Attacking the Pollard Generator

Domingo Gómez, Jaime Gutierrez, and Álvar Ibeas

Abstract— **Let p be a prime and let c be an integer modulo p. The Pollard generator (PG) is a sequence (u_n) of pseudo-random numbers defined by the relation $u_{n+1} \equiv u_n^2 + c \bmod p$. It is shown that if c and $9/14$ of the most significant bits of two consecutive values u_n, u_{n+1} of the PG are given, one can recover in polynomial time the initial value u_0 with a probabilistic algorithm. This result is an improvement of a theorem in a recent paper which requires that $2/3$ of the most significant bits be known.**

Keywords— **Pseudorandom numbers, Pollard generator, Lattice reduction.**

I. Introduction

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

For fixed $c \in \mathbb{F}_p$, the *Pollard Generator* (u_n) of elements of $\mathbb{F}_p$ is given by the recurrence relation

$$u_{n+1} \equiv u_n^2 + c \bmod p, \qquad n = 0, 1, \ldots, \tag{1}$$

where u_0 is the *initial value*.

This generator has many interesting applications in cryptography (see the surveys [5] and [14]). In the cryptographic setting, the initial value u_0 and the *shift* constant c are assumed to be the secret key, and the output of the generator is used as a stream cipher. Of course, if two consecutive values u_n, u_{n+1} are revealed, it is easy to find u_0 and c. So in this setting, we output only the most significant bits of each u_n in the hope that this makes the resulting output sequence difficult to predict. The recent paper [3] shows that not too many bits can be output at each stage: the Pollard generator is unfortunately polynomial time predictable if sufficiently many bits of some consecutive elements are revealed, whenever a small number of secret keys is excluded. Extensions and variants of this problem can be found in [2], [4], [6].

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. To simplify the notation, we assume that $n = 0$ from now on. Then, [3] shows that the values u_{n+j} can be recovered from this information in polynomial time if the approximations w_j are sufficiently good and if a certain small set of initial values u_0 is excluded. Throughout the paper the term polynomial time means polynomial in $\log p$. The previous result [3, Theorem 4] involves another parameter Δ which measures how well the values w_j approximate the terms u_j.

More precisely, we say that w is a Δ-*approximation* to u if $|w - u| \leq \Delta$. So, the case where Δ grows like a fixed power p^δ with $0 < \delta < 1$, corresponds to the situation in which a proportion δ of the least significant bits of terms of the output sequence remains hidden.

Theorem 1 ([3], Thm. 4) Let p be a prime number and let Δ be an integer such that $p > \Delta \geq 1$. For any $c \in \mathbb{F}_p$, there exists a set $\mathcal{V}(\Delta; c) \subseteq \mathbb{F}_p$ of cardinality $\#\mathcal{V}(\Delta; c) = O(\Delta^3)$ with the following property: there exists an algorithm which, when given Δ-approximations w_j, $j = 0, 1$, to two consecutive values u_0, u_1 produced by the Pollard generator (1), returns the value u_0 in deterministic polynomial time if $u_0 \notin \mathcal{V}(\Delta; c)$.

In other words, the Pollard generator is polynomial time predictable if $2/3$ bits of two consecutive elements are revealed. The present paper improves the above result showing that we only need to know $9/14$ of the most significant bits in order to predict the Pollard generator:

Theorem 2 (Main Result) Let p be a prime number and let Δ be an integer such that $p > \Delta \geq 1$. For any $c \in \mathbb{F}_p$, there exists a set $\mathcal{U}(\Delta; c) \subseteq \mathbb{F}_p \times [-\Delta, \Delta]$ of cardinality $\#\mathcal{U}(\Delta; c) = O\left(\max\{\Delta^{15} p^{-4}, \Delta^{19/5}\}\right)$ with the following property: there exists an algorithm which, when given Δ-approximations w_j, $j = 0, 1$, to two consecutive values u_0, u_1 produced by the Pollard generator (1), returns the value u_0 in deterministic polynomial time if $(u_0, u_0 - w_0) \notin \mathcal{U}(\Delta; c)$.

In order to prove this result, we introduce some modifications and additions to the method of [3] which allow us to attack the generators knowing fewer bits of two consecutive elements u_0 and u_1. We demonstrate our technique in the special case when c is public. Of course, this assumption reduces the relevance of the problem to cryptography. We do, however, believe that the extra strength of the result we obtain makes this situation of interest in its own right. This approach can be extended to the case when c is secret. This matter is dealt in Section IV.

The remainder of the paper is structured as follows:

We start with a short outline of some basic facts about lattices in Section II-A and some auxiliary results in Section II-B. Section III is dedicated to proving the main result. In Section IV we explain results obtained when trying to extend the algorithm to instances with private shift c. Finally, Section V makes some final comments and poses some open questions.

This work is partially supported by the Spanish Ministry of Science grant MTM2004-07086.

