# Inferring Sequences Produced by Nonlinear Pseudorandom Number Generators Using Coppersmith's Methods

Aurélie Bauer[1], Damien Vergnaud[2⋆], and Jean-Christophe Zapalowicz[3⋆⋆]

[1] Agence Nationale de la Sécurité des Systèmes d'Information
51 Boulevard de la Tour-Maubourg - 75700 Paris 07 SP, France
`aurelie.bauer@ssi.gouv.fr`
[2] École normale supérieure – C.N.R.S. – I.N.R.I.A.
45, rue d'Ulm, F-75230 Paris CEDEX 05, France
[3] INRIA Rennes – Bretagne Atlantique
Campus de Beaulieu, 35042, Rennes, France
`jean-christophe.zapalowicz@inria.fr`

**Abstract.** *Number-theoretic* pseudorandom generators work by iterating an algebraic map $F$ (public or private) over a residue ring $\mathbb{Z}_N$ on a secret random initial seed value $v_0 \in \mathbb{Z}_N$ to compute values $v_{n+1} = F(v_n) \bmod N$ for $n \in \mathbb{N}$. They output some consecutive bits of the state value $v_n$ at each iteration and their efficiency and security are thus strongly related to the number of output bits. In 2005, Blackburn, Gomez-Perez, Gutierrez and Shparlinski proposed a deep analysis on the security of such generators. In this paper, we revisit the security of number-theoretic generators by proposing better attacks based on Coppersmith's techniques for finding small roots on polynomial equations. Using intricate constructions, we are able to significantly improve the security bounds obtained by Blackburn *et al.*.

**Keywords**: Nonlinear Pseudorandom number generators, Euclidean lattice, LLL algorithm, Coppersmith's techniques, Unravelled linearization

## 1 Introduction

This paper aims to present new cryptanalytic results on some nonlinear number-theoretic pseudorandom number generators. We show that several generators are insecure if sufficiently many bits are output at each clocking cycle. In particular, this provides an upper bound on the generators' security. The attacks used the well-known Coppersmith methods for finding small roots on polynomial equations and outperform previously known results [2–4, 10, 11].

**Prior work.** One of the most fundamental cryptographic primitives is the *pseudorandom bit generator*. It is a deterministic algorithm that expands a few truly

⋆ This author was supported in part by the European Commission through the ICT Program under contract ICT-2007-216676 ECRYPT II.

⋆⋆ Work done while at Agence Nationale de la Sécurité des Systèmes d'Information.

random bits to a longer sequence of bits that cannot be distinguished from uniformly random bits by a computationally bounded algorithm. It has numerous uses in cryptography, e.g. in signature schemes or public-key encryption schemes.

*Number-theoretic* pseudorandom generators work by iterating an algebraic map $F$ (public or private) over a residue ring $\mathbb{Z}_N$ on a secret random initial seed value $v_0 \in \mathbb{Z}_N$ to compute the intermediate state values $v_{i+1} = F(v_i) \bmod N$ for $i \in \mathbb{N}$ and outputting (some consecutive bits of) the state value $v_i$ at each iteration. The input $v_0$ of the generator (and possibly the description of $F$) is called the *seed* and the output is called the *pseudorandom sequence*. The case where $F$ is an affine function is known as the *linear congruential generator*. This generator is efficient and has good statistical properties. Unfortunately, it is cryptographically insecure: Boyar [7] proved that - with a sufficiently long run of the pseudorandom sequence - one can recover the seed in time polynomial in the bit-size of $N$ and Stern [17] proved that this is also the case even if one outputs only the most significant bits of each $v_i$ (see also [6, 15]).

It was suggested to use a non-linear algebraic map $F$ in order to avoid these attacks but several works [2–4, 10, 11] showed that not too many bits can be output at each stage. Blackburn, Gomez-Perez, Gutierrez and Shparlinski [3, 4] proved that some generators are polynomial time predictable if sufficiently many bits of some consecutive values of the pseudorandom sequence are revealed (even when $F$ is kept private).

Blackburn et al.'s results are based on a lattice basis reduction attack, using a certain linearization technique. A natural idea – already stated in [3] – is instead of using only linear relations in the attack, to use also relations that are derived by taking products of them. This technique was proposed by Coppersmith to find small roots on polynomial equations [8, 9]. In Coppersmith's method, a family of polynomials is first derived from the polynomial whose root is wanted. This family naturally gives a lattice basis and short vectors of this lattice possibly provide the wanted root. Blackburn *et al.* claimed that "this approach 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 goal in this paper is to investigate this issue.

**Our contributions.** We show that if a sufficient number of the most significant bits of several consecutive values $v_i$ of non-linear algebraic pseudorandom generator are given, one can recover the seed $v_0$ (even in the case where the coefficients of $F$ are unknown). We tackle these issues with Coppermith's lattice-based technique for calculating the small roots of multivariate polynomials modulo an integer. This method is heuristic, which is also the case of some arguments of Blackburn *et al.* showing that their basic results could be strengthened if the number of pseudorandom bits known to the attacker is increased. If $F$ is a polynomial of degree $d$ known to the attacker, Blackburn *et al.*'s result [4] proved that the generator can be predicted if one outputs a proportion $(d^2-1)/d^2$ of the most significant bits of two consecutive intermediate state values. We improve this result (*cf.* Section 3) by showing that this is also the case if one outputs a proportion as large as $d/(d+1)$ of the most significant bits of two consecutive intermediate state values (or $(d-1)/d$ for sufficiently many consecutive intermediate state values).

