-negative integer and (ii) the number of elements of any given size is finite. We define another function χ, called a *parameter* function, such that $\chi(\tilde{f}_\ell) = k_\ell$ (respectively $\chi(y_{\mathbf{k}}) = k_i$, where k_i is the degree of the variable y_i in $y_{\mathbf{k}}$). This allows us to compute for some non negative integer t, ψ (respectively α_i) as:

$$\psi = \chi_{<t}(\mathfrak{P}) = \sum_{a \in \mathfrak{P}: S(a) < t} \chi(a) \quad \alpha_i = \chi_{<t}(\mathfrak{M}) = \sum_{a \in \mathfrak{P}: S(a) < t} \chi(a).$$

To do so we should be able to compute given a combinatorial class $\mathfrak{A}$ ($\mathfrak{A} = \mathfrak{P}$ or $\mathfrak{A} = \mathfrak{M}$) with size function S and the parameter function χ,

$$\chi_{\leqslant p}(\mathfrak{A}) = \sum_{a \in \mathfrak{A}: S(a) \leqslant p} \chi(a) \ .$$

We proceed as follows:

1. We give another description of $\mathfrak{A}$ with respect to S and χ. This description associates to the combinatorial class an ordinary generating function (OGF) $F(z, u)$ (using Table 1, see [BCTV16] for details). When the class contains elements of different sizes (such as variables of degree 1 and polynomials of degree e), the variables in the OGF are represented by the atomic element $\mathcal{Z}$ and the polynomials by the element $\mathcal{Z}^e$, in order to take into account the degree of these polynomials. Then we "mark" the element useful for the parameter, with a new variable u. At this level we only know how to compute $\sum_{a \in \mathfrak{A}: S(a) = p} \chi(a)$. An easier way to compute $\chi_{\leqslant p}(\mathfrak{A})$ is to force all elements a of size less than or equal to p to be of size exactly p by adding enough times a *dummy* element y_0 such that $\chi(y_0) = 0$. In our context of polynomials, the aim of the dummy variable y_0 is to homogenize the polynomial.
2. We have:

$$\chi_{\leqslant}(\mathfrak{A})(z) = \sum_{p=0}^{+\infty} \chi_{\leqslant p}(\mathfrak{A}) z^p = \left. \frac{\partial F(z, u)}{\partial u} \right|_{u=1},$$

Table 1. Combinatorics constructions and their OGF

	Construction	OGF
Atomic class	$\mathcal{Z}$	$Z(z) = z$
Neutral class	ε	$E(z) = 1$
Disjoint union	$\mathcal{A} = \mathcal{B} + \mathcal{C}$ (when $\mathcal{B} \cap \mathcal{C} = \emptyset$)	$A(z) = B(z) + C(z)$
Complement	$\mathcal{A} = \mathcal{B} \setminus \mathcal{C}$ (when $\mathcal{C} \subseteq \mathcal{B}$)	$A(z) = B(z) - C(z)$
Cartesian product	$\mathcal{A} = \mathcal{B} \times \mathcal{C}$	$A(z) = B(z) \cdot C(z)$
Cartesian exponentiation	$\mathcal{A} = \mathcal{B}^k = \mathcal{B} \times \cdots \times \mathcal{B}$	$A(z) = B(z)^k$
Sequence	$\mathcal{A} = \textsc{Seq}(\mathcal{B}) = \varepsilon + \mathcal{B} + \mathcal{B}^2 + \ldots$	$A(z) = \frac{1}{1-B(z)}$

3. Since Coppersmith's method is usually used in an asymptotic way, singularity analysis enables us to find the asymptotic value of the coefficients in an simple way by using the following theorem (see [FS09], page 392):

   Theorem 1 (Transfer Theorem). *Let $\mathfrak{A}$ be a combinatorial class with an ordinary generating function F regular enough such that there exists a value c verifying*

   $$F(z) = \sum_{n=0}^{+\infty} F_n z^n \underset{z \to 1}{\sim} \frac{c}{(1-z)^\alpha}$$

   for a non-negative integer α. The asymptotic value of the coefficient F_n is

   $$F_n \underset{n \to \infty}{\sim} (cn^{\alpha-1})/(\alpha-1)! \ .$$

3 Predicting EC-LCG Sequences for Known *Composer*

In the cryptographic setting, the initial value $U_0 = (x_0, y_0)$ and the constants G, a and b are supposed to be the secret key. In the following, we infer the EC-LCG sequence in the case where the *composer* G is known and the curve parameters are kept secret. We show that the generator is insecure if at least a proportion of $4/5$ of the most significant bits of two consecutive values U_0 and U_1 of the sequence is output.