The authors are with the Faculty of Sciences, University of Cantabria, E-39071 Santander, Spain (e-mail: domingo.gomez@.unican.es; jaime.gutierrez@unican.es; alvar.ibeas@unican.es)

II. Lattices and auxiliary results

A. Background on Lattices

Here we collect several well-known facts about lattices which form the background to our algorithms.

We review several results and definitions of concepts related to lattices which can be found in [8]. For more details and more recent references, we also recommend consulting [1], [10], [11], [16], [17], [18].

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

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

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

To each lattice $\mathcal{L}$ one can naturally associate its *volume*:

$$\mathrm{vol}\,(\mathcal{L}) = \left(\det\left(\langle \mathbf{b}_i, \mathbf{b}_j\rangle\right)_{i,j=1}^{s}\right)^{1/2},$$

where $\langle \mathbf{a}, \mathbf{b}\rangle$ denotes the inner product. This definition does not depend on the choice of the basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_s\}$.

For a vector $\mathbf{u}$, let $\|\mathbf{u}\|$ denote its *Euclidean norm*. The first Minkowski theorem, see Theorem 5.3.6 in [8], gives the upper bound

$$\min\{\|\mathbf{z}\| : \ \mathbf{z} \in \mathcal{L} \setminus \{\mathbf{0}\}\} \leq s^{1/2}\,\mathrm{vol}\,(\mathcal{L})^{1/s} \qquad (2)$$

on the shortest nonzero vector in any s-dimensional lattice $\mathcal{L}$ in terms of its volume.

The Minkowski bound (2) motivates a natural question, the *Shortest Vector Problem (SVP)*: how to find a shortest nonzero vector in a lattice. Unfortunately, there are several indications that this problem is **NP**-hard when the dimension grows. This study has suggested several definitions of a *reduced* basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_s\}$ for a lattice, trying to obtain a shortest vector by the first basis element $\mathbf{b}_1$. The celebrated *LLL algorithm* of Lenstra, Lenstra and Lovász [15] provides a concept of *reduced* basis and an approximate solution, enough in many practice applications.

The basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_s\}$ obtained with this method satisfies the two following conditions: when computing the corresponding Gram-Schmidt orthogonal basis $\{\mathbf{b}_1^*, \ldots, \mathbf{b}_s^*\}$ following the usual algorithm

$$\mathbf{b}_i^* := \mathbf{b}_i - \sum_{j<i} \mu_{ij}\mathbf{b}_j, \ \mu_{ij} := \frac{<\mathbf{b}_i, \mathbf{b}_j^*>}{<\mathbf{b}_j^*, \mathbf{b}_j^*>},$$

all the appearing coefficients satisfy $|\mu_{ij}| \leq 1/2$. Besides this, if one contemplates the projections π_i from the space $\mathbb{R}^r$ orthogonally onto $<\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}>^{\perp}$, it must be satisfied that $\delta\|\pi_i(\mathbf{b}_i)\|^2 \leq \|\pi_i(\mathbf{b}_{i+1})\|^2$, $i = 1, \ldots, n-1$, where $1/4 < \delta < 1$ is a fixed parameter on which depends the reduction procedure.

Another related question is the *Closest Vector Problem (CVP)*: given a lattice $\mathcal{L} \subseteq \mathbb{R}^r$ and a shift vector $\mathbf{t} \in \mathbb{R}^r$, the goal consists in finding a vector in the set $\mathbf{t} + \mathcal{L}$ with minimum norm. This problem is usually expressed in an equivalent way: finding a vector in $\mathcal{L}$ closest to the target vector $-\mathbf{t}$. It is well known that CVP is **NP**-hard when the dimension grows.

However, both computational problems SVP and CVP are known to be solvable in deterministic polynomial time provided that the dimension of $\mathcal{L}$ is fixed (see [12], for example). The lattices in this paper are of fixed (and low) dimension.

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 $\{\mathbf{b}_1, \ldots, \mathbf{b}_s\}$ is a basis of the above lattice, by the Hadamard inequality we have:

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

B. Auxiliary results

In lattices of rank 2, one basis $\{\mathbf{u}, \mathbf{v}\}$ is typically called *reduced* (see for instance [13]), if and only if

$$\|\mathbf{u}\|, \|\mathbf{v}\| \leq \|\mathbf{u} + \mathbf{v}\|, \|\mathbf{u} - \mathbf{v}\|.$$

Now we present a technical result for later use.