Blackburn *et al.* [2, 3] then focused on the well-known following number-theoretic pseudorandom generators (where $p$ is a prime, $a \in \mathbb{Z}_p^*$ and $b \in \mathbb{Z}_p$):

- The *Quadratic generator* corresponding to the map $F(x) = ax^2 + b \bmod p$
- The *Pollard generator*, a special case of the quadratic generator when $a = 1$
- The *Inversive generator* corresponding to the map $F(x) = ax^{-1} + b \bmod p$

Our generic results apply to these settings and improve the previous bounds. The theoretical data complexity (*i.e.* the minimum keystream length) of our attack is decreased compared to the attack from [2–4, 10, 11]. Therefore a secure use of these generators requires the output of much fewer bits at each iteration and the efficiency of the schemes is thus degraded.

The table below shows a comparison between our results and what is known in the literature. It gives the proportion of most significant bits output from each consecutive state values necessary to break the generator in (heuristic) polynomial time. The *basic proportion* corresponds to the case where the adversary knows bits coming from the minimum number of intermediate states leading to a feasible attack; while the *asymptotic proportion* corresponds to the case when the bits known by the adversary come from an infinite number of values.

| | | **Basic proportion** | | **Asymptotic proportion** | |
|---|---|---|---|---|---|
| | | **Prior result** | **Our result** | **Prior result** | **Our result** |
| **Quadratic** | $a$,$b$ known | **3/4** | **2/3** | **2/3** | **1/2** |
| **generator** | $a$,$b$ unknown | **18/19** | **11/12** | **11/12** | **2/3** |
| **Pollard** | $b$ known | **9/14** | **3/5** | **9/14** | **1/2** |
| **generator** | $b$ unknown | **3/4** | **5/7** | **2/3** | **3/5** |
| **Inversive** | $a$,$b$ known | **3/4** | **2/3** | **2/3** | **1/2** |
| **generator** | $a$,$b$ unknown | **14/15** | **11/12** | **11/12** | **2/3** |

The results on the quadratic generator (and the inversive generator) are described in Section 3.3 (and Section 3.4) and are direct applications of our general results. Those on the Pollard generator relies on the *unravelled linearization* technique introduced by Hermann and May in 2009 [12] and are described in Section 4.

## 2 Preliminaries

### 2.1 Lattices

**Definition.** If $(b_1, \ldots, b_d)$ are $d$ linearly independent vectors over $\mathbb{Z}^n$, then the lattice $\mathcal{L} = \langle b_1, \ldots, b_d \rangle$ generated by these vectors is defined as the set of all integer linear combination of the $b_i$'s. The set $B = \{b_1, \ldots, b_d\}$ is called a *basis* of $\mathcal{L}$ and $d$ is the *dimension* of $\mathcal{L}$. We restrict ourselves to *full-rank* lattices corresponding to the particular case $d = n$. The quantity $|\det(B)|$ is called the *determinant* of the lattice $\mathcal{L}$.

**LLL-reduced bases.** In 1982, Lenstra, Lenstra and Lovász [16] defined *LLL-reduced* bases of lattices and presented a deterministic polynomial-time algorithm, called *LLL* to compute such a basis. If $(b_1, \ldots, b_n)$ is an LLL-reduced basis of $\mathcal{L}$, the first vector $b_1$ is close to be the shortest non-zero vector of the lattice. Moreover, if $(b_1^\star, \ldots, b_n^\star)$ are the corresponding vectors coming from Gram-Schmidt's orthogonalization, then:

$$\|b_n^\star\|_2 \geq 2^{-(n-1)/4} (\det \mathcal{L})^{1/n} \tag{1}$$

## 2.2 Coppersmith's techniques

In 1996, Coppersmith introduced lattice-based techniques [8, 9] for finding small roots on univariate and bivariate polynomial equations. As these techniques had a wide range of cryptanalytic applications, some reformulations and generalizations to more variables have been proposed [1, 5, 13, 14].

All these methods have allowed to attack many instances of public-key cryptosystems (*e.g.* [12, 15]). In the following, we give more details explaining how such techniques work in practice for the multivariate modular case.

**Definition of the problem.** Let $f(y_1, \ldots, y_n)$ be an irreducible multivariate polynomial defined over $\mathbb{Z}$, having a root $(x_1, \ldots, x_n)$ modulo a known integer $N$ such that $|x_1| < X_1, \ldots, |x_n| < X_n$. The question is to determine the bounds $X_i$ allowing to recover the desired root in polynomial time.