Theorem 2. ***(two consecutive outputs)*** *Given Δ-approximations W_0, W_1 to two consecutive affine value U_0, U_1 produced by the EC-LCG, and given the value of the composer $G = (x_G, y_G)$. Under the heuristic assumption that all created polynomials we get by applying Coppersmiths method with the polynomial set $\mathfrak{P}$ below define an algebraic variety of dimension 0, one can recover the seed U_0 in heuristic polynomial time in $\log p$ as soon as $\Delta < p^\delta$, with $\delta < 1/5$.*

Proof. We suppose without loss of generality that $U_0 \notin \{-G, G\}$. Then, clearing denominators in (1), we can translate

$$U_1 = U_0 \oplus G$$

into the following identities in the field $\mathbb{F}_p$:

$$L_1 = L_1(x_0, y_0, x_1) = 0 \, mod \, p, \quad L_2 = L_2(x_0, y_0, x_1, y_1) = 0 \, mod \, p$$

where $U_0 = (x_0, y_0)$, $U_1 = (x_1, y_1)$ and

$$L_1 = x_G^3 + x_1 x_G^2 - x_0 x_G^2 - 2x_1 x_G x_0 - x_G x_0^2 + x_0^3 + 2y_G y_0 + x_1 x_0^2 - y_G^2 - y_0^2,$$

$$L_2 = y_1 x_G - y_1 x_0 - y_G x_0 + y_G x_1 - y_0 x_1 + y_0 x_G.$$

We denote $W_0 = (\alpha_0, \beta_0)$ and $W_1 = (\alpha_1, \beta_1)$. Then using the equalities $x_j = \alpha_j + e_j$ and $y_j = \beta_j + f_j$, for $j \in \{0, 1\}$, where $|e_j|, |f_j| < \Delta$ leads to the following polynomial system:

$$\begin{cases} f(e_0, e_1, f_0) = 0 \bmod p \\ g(e_0, e_1, f_0, f_1) = 0 \bmod p \ . \end{cases}$$

where $f(z_1, z_2, z_3) = A_1 z_1 + A_2 z_2 + A_3 z_3 + A_4 z_1^2 + A_5 z_1 z_2 + z_1^3 + z_1^2 z_2 - z_3^2 + A_6$ and $g(z_1, z_2, z_3, z_4) = B_1 z_1 + B_2 z_2 + B_3 z_3 + B_4 z_4 + z_1 z_4 + z_2 z_3 + B_5$ are polynomials whose coefficients A_i's and B_i's are functions of x_G, and the approximations values $\alpha_0, \alpha_1, \beta_0, \beta_1$. If we set $u_1 = z_1^3 + z_1^2 z_2 - z_3^2$ and $v_1 = z_1 z_4 + z_2 z_3$, then the polynomial f becomes $f_1(z_1, z_2, z_3, u_1) = A_1 z_1 + A_2 z_2 + A_3 z_3 + A_4 z_1^2 + A_5 z_1 z_2 + u_1 + A_6$ and g becomes $g_1(z_1, z_2, z_3, z_4, v_1) = B_1 z_1 + B_2 z_2 + B_3 z_3 + B_4 z_4 + v_1 + B_5$.