Lemma 3: Let $\{\mathbf{u}, \mathbf{v}\} \subset \mathbb{R}^r$ be a reduced basis of a 2-rank lattice $\mathcal{L}$ and $\mathbf{x} \in \mathcal{L}$. The unique integers $(\alpha, \beta) \in \mathbb{Z}^2$ satisfying $\mathbf{x} = \alpha\mathbf{u} + \beta\mathbf{v}$ also satisfy:

$$\|\alpha\mathbf{u}\|, \|\beta\mathbf{v}\| \leq \frac{2}{\sqrt{3}}\|\mathbf{x}\|.$$

Proof: Since $\{\mathbf{u}, \mathbf{v}\}$ is a reduced basis, the angle between the lines containing its vectors is greater than $\pi/3$:

$$\|\mathbf{u}\|^2, \|\mathbf{v}\|^2 \leq \|\mathbf{u} + \mathbf{v}\|^2, \|\mathbf{u} - \mathbf{v}\|^2$$

$$\langle \mathbf{u}, \mathbf{u}\rangle, \langle \mathbf{v}, \mathbf{v}\rangle \leq \left\{ \begin{array}{l} \langle \mathbf{u}, \mathbf{u}\rangle + \langle \mathbf{v}, \mathbf{v}\rangle + 2\langle \mathbf{u}, \mathbf{v}\rangle \\ \langle \mathbf{u}, \mathbf{u}\rangle + \langle \mathbf{v}, \mathbf{v}\rangle - 2\langle \mathbf{u}, \mathbf{v}\rangle \end{array} \right.$$

$$\left. \begin{array}{l} -\langle \mathbf{u}, \mathbf{u}\rangle \\ -\langle \mathbf{v}, \mathbf{v}\rangle \end{array} \right\} \leq 2\langle \mathbf{u}, \mathbf{v}\rangle \leq \left\{ \begin{array}{l} \langle \mathbf{u}, \mathbf{u}\rangle \\ \langle \mathbf{v}, \mathbf{v}\rangle \end{array} \right.$$

$$2|\langle \mathbf{u}, \mathbf{v}\rangle| \leq \langle \mathbf{u}, \mathbf{u}\rangle, \langle \mathbf{v}, \mathbf{v}\rangle.$$

Therefore,

$$\frac{\langle \alpha\mathbf{u}, \alpha\mathbf{u}\rangle}{\langle \alpha\mathbf{u} + \beta\mathbf{v}, \alpha\mathbf{u} + \beta\mathbf{v}\rangle} = \frac{\alpha^2\langle \mathbf{u}, \mathbf{u}\rangle}{\alpha^2\langle \mathbf{u}, \mathbf{u}\rangle + \beta^2\langle \mathbf{v}, \mathbf{v}\rangle + 2\alpha\beta\langle \mathbf{u}, \mathbf{v}\rangle}.$$

The minimum of the above denominator function in the variable β is attained in

$$\beta = \frac{-\alpha\langle \mathbf{u}, \mathbf{v}\rangle}{\langle \mathbf{v}, \mathbf{v}\rangle}.$$

Thus,

$$\frac{\langle \alpha\mathbf{u}, \alpha\mathbf{u}\rangle}{\langle \alpha\mathbf{u}+\beta\mathbf{v}, \alpha\mathbf{u}+\beta\mathbf{v}\rangle} \leq \frac{\langle \mathbf{u},\mathbf{u}\rangle\langle \mathbf{v},\mathbf{v}\rangle}{\langle \mathbf{u},\mathbf{u}\rangle\langle \mathbf{v},\mathbf{v}\rangle - \langle \mathbf{u},\mathbf{v}\rangle^2} = = \left(1 - \frac{\langle \mathbf{u},\mathbf{v}\rangle^2}{\|\mathbf{u}\|^2\|\mathbf{v}\|^2}\right)^{-1} \leq \frac{4}{3}.$$

■

The following example illustrates that $2/\sqrt{3}$ is an optimal bound.

Example 4: Consider the lattice generated by the reduced basis

$$\{\mathbf{u} = (1,0), \mathbf{v} = (1/2, \sqrt{3}/2)\}.$$

We have:

$$2\mathbf{u} - \mathbf{v} = (3/2, -\sqrt{3}/2),$$

$$\|2\mathbf{u} - \mathbf{v}\| = \sqrt{3},$$

$$\| - \mathbf{v}\| = 1 < \|2\mathbf{u}\| = 2 = \frac{2}{\sqrt{3}}\sqrt{3}.$$

A weaker bound for arbitrary s-dimensional lattices is given in the paper [7].

Lemma 5: Let $\{\mathbf{u}, \mathbf{v}\} \subset \mathbb{R}^r$ be a reduced basis of a 2-rank lattice $\mathcal{L}$. Then we have:

$$\mathrm{vol}\,(\mathcal{L}) \leq \|\mathbf{u}\|\|\mathbf{v}\| \leq \frac{2}{\sqrt{3}}\,\mathrm{vol}\,(\mathcal{L}).$$

Proof: The first inequality is immediate. For the second, note that $\mathrm{vol}\,(\mathcal{L}) = \|\mathbf{u}\|\|\mathbf{v}\|\,|\sin(\widehat{\mathbf{u},\mathbf{v}})|$. By Lemma 3, we have $|\sin(\widehat{\mathbf{u},\mathbf{v}})| \geq \sqrt{3}/2$. ■

III. Proof of the main Result

In a similar way to the proof of Theorem 4 in [3], we are building a "bad" set for which we cannot guarantee the success of the algorithm. However, in this new proof the set does not consist of values for the *seed* u_0. In this case, the exceptional set consists of pairs including both the initial value u_0 and the first approximation error ε_0. We will prove that when the input instance does not match a pair in this "bad" set, the algorithm works correctly. Moreover, since the size of that set is in $O\left(\max\{\Delta^{15}p^{-4}, \Delta^{19/5}\}\right)$, if this quantity remains lower than $p\Delta$, we can probably obtain successful guessing. So, the threshold for the maximum error provided by this proof is $\Delta < p^{5/14}$. Let us start to describe the algorithm.