**Collection of polynomials.** One has to generate a collection of polynomials $f_1, \ldots, f_r$ having $(x_1, \ldots, x_n)$ as a modular root. Usually, we consider multiples and powers of the polynomial $f$, namely $f_\ell = y_1^{\alpha_1^{(\ell)}} \ldots y_n^{\alpha_n^{(\ell)}} f^{k_\ell}$, for $\ell$ in $\{1, \ldots, r\}$. By definition, such polynomials satisfy the relation $f_\ell(x_1, \ldots, x_n) \equiv 0 \mod N^{k_\ell}$, *i.e.* there exists an integer $c_\ell$ such that $f_\ell(x_1, \ldots, x_n) = c_\ell N^{k_\ell}$. From now, let us denote as $M$ the set of monomials appearing in the collection $\{f_1, \ldots, f_r\}$. We then construct a matrix $\mathcal{M}$ by extracting the polynomial coefficients as follows:

$$\mathcal{M} = \left(\begin{array}{cccc|ccc} & & & & f_1 & \ldots & f_r \\ & & & & \downarrow & \downarrow & \downarrow \\ 1 & & & & & & \\ & X_1^{-1} & & & & & \\ & & \ddots & & & & \\ & & & X_1^{-a_1} \ldots X_n^{-a_n} & & & \\ \hline & & & & N^{k_1} & & \\ & & 0 & & & \ddots & \\ & & & & & & N^{k_r} \end{array}\right) \begin{array}{c} \\ \\ 1 \\ y_1 \\ \vdots \\ y_1^{a_1} \ldots y_n^{a_n} \\ \\ \\ \\ \end{array}$$

Every row of the upper part is related to one monomial of the set $M$. The left-hand side contains the bounds corresponding to these monomials (*e.g.* the coefficient $X_1^{-1} X_2^{-2}$ is put in the row related to the monomial $y_1 y_2^2$). Each column of the right-hand side contains a vector coming from the initial collection $\{f_1, \ldots, f_r\}$. We define as $\mathcal{L}$ the lattice generated by $\mathcal{M}$'s rows and we have:

$$|\det(\mathcal{L})| = \frac{N^{k_1 + \cdots + k_r}}{\prod_{(y_1^{a_1} \ldots y_n^{a_n} \in M)} X_1^{a_1} \ldots X_n^{a_n}}.$$

**A short vector in the lattice $\mathcal{L}$.** Let us consider the vectors $r_0$ and $s_0$ defined by $r_0 = (1, x_1, \dots, x_1^{a_1} \dots x_n^{a_n}, -c_1, \dots, -c_r)$ and $s_0 = \mathcal{M} \cdot v_0 \in \mathcal{L}$, such that

$$s_0 = \left(1, (x_1/X_1), \dots, (x_1/X_1)^{a_1} \dots (x_n/X_n)^{a_n}, 0, \dots, 0\right).$$

One has $\|s_0\|_2 \leq \sqrt{\#M}$ and the knowledge of $s_0$ is sufficient to compute the root of $f$. Since in practice, we will not always recover $s_0$, the method consists in looking for a vector which is orthogonal to it. We compute an LLL-reduced basis $B = (b_1, \dots, b_t)$ of (a sublattice of) $\mathcal{L}$ and a Gram-Schmidt's orthogonalization on $B$. As $s_0$ belongs to $\mathcal{L}$, it can be expressed as a linear combination of the $b_i^\star$'s and if its norm is smaller than those of $b_t^\star$, then the dot product $\langle s_0, b_t^\star \rangle = 0$.

Extracting the coefficients in $b_t^\star$ leads to a polynomial $p_1$ defined over $M$ such that $p_1(x_1, \dots, x_n) = 0$ and iterating the process with $b_{t-1}^\star, \dots, b_{t-n+1}^\star$, one gets a multivariate polynomial system $\{p_1(x_1, \dots, x_n) = 0, \dots, p_n(x_1, \dots, x_n) = 0\}$. Under the (heuristic) assumption that these polynomials are algebraically independent, the system can be solved in polynomial time.

**Conditions on the bounds $X_i$'s.** Since $s_0$ is small and we have an upper bound on $\|b_t^\star\|_2$, (*cf.* (1)), the condition $\sqrt{\#M} < 2^{-(t-1)/4}(\det(\mathcal{L}))^{1/t}$ implies $\langle s_0, b_t^\star \rangle = 0$. Removing parameters that do not influence the asymptotic result, this relation can be simplified to $|\det(\mathcal{L})| > 1$, leading to the following final condition:

$$\prod_{(y_1^{a_1} \dots y_n^{a_n} \in M)} X_1^{a_1} \dots X_n^{a_n} < N^{k_1 + \cdots + k_r} \tag{2}$$

The most complex step of the method is the choice of the collection of polynomials, what could be a difficult task when working with multiple polynomials.

## 3 Attacking a non-linear generator

For $N$ an integer of size $\pi$, we denote by $\mathbb{Z}_N$ the residue ring of $N$ elements. A pseudorandom non-linear generator can be defined by the following recurrence sequence:

$$v_{i+1} = F(v_i) \mod N \tag{3}$$

where $F(X) = \sum_{j=0}^{d} c_j X^j$ is a polynomial of degree $d$ in $\mathbb{Z}_N[X]$ and $v_0$ is the secret seed. We assume that this generator outputs the $k$ most significant bits of $v_i$ at each iteration (with $k \in \{1, \dots, \pi\}$), i.e. if $v_i = 2^{\pi-k} w_i + x_i$, $w_i$ is output by the generator and $x_i < 2^{\pi-k} = N^\delta$ stays unknown. We want to recover $x_i < N^\delta$ for some $i \in \mathbb{N}$ from consecutive values of the pseudorandom sequence (with $\delta$ as large as possible) knowing $F$ or not.