Description of the attack. The adversary is therefore looking for the small solutions of the following modular multivariate polynomial system:

$$\begin{cases} f_1(z_1, z_2, z_3, u_1) = 0 \bmod p \\ g_1(z_1, z_2, z_3, z_4, v_1) = 0 \bmod p \ . \end{cases}$$

With $|z_j| < \Delta$, $|u_1| < X = \Delta^3$ and $|v_1| < Y = \Delta^2$. The attack consists in applying Coppersmith's methods for multivariate polynomials. From now, we use the following collection of polynomials (parameterized by some integer $t \in \mathbb{N}$):

$$\mathfrak{P} = \left\{ z_1^{j_1} \dots z_4^{j_4} f_1^{i_1} g_1^{i_2} \bmod p^{i_1+i_2} \,:\, i_1 + i_2 > 0 \,\text{and}\, j_1 + \dots + j_4 + 2i_1 + i_2 < 2t \right\}$$

The list of monomials appearing within this collection can be described as:

$$\mathfrak{M} = \left\{ z_1^{i_1} z_2^{i_2} z_3^{i_3} z_4^{i_4} u_1^{i_5} v_1^{i_6} \bmod \Delta^{i_1+i_2+i_3+i_4} X^{i_5} Y^{i_6} \,:\, i_1 + \dots + i_4 + 2i_5 + i_6 < 2t \right\}.$$

If we use for instance the lexicography order on monomials, (with $z_1 < z_2 < z_3 < z_4 < u_1 < v_1$) on the set of monomials, then the leading monomial (denoted LM) of f_1 is $LM(f_1) = u_1$ and $LM(g_1) = v_1$. Then the polynomials in $\mathfrak{P}$ are linearly independent since we have prohibited the multiplication by u_1 and v_1.

Bounds for the Polynomials modulo p**.** We consider the set $\mathfrak{P}$ as a combinatorial class, with the size function $S(z_1^{j_1} \dots z_4^{j_4} f_1^{i_1} g_1^{i_2}) = j_1 + \dots + j_4 + 2i_1 + i_2$ and the parameter function $\chi(z_1^{j_1} \dots z_4^{j_4} f_1^{i_1} g_1^{i_2}) = i_1 + i_2$. The degree of each variable z_i, u_1, v_1 is 1, whereas the degree of f_1 is 2 and the degree of g_1 is 1. For

the sake of simplicity, we can consider $0 \leqslant i_1 + i_2$, since the parameter function equals 0 for elements $z_1^{j_1} \dots z_4^{j_4} f_1^{i_1} g_1^{i_2}$ with $i_1 + i_2 = 0$.
We can described $\mathfrak{P}$ as: $\prod_{i=1}^{4} \textsc{Seq}(Z) \times \textsc{Seq}(uZ^2) \times \textsc{Seq}(uZ) \times \textsc{Seq}(Z)$, where the last term is for the *dummy* value z_0.
This leads to the generating function:

$$F(z,u) = \left(\frac{1}{1-z}\right)^5 \times \frac{1}{1-uz^2} \times \frac{1}{1-uz}.$$

As $z \to 1$, $1 - z^n \sim n(1-z)$ leads to:

$$\left.\frac{\partial F}{\partial u}(u,z)\right|_{u=1} \underset{z\to 1}{\sim} \frac{3(1-z)}{4(1-z)^9} \sim \frac{3}{4(1-z)^8},$$

since $2t \sim 2t - 1$, this leads to: $\chi_{<2t}(\mathfrak{P}) \sim \frac{3}{4} \times \frac{(2t)^7}{7!}$

Bounds for the monomials modulo Δ. We consider the set $\mathfrak{M}$ as a combinatorial class, with the size function $S(z_1^{i_1} \dots z_4^{i_4} u^{i_5} v^{i_6}) = i_1 + \dots + i_4 + 2i_5 + i_6$ and the parameter function $\chi(z_1^{i_1} \dots z_4^{i_4} u^{i_5} v^{i_6}) = i_1 + \dots + i_4$. As z_1, z_2, z_3, z_4, u, v "count for" $1, 1, 1, 1, 2$ and 1 respectively in the condition of the set, we can described $\mathfrak{M}$ as: $\textsc{Seq}(Z^2) \times \textsc{Seq}(Z) \times \prod_{i=1}^{4} \textsc{Seq}(uZ) \times \textsc{Seq}(Z)$, where the last term is for the *dummy* value z_0.
Which leads to the generating function: $F(z,u) = \frac{1}{(1-z^2)(1-z)^2} \times \left(\frac{1}{1-uz}\right)^4$. As previously, we obtain $\chi_{<2t,\Delta}(\mathfrak{M}) \sim \frac{2(2t)^7}{7!}$.