Let $\varepsilon_j = u_j - w_j$, $j = 0, 1$. From

$$u_1 \equiv u_0^2 + c \bmod p,$$

we obtain

$$2w_0\varepsilon_0 + \varepsilon_0^2 - \varepsilon_1 + w_0 + c - w_1 \equiv_p 0, \tag{4}$$

which can be looked on as a linear equation over a vector containing enough information to discover the goal u_0. In fact, the vector

$$\mathbf{e} = \left(\Delta\varepsilon_0, \varepsilon_0^2 - \varepsilon_1\right) = (\Delta e_1, e_2)$$

is a solution to the following linear system of congruences:

$$\begin{aligned} 2w_0x_1 + \Delta x_2 &\equiv \Delta(w_1 - c - w_0^2) \bmod p, \\ x_1 &\equiv 0 \bmod \Delta. \end{aligned} \tag{5}$$

Moreover, $\mathbf{e}$ is a relatively short vector. We have:

$$|e_1| \leq \Delta, \qquad |e_2| \leq 2\Delta^2;$$

thus

$$\|\mathbf{e}\| \leq \sqrt{5}\Delta^2. \tag{6}$$

Let $\mathcal{L}$ be the lattice consisting of integer solutions $\mathbf{x} = (x_1, x_2) \in \mathbb{Z}^2$ of the system of congruences:

$$\begin{aligned} 2w_0x_1 + \Delta x_2 &\equiv 0 \bmod p, \\ x_1 &\equiv 0 \bmod \Delta. \end{aligned} \tag{7}$$

It is easily checked that $\mathcal{L}$ is a two-dimensional lattice with volume $p\Delta$. A particular solution of the linear system (5) is $\mathbf{t} = (\Delta, w_1 - c - w_0^2 - 2w_0)$. Now, we can apply an algorithm solving the CVP for the shift vector $\mathbf{t}$ and the lattice $\mathcal{L}$ to obtain a vector $\mathbf{f} = (\Delta f_1, f_2)$ satisfying equations (5) and

$$\|\mathbf{f}\| \leq \sqrt{5}\Delta^2. \tag{8}$$

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. If not, it can be shown (as in [3]) that u_0 belongs to a subset $\mathcal{V} \subseteq \mathbb{F}_p$ of cardinality $\#\mathcal{V} = O(\Delta^3)$. Our approach here is more involved.

We compute in polynomial time (for instance, as explained in [13]) a reduced basis

$$\{\mathbf{g} = (\Delta g_1, g_2), \mathbf{h} = (\Delta h_1, h_2)\}$$

of the lattice $\mathcal{L}$, with $\|\mathbf{g}\| \leq \|\mathbf{h}\|$. Since $\mathbf{e} - \mathbf{f} \in \mathcal{L}$, there exist integers γ_1 and γ_2 satisfying:

$$\mathbf{e} - \mathbf{f} = \gamma_1\mathbf{g} + \gamma_2\mathbf{h}. \tag{9}$$

By (6) and (8) we have $\|\mathbf{e} - \mathbf{f}\| \leq 2\sqrt{5}\Delta^2$ and thanks to Lemma 3 we obtain the following bounds:

$$\begin{aligned} |\gamma_1| &\leq \max\left\{1, \left\lfloor \frac{4\sqrt{5}\Delta^2}{\sqrt{3}\|\mathbf{g}\|} \right\rfloor\right\} =: \Delta_1, \\ |\gamma_2| &\leq \max\left\{1, \left\lfloor \frac{4\sqrt{5}\Delta^2}{\sqrt{3}\|\mathbf{h}\|} \right\rfloor\right\} =: \Delta_2. \end{aligned} \tag{10}$$

So, the missing information is now γ_1 and γ_2. From (9) we derive two new equations:

$$\begin{aligned} \varepsilon_0 &= f_1 + \gamma_1 g_1 + \gamma_2 h_1, \\ \varepsilon_0^2 - \varepsilon_1 &= f_2 + \gamma_1 g_2 + \gamma_2 h_2. \end{aligned} \tag{11}$$

Eliminating ε_0 from the above equation, we obtain:

$$(f_1 + \gamma_1 g_1 + \gamma_2 h_1)^2 - \varepsilon_1 = f_2 + \gamma_1 g_2 + \gamma_2 h_2.$$