### 3.1 Case $F$ known

Any non-linear pseudorandom generator defined by a known iteration function $F$ can be broken when sufficiently many bits are output at each iteration. In the following, we determine that amount of output bits when two (Theorem 1) then more (Theorem 2) consecutive outputs are known to the attacker.

**Theorem 1 (Two consecutive outputs).** *Let $\mathcal{G}$ be a non-linear pseudorandom generator defined by a known iteration function $F(X)$ of degree $d$. If an adversary has access to two consecutive outputs of $\mathcal{G}$ then it will be able to predict the entire sequence that follows ; under the condition that at least $\frac{d}{d+1}\pi$ most significant bits are output at each iteration, that is:*

$$\delta < \frac{1}{d+1}$$

*Proof.* Suppose the adversary is given two approximations $w_0$ and $w_1$ of two consecutive values $v_0$ and $v_1$ that satisfy (3). By denoting $v_0$ as $2^{\pi-k}w_0+x_0$ and $v_1 = 2^{\pi-k}w_1 + x_1$, we obtain:

$$2^{\pi-k}w_1 + x_1 - \sum_{j=0}^{d} c_j(2^{\pi-k}w_0 + x_0)^j = 0 \mod N$$

Let $f(y_0, y_1)$ be the polynomial $y_1+a_0+a_1y_0+\cdots+a_dy_0^d$ defined by this equation, where the values $a_i$, that explicitly depend on $w_0, w_1$ and the coefficients $c_i$, are known to the adversary. The goal is to compute efficiently the (small) modular root $(x_0, x_1)$ of $f(y_0, y_1)$. To do so, let us consider the following collection of polynomials:

$$\{y_0^j f^i(y_0, y_1) \quad | \quad di + j \leq dm \quad \wedge \quad i > 0\}$$

where $m \geq 1$ is a fixed integer. Knowing the shape of $f$, the list of monomials appearing within this collection can be described as:

$$\{y_1^i y_0^j \quad | \quad di + j \leq dm\}$$

Using Coppersmith's method, the right-hand side (*resp.* the left-hand side) of (2) is then equal to:

$$\prod_{i=1}^{m}\prod_{j=0}^{d(m-i)} N^i = N^{\frac{1}{6}m(m+1)(dm-d+3)} \qquad \left( \textit{resp.} \prod_{i=0}^{m}\prod_{j=0}^{d(m-i)} N^{i\delta}N^{j\delta} \right).$$

Thus, the algorithm (heuristically) outputs the root of $f$ in polynomial time as soon as:

$$\delta < \frac{\frac{1}{6}m(m+1)(dm-d+3)}{\frac{1}{12}m(m+1)(2d^2m+2dm+6+d^2+d)} \xrightarrow[m\to+\infty]{} \frac{1}{d+1} \tag{4}$$

□

This bound is better than those previously obtained by Blackburn *et al.* [3]. Indeed, their result was approximately $\delta < 1/d^2$ when two consecutive outputs are known to the attacker.

**Theorem 2 (More consecutive outputs).** *Let $\mathcal{G}$ be a non-linear pseudorandom generator defined by a known iteration function $F(X)$ of degree $d$. If an* *adversary has access to $n+2$ (with $n \geq 1$) consecutive outputs of $\mathcal{G}$ then it will be able to predict the entire sequence that follows ; under the condition that at least $\frac{d^{n+2}-d^{n+1}}{d^{n+2}-1}\pi$ most significant bits are output at each iteration, that is:*

$$\delta < \frac{d^{n+1}-1}{d^{n+2}-1}$$

*Proof.* Let us assume that the attacker knows $n+2$ consecutive outputs of the generator $w_0, \dots, w_{n+1}$. Writing $v_i$ as $2^{\pi-k}w_i + x_i$ (for $i \in \{0, \dots, n+1\}$), we want to recover the solution $(x_0, \dots, x_{n+1})$ of the multivariate polynomial system:

$$\begin{cases} f_0(y_0, y_1) &= y_1 + a_{00} + a_{01}y_0 + \dots + a_{0d}y_0^d \mod N \\ &\vdots \\ f_n(y_n, y_{n+1}) &= y_{n+1} + a_{n0} + a_{n1}y_n + \dots + a_{nd}y_n^d \mod N \end{cases}$$

where each polynomial $f_i$ is constructed in the same way as for the "two consecutive outputs" case. From now, we use the following collection of polynomials:

$$\left\{ y_0^j f_0^{i_0} \dots f_n^{i_n} \quad | \quad d(i_0 + di_1 + \dots + d^n i_n) + j \leq dm \quad \wedge \quad i_0 + \dots + i_n > 0 \right\}$$

where $m \geq 1$ is a fixed integer. As it seems to be a difficult task to describe the set of monomials appearing in that collection for the general case, we first focus on what happens with two polynomials $f_0$ and $f_1$. In that case, the set can be described by the powers of these polynomials, that is

$$\left\{ (y_0^j y_1^i) \cdot (y_1^k y_2^l) \quad | \quad di + j \leq dm \quad \wedge \quad dl + k \leq dm - di - j \right\}$$