Bounds for the monomials modulo X (respectively modulo Y). We consider the set $\mathfrak{M}$ as a combinatorial class, with the size function $S(z_1^{i_1} \dots z_4^{i_4} u^{i_5} v^{i_6}) = i_1 + \dots + i_4 + 2i_5 + i_6$ and the parameter function $\chi(z_1^{i_1} \dots z_4^{i_4} u^{i_5} v^{i_6}) = i_5$ (respectively $\chi(z_1^{i_1} \dots z_4^{i_4} u^{i_5} v^{i_6}) = i_6$). As z_1, z_2, z_3, z_4, u, v "count for" $1, 1, 1, 1, 2$ and 1 respectively in the condition of the set, we can described $\mathfrak{M}$ as: $\prod_{i=1}^{5} \textsc{Seq}(Z) \times \textsc{Seq}(uZ^2) \times \textsc{Seq}(Z)$ (respectively $\prod_{i=1}^{4} \textsc{Seq}(Z) \times \textsc{Seq}(Z^2) \times \textsc{Seq}(uZ) \times \textsc{Seq}(Z)$) where the last one is for the *dummy* value z_0.
Which leads to the generating function: $F(z,u) = \frac{1}{(1-z)^6} \times \frac{1}{1-uz^2}$ (respectively $F(z,u) = \frac{1}{(1-z)^5(1-z^2)} \times \frac{1}{1-uz}$). This leads to: $\chi_{<2t,X}(\mathfrak{M}) \sim \frac{(2t)^7}{4\times 7!}$ (respectively $\chi_{<2t,Y}(\mathfrak{M}) \sim \frac{(2t)^7}{2\times 7!}$).

Condition. We denote by $\nu_1 = \chi_{<2t,\Delta}(\mathfrak{M})$, $\nu_2 = \chi_{<2t,X}(\mathfrak{M})$, $\nu_3 = \chi_{<2t,Y}(\mathfrak{M})$ and $\varepsilon = \chi_{<2t}(\mathfrak{P})$. The inequality (2) is $p^{\varepsilon} > \Delta^{\nu_1} X^{\nu_2} Y^{\nu_3}$, ie $\Delta < p^{\frac{\varepsilon}{\nu_1 + 3\nu_2 + 2\nu_3}}$, where:

$$\frac{\varepsilon}{\nu_1 + 3\nu_2 + 2\nu_3} \sim \frac{\chi_{<2t}(\mathfrak{P})}{\chi_{<2t,\Delta}(\mathfrak{M}) + 3\chi_{<2t,X}(\mathfrak{M}) + 2\chi_{<2t,Y}(\mathfrak{M})} \sim \frac{1}{5},$$

this leads to the claimed bound: $\Delta < p^{\frac{1}{5}}$. □

This bound improves the known bound $\Delta < p^{1/6}$. Next we further improve the previous bound and we show that the generator is insecure if at least a proportion of 8/11 of the most significant bits of an arbitrary large number of consecutive values U_i of the sequence is output.

Theorem 3. ***(more consecutive outputs)*** *Given Δ-approximations W_0, W_1,...,W_n (for some integer $n > 1$) to $n+1$ consecutive affine values U_0, U_1,...,U_n produced by the EC-LCG, and given the value of the composer $G = (x_G, y_G)$. Under the heuristic assumption that all created polynomials we get by applying Coppersmiths method with the polynomial set $\mathfrak{P}$ below define an algebraic variety of dimension 0, one can recover the seed U_0 in polynomial time in $\log p$ as soon as $\Delta < p^{\delta}$, with $\delta < \frac{3n}{11n+4}$.*

Proof. (Sketch) We can generalize the previous proof by considering n couples of consecutive values (U_i, U_{i+1}), $i \in \{0, \ldots, n-1\}$ and the same variable change to get n couple of polynomials f_{i+1}, g_{i+1} of the same shape as f_1 and g_1. We then apply the method to the following collection of polynomials:

$$\mathfrak{P} = \left\{ \begin{array}{l} z_0^{j_0} \ldots z_{2n+1}^{j_{2n+1}} f_1^{i_1} \ldots f_n^{i_n} g_1^{l_1} \ldots g_n^{l_n} \bmod p^{i_1+l_1\cdots+i_n+l_n} \\ \text{s.t. } i_1 + l_1 + \cdots + i_n + l_n > 0 \\ \text{and } j_0 + \cdots + j_{2n+1} + 2(i_1 + \cdots + i_n) + l_1 + \cdots + l_n < 2t \end{array} \right\},$$