Operating, we reach an equation involving six variables related to γ_1, γ_2 and ε_1:

$$f_1^2 + 2f_1\gamma_1 g_1 + 2f_1\gamma_2 h_1 + g_1^2\gamma_1^2 + 2g_1h_1\gamma_1\gamma_2 + h_1^2\gamma_2^2 = \\ = f_2 + \gamma_1 g_2 + \gamma_2 h_2 + \varepsilon_1. \tag{12}$$

Therefore, the system of congruences (13)

$$\begin{aligned} (2f_1g_1 - g_2)\Delta_1 x_1 + (2f_1h_1 - h_2)\Delta_2 x_2 + g_1^2\Delta_1^2 x_3 + \\ +2g_1h_1\Delta_1\Delta_2 x_4 + h_1^2\Delta_2^2 x_5 - \Delta x_6 = (f_2 - f_1^2)\Delta_1^2\Delta_2^2\Delta, \\ x_1 \equiv 0 \bmod \Delta_1\Delta_2^2\Delta, \\ x_2 \equiv 0 \bmod \Delta_1^2\Delta_2\Delta, \\ x_3 \equiv 0 \bmod \Delta_2^2\Delta, \\ x_4 \equiv 0 \bmod \Delta_1\Delta_2\Delta, \\ x_5 \equiv 0 \bmod \Delta_1^2\Delta, \\ x_6 \equiv 0 \bmod \Delta_1^2\Delta_2^2 \end{aligned} \tag{13}$$

contains the vector

$$\mathbf{\Gamma} = (\Delta_1\Delta_2^2\Delta\gamma_1, \Delta_1^2\Delta_2\Delta\gamma_2, \Delta_2^2\Delta\gamma_1^2, \\ \Delta_1\Delta_2\Delta\gamma_1\gamma_2, \Delta_1^2\Delta\gamma_2^2, \Delta_1^2\Delta_2^2\varepsilon_1).$$

The bound (10) implies that

$$\|\mathbf{\Gamma}\| \leq \sqrt{6}\Delta_1^2\Delta_2^2\Delta. \tag{14}$$

Let $\mathcal{L}'$ be the lattice consisting of integer solutions $\mathbf{x} = (x_1, \ldots, x_6) \in \mathbb{Z}^6$ of the homogeneus system obtained from (13):

$$\begin{aligned} (2f_1g_1 - g_2)\Delta_1 x_1 + (2f_1h_1 - h_2)\Delta_2 x_2 + \\ g_1^2\Delta_1^2 x_3 + 2g_1h_1\Delta_1\Delta_2 x_4 + h_1^2\Delta_2^2 x_5 - \Delta x_6 = 0, \\ x_1 \equiv 0 \bmod \Delta_1\Delta_2^2\Delta, \\ x_2 \equiv 0 \bmod \Delta_1^2\Delta_2\Delta, \\ x_3 \equiv 0 \bmod \Delta_2^2\Delta, \\ x_4 \equiv 0 \bmod \Delta_1\Delta_2\Delta, \\ x_5 \equiv 0 \bmod \Delta_1^2\Delta, \\ x_6 \equiv 0 \bmod \Delta_1^2\Delta_2^2. \end{aligned} \tag{15}$$

Applying an algorithm solving the CVP for the shift vector $(0,0,0,0,0,(f_1^2 - f_2)\Delta_1^2\Delta_2^2)$ and the lattice $\mathcal{L}'$, we obtain a vector

$$\mathbf{F} = (\Delta_1\Delta_2^2\Delta F_1, \Delta_1^2\Delta_2\Delta F_2, \Delta_2^2\Delta F_3, \\ \Delta_1\Delta_2\Delta F_4, \Delta_1^2\Delta F_5, \Delta_1^2\Delta_2^2 F_6),$$

satisfying equations (13) and:

$$\|\mathbf{F}\| \leq \sqrt{6}\Delta_1^2\Delta_2^2\Delta. \tag{16}$$

Again, we note that we may compute $\mathbf{F}$ in polynomial time from the information we are given. We might hope that

$$\varepsilon_0 = f_1 + F_1 g_1 + F_2 h_1.$$

Let us bound the "bad" possibilities for which this identity may be wrong. The vector $\mathbf{d}$ defined by $\mathbf{F} - \mathbf{\Gamma}$ lies in $\mathcal{L}'$:

$$\mathbf{d} = (\Delta_1\Delta_2^2\Delta d_1, \Delta_1^2\Delta_2\Delta d_2, \Delta_2^2\Delta d_3, \\ \Delta_1\Delta_2\Delta d_4, \Delta_1^2\Delta d_5, \Delta_1^2\Delta_2^2 d_6).$$