Another way of expressing this set is $\left\{ y_0^j y_1^i y_2^l \quad | \quad di + j + dl \leq dm \right\}$. From that point, by induction on $n$, we can show that the monomials appearing in the collection can be described as:

$$\left\{ y_0^j y_1^{i_0} \dots y_{n+1}^{i_n} \quad | \quad d(i_0 + di_1 + \dots + d^n i_n) + j \leq dm \right\}$$

The right-hand side and the left-hand side of (2) is then equal to $N^{A(m,n)}$ and $N^{B(m,n)}$ respectively, where:

$$A(m,n) = \sum_{i_0=0}^{m} \sum_{i_1=0}^{\lfloor \frac{m-i_0}{d} \rfloor} \dots \sum_{j=0}^{d(m-\sum_{p=0}^{n} d^p i_p)} i_0 + \dots + i_n$$

$$B(m,n) = \sum_{i_0=0}^{m} \sum_{i_1=0}^{\lfloor \frac{m-i_0}{d} \rfloor} \dots \sum_{j=0}^{d(m-\sum_{p=0}^{n} d^p i_p)} i_0 + \dots + i_n + j$$

Our goal is to obtain an asymptotic expression of the multiples sums $A(m,n)$ and $B(m,n)$ which depends on the number of outputs $n$, when $m$ goes to $+\infty$. It is quite clear that the floor function appearing in the upper bound of the sums can be omitted and we will use several times a trick from [12] which consists in letting indices of a sum run over a larger range in order to obtain a symmetric formula that is easier to evaluate. Basically, it relies on the following observation which holds for any function $f$:

$$\sum_{i=0}^{N} f(i) = \frac{1}{d} \sum_{i=0}^{dN} f(\lfloor \frac{i}{d} \rfloor).$$

Applying this trick $n$ times on $A(m,n)$, one obtains:

$$\begin{aligned} A(m,n) &\simeq \frac{1}{d} \cdots \frac{1}{d^n} \sum_{i_0=0}^{m} \sum_{i_1=0}^{m-i_0} \cdots \sum_{j=0}^{d(m-\sum_{p=0}^{n} i_p)} i_0 + \frac{1}{d} i_1 + \cdots + \frac{1}{d^n} i_n \\ &\simeq d \cdot \frac{1}{d} \cdots \frac{1}{d^n} \sum_{i_0=0}^{m} \sum_{i_1=0}^{m-i_0} \cdots \sum_{j=0}^{m-\sum_{p=0}^{n} i_p} i_0 + \frac{1}{d} i_1 + \cdots + \frac{1}{d^n} i_n \end{aligned}$$

and similarly

$$B(m,n) = d \cdot \frac{1}{d} \cdots \frac{1}{d^n} \sum_{i_0=0}^{m} \sum_{i_1=0}^{m-i_0} \cdots \sum_{j=0}^{m-\sum_{p=0}^{n} i_p} i_0 + \frac{1}{d} i_1 + \cdots + \frac{1}{d^n} i_n + dj.$$

We get for $A(m,n)$ and $B(m,n)$:

$$A(m,n) \simeq \frac{1}{d^2} \cdots \frac{1}{d^n} \left( \frac{d^{n+1}-1}{d^n(d-1)} \right) p_1 \text{ and } B(m,n) \simeq \frac{1}{d^2} \cdots \frac{1}{d^n} \left( \frac{d^{n+2}-1}{d^n(d-1)} \right) p_1$$

where

$$p_1 = \sum_{i_0=0}^{m} \sum_{i_1=0}^{m-i_0} \cdots \sum_{j=0}^{m-\sum_{p=0}^{n} i_p} i_0.$$

We obtain in consequence the following bound:

$$\delta < \frac{A(m,n)}{B(m,n)} \simeq \frac{d^{n+1}-1}{d^{n+2}-1}$$

□

When the number of consecutive values known by the adversary tends to infinity, this condition becomes $\delta < 1/d$. Knowing that $d$ is the degree of the iteration function, this result seems to be the optimal one when using Coppersmith's technique.

## 3.2 Case $F$ unknown

We show that a non-linear pseudorandom generator defined by an unknown iteration function $F$ can also be broken. In order to apply Coppersmith's technique, one needs to construct a polynomial $P$ (from the unknown iteration function $F$) with a root encoding the secret seed. We will see in the forthcoming sections that one could use elimination techniques to find such a $P$. Let us denote $D$ the degree of $P$ (depending on $d = \deg F$ and on the elimination technique used) and we consider a monomial order such that the leading coefficient[1] of $P$ is equal to 1 modulo $N$. Since there are $d+1$ unknown coefficients in $F$, one requires $d+2$ consecutive equations of the form $v_{i+1} = F(v_i) \bmod N$, and thus $d+3$ consecutive outputs of the generator.