and the following set of monomials:

$$\mathfrak{M} = \left\{ \begin{array}{l} z_0^{j_0} \ldots z_{2n+1}^{j_{2n+1}} u_1^{i_1} v_1^{l_1} \ldots u_n^{i_n} v_n^{l_n} \bmod \Delta^{j_0+\cdots+j_{2n+1}} X^{i_0+\cdots+i_n} Y^{l_0+\cdots+l_n} \\ \text{s.t. } j_0 + \cdots + j_{2n+1} + 2(i_1 + \cdots + i_n) + l_1 + \cdots + l_n < 2t \end{array} \right\},$$

to get the result (see the full version of the paper for the complete proof). □

4 Predicting EC-LCG Sequences for Unknown Composer

In this section, we infer the EC-LCG sequence in the case where the *composer* G is unknown and the curve parameters are kept secret. In the following, We show that the generator is insecure if at least a proportion of 23/24 of the most significant bits of three consecutive values U_0 and U_1 and U_2 of the sequence is output.

Theorem 4. ***(three consecutive outputs)*** *Given Δ-approximations W_0, W_1, W_2 to three consecutive affine values U_0, U_1, U_2 produced by the EC-LCG. Under the heuristic assumption that all created polynomials we get by applying Coppersmiths method with the polynomial set $\mathfrak{P}$ below define an algebraic variety of dimension 0, one can recover the seed U_0 and the composer G in polynomial time in $\log p$ as soon as $\Delta < p^{\delta}$ with $\delta < 1/24$.*

Proof. We set $U_0 = (x_0, y_0)$, $U_1 = (x_1, y_1)$, $U_2 = (x_2, y_2)$, $W_0 = (\alpha_0, \beta_0)$, $W_1 = (\alpha_1, \beta_1)$ and $W_2 = (\alpha_2, \beta_2)$. We then have the equalities:

$$x_i = \alpha_i + e_i,\ y_j = \beta_j + f_j, \quad \text{where} \quad |e_i|, |f_i| < \Delta,\ i \in \{0, 1, 2\}. \tag{3}$$

We also have:

$$\begin{cases} y_0^2 = x_0^3 + ax_0 + b \\ y_1^2 = x_1^3 + ax_1 + b \\ y_2^2 = x_2^3 + ax_2 + b \ . \end{cases}$$

Eliminating the curve parameters a, b and assuming without loss of generality that $U_2 \neq \pm U_1$ (that is, $x_2 \neq x_1$), we obtain the following equation:

$$y_2^2(x_0-x_1)+x_2^3(x_1-x_0)+x_0^3(x_2-x_1)+y_0^3(x_1-x_2)+x_1^3(x_0-x_2)+y_1^2(x_2-x_0) = 0$$

Using the equalities (3), leads to the equation:

$$f(e_0, e_1, e_2, f_0, f_1, f_2) = 0 \bmod p$$

where f is a polynomial of degree 4 whose coefficients are functions of $\alpha_0, \alpha_1, \alpha_2$, β_0, β_2, and β_2.

Description of the attack. The adversary is therefore looking for the solutions smaller than Δ of the following modular multivariate polynomial equation:

$$f(z_1, \ldots, z_6) = 0 \bmod p$$

The attack consists in applying Coppersmith's methods as in the former subsection. If we consider monomials with respect to lexicographic order, then the leading monomial of f is $z_1^3 z_2$. From now on, we use the following collection of polynomials:

$$\mathfrak{P} = \{\tilde{f}_{j_1,\ldots,j_6,i} = z_1^{j_1} \ldots z_6^{j_6} f^i \bmod p^i : i > 0 \quad \text{and} \quad j_1 + \cdots + j_6 + 4i < 4t \quad \text{and } (0 \leqslant j_1 < 3 \vee j_2 = 0)\},$$