The bounds (14) and (16) imply $\|\mathbf{d}\| \leq 2\sqrt{6}\Delta_1^2\Delta_2^2\Delta$ and

$$\begin{aligned} |F_1 - \gamma_1| = |d_1| \leq 2\sqrt{6}\Delta_1, \\ |F_2 - \gamma_2| = |d_2| \leq 2\sqrt{6}\Delta_2, \\ |F_3 - \gamma_1^2| = |d_3| \leq 2\sqrt{6}\Delta_1^2, \\ |F_4 - \gamma_1\gamma_2| = |d_4| \leq 2\sqrt{6}\Delta_1\Delta_2, \\ |F_5 - \gamma_2^2| = |d_5| \leq 2\sqrt{6}\Delta_2^2, \\ |F_6 - \varepsilon_1| = |d_6| \leq 2\sqrt{6}\Delta. \end{aligned} \tag{17}$$

Setting $\mathbf{q} = d_1\mathbf{g} + d_2\mathbf{h} = (\Delta q_1, q_2)$, we have that $\mathbf{q}$ lies in $\mathcal{L}$. Using equations (15), we obtain :

$$q_1 = d_1 g_1 + d_2 h_1, \quad \text{and}$$
$$q_2 = 2f_1 g_1 d_1 + 2f_1 h_1 d_2 + g_1^2 d_3 + 2g_1 h_1 d_4 + h_1^2 d_5 - d_6.$$

Hence

$$2q_1 w_0 + q_2 \equiv 0 \mod p. \tag{18}$$

After substitutions $w_0 = u_0 - \varepsilon_0 = u_0 - (f_1 + \gamma_1 g_1 + \gamma_2 h_1)$ in the above equation (18) we find

$$2q_1 u_0 \equiv E \bmod p, \tag{19}$$

where $E = 2q_1(f_1 + \gamma_1 g_1 + \gamma_2 h_1) - q_2$.

And substituting the expressions for q_1, q_2 in E and operating, we obtain:

$$E = g_1^2(2\gamma_1 d_1 - d_3) + 2g_1 h_1(-d_4 + \gamma_1 d_2 + \gamma_2 d_1) + \\ + h_1^2(-d_5 + 2\gamma_2 d_2) + d_6. \tag{20}$$

Since we are assuming that $\varepsilon_0 \neq f_1 + F_1 g_1 + F_2 h_1$, we get $q_1 = d_1 g_1 + d_2 h_1 \neq 0$. So, for each value q_1 and E there is one unique value u_0 satisfying congruence (19).

Let $\mathcal{U}$ be the set consisting of all pairs (u_0, ε_0) for which, setting $\{\mathbf{g} = (\Delta g_1, g_2), \mathbf{h} = (\Delta h_1, h_2)\}$ as the output of the algorithm used to obtain a reduced basis when given the

lattice (7) and $w_0 := u_0 - \varepsilon_0$, there exist nine integers:

$$\begin{array}{llll} d_1, & |d_1| \le 2\sqrt{6}\Delta_1, & d_6, & |d_6| \le 2\sqrt{6}\Delta, \\ d_2, & |d_2| \le 2\sqrt{6}\Delta_2, & f_1, & |f_1| \le \sqrt{5}\Delta, \\ \gamma_1, & |\gamma_1| \le \Delta_1, & & \\ \gamma_2, & |\gamma_2| \le \Delta_2, & & \\ d_3, & |d_3| \le 2\sqrt{6}\Delta_1^2, & & \\ d_4, & |d_4| \le 2\sqrt{6}\Delta_1\Delta_2, & & \\ d_5, & |d_5| \le 2\sqrt{6}\Delta_2^2, & & \end{array} \tag{21}$$

satisfying the following two equations:

$$2(d_1g_1 + d_2h_1)u_0 \equiv g_1^2(2\gamma_1 d_1 - d_3) + 2g_1h_1(-d_4 + \gamma_1 d_2 + \gamma_2 d_1) + h_1^2(-d_5 + 2\gamma_2 d_2) + d_6 \bmod p, \tag{22}$$

$$(2f_1g_1 - g_2)d_1 + (2f_1h_1 - h_2)d_2 + g_1^2 d_3 + 2g_1h_1 d_4 + h_1^2 d_5 - d_6 = 0. \tag{23}$$

It is clear that for any pair outside $\mathcal{U}$, the algorithm proposed must work properly. In order to bound the size of this set, we introduce another set $\mathcal{T}$, consisting of pairs $(u_0, \varepsilon_0) \in \mathbb{F}_p \times [-\Delta, \ldots, \Delta]$ for which there are integers q_1 and E:

$$|q_1| \le 84\Delta^{4/5}, \quad |E| \le 20\sqrt{6}\Delta^2$$

satisfying:

$$2u_0q_1 \equiv E, \ q_1 \neq 0 \bmod p.$$

Since the second component ε_0 in the pair is meaningless in this definition, we can state $\#\mathcal{T} = O(\Delta^{19/5})$. Let us now measure the difference set $\mathcal{U}\backslash\mathcal{T}$: if (u_0, ε_0) and (u_0', ε_0') are two elements in this set, by the definition of $\mathcal{U}$ there are integers $(d_1, d_2, \gamma_1, \gamma_2, d_3, d_4, d_5, d_6, f_1)$ corresponding to (u_0, ε_0), and $(d_1', d_2', \gamma_1', \gamma_2', d_3', d_4', d_5', d_6,' f_1')$ corresponding to (u_0', ε_0') satisfying (21), (22) and (23). Suppose $(d_1, d_2, \gamma_1, \gamma_2, d_3, d_4, d_5) = (d_1', d_2', \gamma_1', \gamma_2', d_3', d_4', d_5')$ and $u_0 - \varepsilon_0 = u_0' - \varepsilon_0'$. From equation (23), we find