**Theorem 3** **($d+3$ consecutive outputs).** *Let $\mathcal{G}$ be a non-linear pseudorandom generator defined by an unknown iteration function $F(X)$ of degree $d$. We consider an adversary that has access to $d+3$ consecutive outputs of $\mathcal{G}$ and can compute a polynomial $P$ of degree $D$ and a monomial order as above.*

*It will be able to predict the entire sequence that follows ; under the condition that at least $\frac{D^2(d+3)-1}{D(d+3)}\pi$ most significant bits are output at each iteration, that is $\delta < \frac{1}{D^2(d+3)}$. Moreover, if one assumes that the degree of the leading monomial of $P$ is equal to $D$, then this bound can be improved to:*

$$\delta < \frac{1}{D(d+3)}.$$

*Proof.* Let us assume that the adversary knows $w_0, \dots, w_{d+2}$. By manipulating the system $\big(v_{i+1} = F(v_i) \mod N, i \in \{0, \dots, d+1\}\big)$ one obtains a polynomial $P$ satisfying $P(x_0, \dots, x_{d+2}) = 0 \mod N$. Since the shape of $P$ and its degree $D$ both depend on the technique used to manipulate the initial system, describing the monomials appearing in $P$ and therefore in $P^m$ is an impossible task. Consequently, the only way to perform Coppersmith's method is to choose a simpler but larger set of monomials which necessarily contains those of $P^m$:

$$\Big\{ y_0^{j_0} \dots y_{d+2}^{j_{d+2}} \quad | \quad j_0 + j_1 + \dots + j_{d+2} \le Dm \Big\}$$

The leading monomial of $P$, $LM(P)$, can be described as $y_0^{\alpha_0} \dots y_{d+2}^{\alpha_{d+2}}$ where at least one of the $\alpha_i$ is non negative. Without loss of generality, we can assume for now that $\alpha_0 > 0$. In that case, one can apply Coppersmith's method on the following collection of polynomials:

$$\Big\{ y_1^{j_1} \dots y_{d+2}^{j_{d+2}} P^i \quad | \quad Di + j_1 + \dots + j_{d+2} \le Dm \ \wedge \ 1 \le i \le m \Big\}$$

As $y_0$ only comes from the powers of $P$, the prohibition of the multiplication by $y_0$ ensures that the collection of polynomials will be linearly independent. The right-hand side (*resp.* the left-hand side) of (2) is then equal to $N$ to the power:

$$\sum_{1 \le i \le m} \quad \sum_{j_1 + \dots + j_{d+2} \le Dm - Di} i \qquad \left( \textit{resp.} \sum_{j_0 + \dots + j_{d+2} \le Dm} \delta(j_0 + \dots + j_{d+2}) \right).$$

[1] In the general case, this condition is almost always satisfied and this is obviously true when $N$ is prime

We can show that this formula leads to the following condition:

$$\delta < \frac{1}{D^2(d+3)}$$

In fact, this result can be improved if one assumes that the degree of $LM(P)$ is equal to $D$. Indeed, this monomial can be described as $y_0^{\alpha_0} \dots y_{d+2}^{\alpha_{d+2}}$ with $\sum_{i=0}^{d+2} \alpha_i = D$. In order to keep the linear independency between the polynomials, one should only consider polynomials of the form $Mon \times P^i$ such that $Mon \neq LM(P)$. This leads to the following collection:

$$\left\{ y_0^{j_0} y_1^{j_1} \dots y_{d+2}^{j_{d+2}} P^i \;\middle|\; \begin{array}{ll} & Di + j_0 + j_1 + \cdots + j_{d+2} \leq Dm \\ \wedge & 1 \leq i \leq m \\ \wedge & (j_0 < \alpha_0) \cup \cdots \cup (j_{d+2} < \alpha_{d+2}) \end{array} \right\}$$

Using the same kind of tricks as in the proof of Theorem 2, the resulting asymptotic bound becomes:

$$\delta < \alpha_0 \frac{1}{D^2(d+3)} + \cdots + \alpha_{d+2} \frac{1}{D^2(d+3)} = \frac{1}{D(d+3)}$$

□

**More consecutive outputs.** We want to generalize the previous attack when the adversary is given access to more consecutive outputs. Let us assume, for instance, that it has access to $d+n+2$ consecutive values $w_0, \dots, w_{d+1+n}$ ; its goal is then to compute the (small) solution $(x_0, \dots, x_{n+d+1})$ of the multivariate polynomial system $(P_1(y_0, \dots, y_{d+2}), \dots, P_n(y_{n-1}, \dots, y_{n+d+1}))$ where the polynomials $P_i$ of degree $D$, are defined as in the previous section. As before, finding a general description of the monomials appearing in these polynomials is a challenging task. Thus we consider a larger set of monomials, easier to describe:

$$\left\{ y_0^{j_0} \dots y_{d+1+n}^{j_{d+1+n}} \quad | \quad j_0 + j_1 + \cdots + j_{d+1+n} \leq Dm \right\}$$