One can check that the polynomials $\tilde{f}_{j_1,\ldots,j_6,i}$ are linearly independent since $LM(f) \neq z_1^{j_1} \ldots z_6^{j_6}$ for each $\tilde{f}_{j_1,\ldots,j_6,i}$ from $\mathfrak{P}$. The list of monomials appearing within this collection can be described as:

$$\mathfrak{M} = \left\{ z_1^{j_1} \ldots z_6^{j_6} \bmod \Delta^{j_1+\cdots+j_6} : j_1 + \cdots + j_6 < 4t \right\}.$$

Bounds for the Polynomials modulo p. We consider the set $\mathfrak{P}$ as a combinatorial class, with the size function $S(\tilde{f}_{j_1,\ldots,j_6,i}) = j_1 + \cdots + j_6 + 4i$ and the parameter function $\chi(\tilde{f}_{j_1,\ldots,j_6,i}) = i$. Since the degree of each variable z_i is 1 and the degree of f is 4, we can described $\mathfrak{P}$ as:

$$\prod_{i=1}^{4} \textsc{Seq}(Z) \times \textsc{Seq}(uZ^4) \times \left(\underbrace{(\varepsilon + Z + Z^2)}_{z_1} \underbrace{(\varepsilon + Z\textsc{Seq}(Z))}_{z_2} + \underbrace{Z^3\textsc{Seq}(Z)}_{z_1} \right) \times \textsc{Seq}(Z),$$

where the last term is for the *dummy* value z_0. This leads to the generating function:

$$F(z,u) = \left(\frac{1}{1-z}\right)^5 \times \frac{1}{1-uz^4} \times \left((1+z+z^2)(1+z/(1-z)) + \frac{z^3}{1-z} \right).$$

This leads to: $\chi_{<4t}(\mathfrak{P}) \sim \frac{1}{4} \times \frac{(4t)^7}{7!}$

Bounds for the monomials modulo Δ. We consider the set $\mathfrak{M}$ as a combinatorial class, with the size function $S(z_1^{j_1} \dots z_6^{j_6}) = j_1 + \dots + j_6$ and the parameter function $\chi(z_1^{j_1} \dots z_6^{j_6}) = j_1 + \dots + j_6$. Since the degree of each z_i is 1, we can then described $\mathfrak{M}$ as: $\prod_{i=1}^{6} \textsc{Seq}(uZ) \times \textsc{Seq}(Z)$, where the last term is for the *dummy* value z_0. Which leads to the generating function: $F(z,u) = \left(\frac{1}{1-uz}\right)^6 \times \frac{1}{1-z}$. We then obtain: $\chi_{<4t}(\mathfrak{M}) \sim \frac{6(3t)^7}{7!}$

Condition. If we denote by $\nu = \chi_{<4t}(\mathfrak{P})$, and $\varepsilon = \chi_{<4t}(\mathfrak{M})$, the inequality (2) is $p^\nu > \Delta^\varepsilon$, ie $\Delta < p^{\frac{\nu}{\varepsilon}}$, where: $\frac{\nu}{\varepsilon} \sim \frac{\chi_{<4t}(\mathfrak{P})}{\chi_{<4t}(\mathfrak{M})} \sim \frac{1}{24}$, this leads to the claimed bound: $\Delta < p^{\frac{1}{24}}$. □

This bound improves the known bound $\Delta < p^{1/46}$. Next, we further improve the previous bound and we show that the generator is insecure if at least a proportion of 7/8 of the most significant bits of an arbitrary large number of consecutive values U_i of the sequence is output.

Theorem 5. ***(more consecutive outputs)***
Given Δ-approximations W_0, W_1,...,W_{n+1} (for some integer $n > 1$) to $n+2$ consecutive affine values U_0, U_1,...,U_{n+1} produced by the EC-LCG. Under the heuristic assumption that all created polynomials we get by applying Coppersmiths method with the polynomial set $\mathfrak{P}$ below define an algebraic variety of dimension 0, one can recover the seed U_0 and the composer G in polynomial time in $\log p$ as soon as $\Delta < p^\delta$ with $\delta < n/4(2n+4)$.