$$2(f_1 - f_1')(d_1g_1 + d_2h_1) = d_6 - d_6'.$$

The bounds (21) imply that $d_1g_1 + d_2h_2 = \Omega\left(\Delta^{4/5}\right)$, and then, $|f_1 - f_1'|$ is bounded by $O(\Delta^{1/5})$. Now, using equations (7), we reach:

$$2u_0(f_1 - f_1') \equiv \varepsilon_0(f_1 - f_1') + (f_2 - f_2') \bmod p.$$

However, using once again the fact that the first pair selected is outside $\mathcal{T}$, it must be $f_1 - f_1' = 0$. So, the integers $u_0 - \varepsilon_0$, $d_1, d_2, \gamma_1, \gamma_2, d_3, d_4$ and d_5 determine a unique element of $\mathcal{U}\backslash\mathcal{T}$. Hence, fixed the integer $u_0 - \varepsilon_0$, the number of choices is bounded by $O\left((\Delta_1\Delta_2)^5\right)$. Using Lemma 5, $O\left((\Delta_1\Delta_2)^5\right) = O\left((\Delta^3 p^{-1})^5\right)$. Then, $\#\left(\mathcal{U}\backslash\mathcal{T}\right) = O(\Delta^{15}p^{-4})$.

Finally, $\#\mathcal{U} \le \#\left(\mathcal{U}\backslash\mathcal{T}\right) + \#\mathcal{T}$.

IV. Unknown shift

As we pointed out in the introduction, this paper is mainly devoted to the Pollard generator, when the shift parameter is supposed to be known. If this information is not previously given, the paper [3] requires three (instead of two) consecutive approximations to sequence elements to performing a lattice attack. Usually, it is easy to have access to lots of approximations; so, this is not a restrictive feature. However, the algorithm in [3] requires better-quality approximations: one can recover the seed u_0 when $\Delta < 1/4$.

One can develop a similar algorithm to the one presented in Section III, but it is not immediately clear how to bound the failure possibilities as we have done there. The admitted error, as observed in empirical tests, grows from $\delta = 0.25$ to $\delta \simeq 0.261$. So, the improving factor with this two-round technique is only about 4%. Moreover, in [3] a heuristic method that reached $\delta = 1/3$ as maximum tolerance was presented. The following table compares the proportion of unkown bits with the success percentage in the tests.

δ	0.26	0.2613	0.2616	0.2618	0.262
	100%	100%	48%	16%	0

V. Remarks and Open Questions

Obviously our result is nontrivial only for $\Delta = O(p^{5/14})$. Thus, increasing the size of the admissible values of Δ is very interesting. One of the main differences of our approach, when compared with previous papers on this matter, is that we use the Closest Vector Problem instead of the Shortest Vector Problem and that we apply it twice. We believe that if we repeat this procedure more times we can get a better bound, say $\Delta = O(p^{2/5})$. This question undoubtedly deserves further study.

We have implemented the algorithm of our result in **C++** language using the **NTL** library; see [23]. Testing the algorithm with this implementation confirms the threshold $\delta = 1/3$ for maximum fraction of bits hidden when we only perform on lattice reduction (as in [3]). However, the full algorithm, as explained above, successfully obtains its output in instances even more difficult than $\delta = 5/14 = 0.35714\ldots$ The following table shows these tests results for a 1000-bit prime generator, following the one-round algorithm in [3] and the two-round one presented here.

δ	0.3	0.3307	0.3333	0.3385	0.3615	0.3641
1r	100%	100%	50%	0	0	0
2r	100%	100%	100%	100%	98%	0

Tests with higher-size prime generators set the threshold empirically in $\delta \simeq 0.363$. However, we fail to give a rigorous proof for a bound better than $\delta = 5/14$.

When the shift c also remains unknown, we can extend the bound in [3] ($\delta = 0.25$) to $\delta \simeq 0.2613$. Anyway, this is clearly worse than the heuristic method in [3] ($\delta = 1/3$).

We also believe that our approach works for other nonlinear pseudorandom number generators. Partial results can be consulted in paper [7]. Finally, it is not clear for us how to attack generators when the modulus p is hidden.

References

[1] M. Ajtai, R. Kumar, and D. Sivakumar, "A sieve algorithm for the shortest lattice vector problem". *Proc. 33rd ACM Symp. on Theory of Comput. (STOC 2001)*, Association for Computing Machinery, 2001, 601–610.