Let us express the leading monomial of $P_1$ as $y_0^{\alpha_0} \dots y_{d+2}^{\alpha_{d+2}}$ with at least one of the $\alpha_i \geq 1$, the leading monomial of $P_2$ as $y_1^{\alpha_0} \dots y_{d+3}^{\alpha_{d+2}}$ and those of $P_n$ as $y_{n-1}^{\alpha_0} \dots y_{n+d+1}^{\alpha_{d+2}}$, using a monomial order such as *lex* or *hlex* with $y_0 < \cdots < y_{d+1+n}$. Without loss of generality, we can assume that $\alpha_0 > 0$. From that, one can apply Coppersmith's method on the following collection of polynomials:

$$\left\{ y_n^{j_1} \dots y_{n+d+1}^{j_{d+2}} P_1^{i_1} \dots P_n^{i_n} \;\middle|\; \begin{array}{ll} & D(i_1 + \cdots + i_n) + j_1 + \cdots + j_{d+2} \leq Dm \\ \wedge & 1 \leq i_1 + \cdots + i_n \leq m \end{array} \right\}$$

The prohibition of the multiplication by $y_0, \dots, y_{n-1}$ ensures that all the polynomials of the collection are linearly independent. Thus, the right-hand side (*resp.* the left-hand side) of (2) is equal to $N$ to the power:

$$\sum_{\substack{1 \leq i_1 + \cdots + i_n \leq m \\ j_1 + \cdots + j_{d+2} \leq Dm - D(i_1 + \cdots + i_n)}} i_1 + \cdots + i_n \quad \left( resp. \sum_{j_0 + \cdots + j_{n+d+1} \leq Dm} \delta(j_0 + \cdots + j_{n+d+1}) \right).$$

We can show that the resulting asymptotic bound is $\delta < \dfrac{n}{D^{n+1}(n+d+2)}$ (details can be found in the full version).

*Remark 1.* This bound is not interesting as its value decreases when the adversary is given access to more outputs. However, we are convinced that it can significantly be improved. Indeed, using the same kind of techniques as in the previous case, we might be able to gain a factor $D$ for each involved polynomial and get:

$$\delta < \frac{n}{D(n+d+2)} \xrightarrow[m\to+\infty]{} \delta < \frac{1}{D}$$

In practice we notice that this conjecture seems to be true, see for instance the analysis of the quadratic generator in Section 3.3.

## 3.3 Application: Attacking the quadratic generator

For $p$ a prime of size $\pi$, the notation $\mathbb{Z}_p$ refers to the field of $p$ elements. The quadratic generator is defined by the following recurrence sequence:

$$v_i = av_{i-1}^2 + b \mod p \tag{5}$$

In that particular case, the iteration function $F(x)$ is defined as $F(x) = ax^2 + b$ where $a \in \mathbb{Z}_p^*$ and $b \in \mathbb{Z}_p$ are constant values. Exactly as before, we denote the secret seed as $v_0 \in \mathbb{Z}_p$ and we assume that the generator outputs the $k$ most significant bits of $v_i$ at each iteration (with $k \in \{1, \ldots, \pi\}$). In other words, each value $v_i$ can be written as $2^{\pi-k}w_i + x_i$ where $w_i$ is output by the generator and $x_i < 2^{\pi-k} = p^\delta$ stays unknown. Our goal consists in recovering the value $x_i < p^\delta$ for some $i \in \mathbb{N}$ by using some consecutive values output by the pseudorandom sequence (with $\delta$ as large as possible).

**Case *F* known.** If the adversary is given access to two consecutive outputs of the generator, then it can break the scheme under the condition that sufficiently many bits are output by the generator at each iteration. More precisely, for a fixed value $m$ (that will define the size of the corresponding lattice), the bound on $\delta$ should respect the following condition, directly coming from Equation (4) in Theorem 1:

$$\delta < \frac{1}{6} \cdot \frac{2m+1}{m+1}$$

In particular, taking $m = 1$ leads to the bound $\delta < 1/4$ previously reached by Blackburn *et al.* [3]. This bound can be improved to $\delta < 1/3$ when the quantity $m$ goes to infinity. This value is exactly the same as those previously obtained by Blackburn *et al.* [3] when the authors assume that the adversary is given access to an infinite number of outputs, whereas it only requires here two outputs of the generator. Finally, when increasing the number of known outputs to infinity, the condition becomes $\delta < 1/2$ (see Theorem 2).

**Case *F* unknown.** Knowing that the coefficients $a$ and $b$ appearing in the iteration function $F(x) = ax^2 + b$ are unknown to the attacker, the first step consists in expressing the relations between the outputs of the generator exclusively in terms of known quantities. More precisely, by using four consecutive outputs, the adversary is able to eliminate the quantities $a$ and $b$ by considering the following polynomial $P$ of degree 3:

$$\begin{aligned}P = c + c_0y_0 + c_1y_1 + c_2y_2 + c_3y_3 + d_0(y_0^2 - y_1^2) + d_1(y_2y_0 - y_0y_3) + d_2(y_1^2 - 3y_2^2)\\ +2d_2y_1y_2 + d_3(y_2^2 - 3y_1^2 + 2y_1y_3) + e(y_2^2y_1 - y_1^3 + y_1^2y_3 - y_2^3 - y_0^2y_3 + y_0^2y_2) \bmod p\end{aligned}$$

As each coefficient in this polynomial is inversible modulo $p$, one can consider that $LM(P)) = 1$. Thus, applying Theorem 3, one reaches the bound $\delta < 1/15$, knowing that the degree $d$ of the iteration function $F$ is equal to 2 and those of the polynomial $P$ is 3. In fact, this bound can be improved as the coefficient related to $x$ in the iteration function, is equal to zero. Indeed, the denominator in the formula given by Theorem 3 can in fact be expressed as $D.\ell(n)$ where $\ell(n)$ is the number of required outputs. In this particular case, as $\ell(n)$ is equal to four, the bound thus becomes $\delta < 1/12$. In the same scenario, Blackburn *et al.* [3] reached the value $\delta < 1/19$.