Proof. See the full version of the paper. □

5 Conclusion

We analyzed the security of the Elliptic Curve Linear Congruential Generator (EC-LCG). In the case where the *composer* is known, we showed that this generator is insecure if at least a proportion of 8/11 of the most significant bits of an arbitrary large number of consecutive values U_i of the sequence is output. We also consider the cryptographic setting where the *composer* is unknown and we showed that this generator is insecure if at least a proportion of 7/8 of the most significant bits of an arbitrary large number of consecutive values U_i of the sequence is output. Our results are theoretical since in practice, the performance of Coppersmith's method in our attacks is bad because of large dimension of the constructed lattice but they are good evidences of the weaknesses of this generator. This generator should then be used with great care.

Acknowledgments The author was supported in part by the French ANR JCJC ROMAnTIC project (ANR-12-JS02-0004) and by the Simons foundation Pole PRMAIS. I would like to thank anonymous referees for their helpful comments.

References

[BCTV16] F. Benhamouda, C. Chevalier, A. Thillard, and D. Vergnaud. Easing coppersmith methods using analytic combinatorics: Applications to public-key cryptography with weak pseudorandomness. In C. -M. Cheng, et al, editors, *PKC 16*, PartII, volume 9615 of *Lect. Notes Comput. Sci.*, pages 36–66, 2016.

[BD02] P. Beelen, and J. Doumen. Pseudorandom sequences from elliptic curves. Finite fields with applications to coding theory. *Cryptography and related areas.Springer-Verlag, Berlin, pages 37–52, 2002.*

[BSS99] I. F. Blake, G. Seroussi, and N. P. Smart. *Elliptic curves in cryptography.* Cambridge: Cambridge University Press, 1999.

[BVZ12] A. Bauer, D. Vergnaud, and J-C. Zapalowicz. Inferring sequences produced by nonlinear pseudorandom number generators using Coppersmith's methods. In M. Fischlin, J. Buchmann, and M. Manulis, editors, *PKC 12*, volume 7293 of *Lect. Notes Comput. Sci.*, pages 609–626, 2012.

[Cop96a] D. Coppersmith. Finding a small root of a univariate modular equation. In U. M. Maurer, editor, *EUROCRYPT 96*, volume 1070 of *Lect. Notes Comput. Sci.*, pages 155–165, 1996.

[Cop96b] D. Coppersmith. Finding a small root of a bivariate integer equation; factoring with high bits known. In U. M. Maurer, editor, *EUROCRYPT 96*, volume 1070 of *Lect. Notes Comput. Sci.*, pages 178–189, 1996.

[FS09] P. Flajolet and R. Sedgewick. *Analytic Combinatorics.* Cambridge University Press (January 2009).

[GBS00] G. Gong, T. A. Berson, and D. A. Stinson. Elliptic curve pseudorandom sequence generators. volume 1758 of *Lect. Notes Comput. Sci.*, pages 34–49, 2000.

[GI07] J. Gutierrez and A. Ibeas. Inferring sequences produced by a linear congruential generator on elliptic curves missing high-order bits. *Design Code Cryptogr*, 45:199–212, 2007.

[GL01] G. Gong and C. C. Y. Lam. Linear recursive sequences over elliptic curves. *In: Proc. intern. conf. on sequences and their applications, Bergen 2001.* Springer-Verlag, London, pages 182–196, 2001.

[Hal94] S. Hallgren. Linear congruential generators over elliptic curves. *Preprint CS-94-143, Dept. of Comp. Sci., 1994.*

[HS02] F. Hess and I. E. Shparlinski. On the linear complexity and multidimensional distribution of congruential generators over elliptic curves. *Design Code Cryptogr*, 35:111–117, 2005.

[JM06] E. Jochemsz, and A. May. A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants. In X. Lai, K. Chen, editors, *ASIACRYPT 06*, volume 4284 of *Lect. Notes Comput. Sci.*, pages 267–282, 2006.

[MS02] E. Mahassni and I. E. Shparlinski. On the uniformity of distribution of congruential generators over elliptic curves. *In: Proc. intern. conf. on sequences and their applications.* Bergen 2001. Springer-Verlag, London , pages 257–264, 2002.

[Shp05] I. E. Shparlinski. Pseudorandom points on elliptic curves over finite fields. *Preprint*, 2005.

[Was08] L. C. Washington. *Elliptic curves. Number theory and cryptography. 2nd ed.* Boca Raton, FL: Chapman and Hall/CRC, 2nd ed. edition, 2008.