[2] S. R. Blackburn, D. Gómez, J. Gutierrez, and I. E. Shparlisnki, "Predicting the inversive generator". *Cryptography and Coding*, LNCS **2898** (2003), 264–275.

[3] S. R. Blackburn, D. Gómez, J. Gutierrez, and I. E. Shparlinski, "Predicting nonlinear pseudorandom number generators". *Math. Computation*, **74** (2005), 1471–1494.

[4] S. R. Blackburn, D. Gómez, J. Gutierrez, and I. E. Shparlinski, "Reconstructing noisy polynomial evaluation in residue rings". Journal of Algorithms. S 0196-6774(04)00115-4/FLA AID: 1388. Electronically available.

[5] E. F. Brickell and A. M. Odlyzko, "Cryptanalysis: A survey of recent results". *Contemp. Cryptology*, IEEE Press, 1992, 501–540.

[6] D. Gómez, J. Gutierrez, and A. Ibeas, "Cryptanalysis of the Quadratic Generator". *Progress in Criptology - INDOCRYPT 2005*, LNCS **3797** (2005), Springer-Verlag, 118–129

[7] D. Gómez, J. Gutierrez, and A. Ibeas, "An algorithm for finding small solutions of multivariate equations over the integers". *Preprint*, University of Cantabria, 2005

[8] M. Grötschel, L. Lovász, and A. Schrijver, "Geometric algorithms and combinatorial optimization". Springer-Verlag, 1993.

[9] P. M. Gruber and C. G. Lekkerkerker, "Geometry of numbers". Second edition, North-Holland, 1987.

[10] A. Joux and J. Stern, "Lattice reduction: A toolbox for the cryptanalyst". *J. Cryptology*, **11** (1998), 161–185.

[11] R. Kannan, "Algorithmic geometry of numbers". *Annual Review of Comp. Sci.*, **2** (1987), 231–267.

[12] R. Kannan, "Minkowski's convex body theorem and integer programming". *Math. Oper. Res.*, **12** (1987), 415–440.

[13] M. Kaib and C.P. Schnorr: "The Generalized Gauss Reduction Algorithm". Journal of Algorithms, **21** (1996), n. 3, 565–578.

[14] J. C. Lagarias, "Pseudorandom number generators in cryptography and number theory". *Proc. Symp. in Appl. Math.*, AMS, **42** (1990), 115–143.

[15] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients". *Mathematische Annalen*, **261** (1982), 515–534.

[16] D. Micciancio and S. Goldwasser, "Complexity of lattice problems". Kluwer, 2002.

[17] P. Q. Nguyen and J. Stern, "Lattice reduction in cryptology: An update". *Proc. ANTS-IV*, LNCS **1838** (2000), Springer-Verlag, 85–112.

[18] P. Q. Nguyen and J. Stern, "The two faces of lattices in cryptology", *Cryptography and Lattices*, LNCS **2146** (2001), Springer-Verlag, 146–180.

[19] H. Niederreiter, "New developments in uniform pseudorandom number and vector generation". *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, Lecture Notes in Statistics **106** (1995), Springer-Verlag, 87–120.

[20] H. Niederreiter, "Design and analysis of nonlinear pseudorandom number generators". *Monte Carlo Simulation*, A.A. Balkema Publishers, 2001, 3–9.

[21] H. Niederreiter and I. E. Shparlinski, "Recent advances in the theory of nonlinear pseudorandom number generators". *Proc. Conf. on Monte Carlo and Quasi-Monte Carlo Methods, 2000*, Springer-Verlag, 2002, 86–102.

[22] H. Niederreiter and I. E. Shparlinski, "Dynamical systems generated by rational functions". *Applied Algebra, Algebraic Algorithms and Error Correcting Codes – AAECC-15*, LNCS **2643** (2003), Springer-Verlag, 6–17.

[23] V. Shoup, "Number theory **C++** library (NTL)", version 5.3.1, available at `http://www.shoup.net/ntl/`.

Domingo Gómez received the M.Sc. and Ph.D. degrees in mathematics from the University of Cantabria, Spain in 2002 and 2006 respectively. Currently, he is a postdoctoral student at University of Cantabria.
His research interests include applied algebra, computational number theory and its applications, in particular cryptography and pseudorandom number generation.

Jaime Gutierrez received the M.Sc. degree in 1982 from University of Valladolid, Spain, and the Ph.D. degree in mathematics in 1988 from University of Cantabria, Spain. Since 1992, he has been with the Department of Mathematics and Computing, University of Cantabria.
His main research interests include computer algebra, computational number theory with its applications to coding and cryptography.

Álvar Ibeas was born in San Sebastián, Spain on December 1981. He received the M.Sc. (Hons.) degree in mathematics from the University of Cantabria, Spain in 2004. In October 2004, he became a graduate student at the Department of Mathematics and Computing at the University of Cantabria, where he is currently working on his Ph.D. dissertation.
His research interests include applied algebra, computational number theory and its applications, in particular cryptography and pseudorandom number generation.