We assume that the adversary is given access to more consecutive outputs and generalize the previous construction using the fact that the iteration function $F$ contains one zero coefficient. In that case, if the set of monomials stays easy to formulate, namely $\{y_0^{j_0} \dots y_{n+2}^{j_{n+2}} \mid j_0 + \dots + j_{n+2} \le 3m\}$, this is not the case for the collection of polynomials which becomes:

$$\left\{ y_0^{j_0} y_1^{j_1} y_2^{a_2} \dots y_{n+1}^{a_{n+1}} y_{n+2}^{j_{n+2}} P_1^{i_1} \dots P_n^{i_n} \;\middle|\; \begin{array}{l} 0 < i_1 + \dots + i_n \le m \\ 0 \le a_\ell \le \min(2, 3m - 3\sum_{t=1}^{n} i_p - \sum_{t=2}^{\ell-1} a_p) \\ (\text{for } \ell \in \{2, \dots, n+1\}) \\ j_0 + j_1 + j_{n+2} \le 3m - 3(i_1 + \dots + i_n) \\ \qquad\qquad\qquad - (a_2 + \dots + a_{n+1}) \end{array} \right\}$$

The estimation of the "weight" of these two sets allows to reach the asymptotic bound $\delta < 1/3$ when both $m$ (related to the dimension of the involved lattice) and $n$ go to infinity (*cf.* the full version of the paper). This value seems to confirm the conjecture $\delta < 1/D$ discussed in Remark 1. Moreover, it significantly improves the bound $\delta < 1/12$ previously obtained by Blackburn *et al.* in [3] and it provides interesting asymptotic bounds for small values of $n$ (when $m$ goes to infinity):

| Number of outputs | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|
| Asymptotic bound | 1/12 | 2/15 | 1/6 | 4/21 | 5/24 | 2/9 | 7/30 | 8/33 | 1/4 |

## 3.4 The Inversive Generator

The inversive generator is defined by the recurrence sequence $v_i = av_{i-1}^{-1} + b \bmod p$ where $p$ is a prime and $a$,$b \in \mathbb{Z}_p$. As usual, we assume that this generator outputs the $k$ most significant bits at each iteration. When $a$ and $b$ are known, the polynomial $h(y_0, y_1) = y_0y_1 + c_0y_0 + c_1y_1 + c$ can be constructed, using two consecutive outputs, where $c$, $c_0$, $c_1$ are constant values.

Let us now look at the link between the geometrical representation of the polynomial $h(y_0, y_1)$, namely a square, and those of $f(y_0, y_1) = y_1 - c_0 y_0 - a y_0^2 + c \mod p$, which corresponds to the polynomial defined for the quadratic generator with two outputs when $a$ and $b$ are known, that can be represented as a triangle. The denominator appearing in the value $\delta$, coming from Equation (2), can be

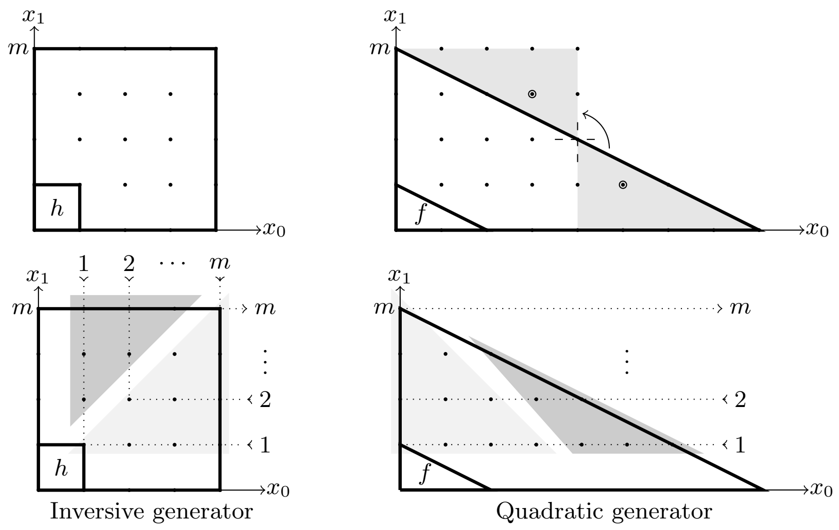

**Fig. 1.** Geometrical link between $f$ and $h$

seen as the sum of the coordinates of each point belonging to the form defined by the polynomial. For the inversive generator, this sum can be expressed as:

$$\sum_{x_0=0}^{m} \sum_{x_1=0}^{m} x_0 + x_1 = m(m+1)^2 = \sum_{x_1=0}^{m} \sum_{x_0=0}^{2m-x_1} x_0 + x_1$$

The collection of polynomials involved in the quadratic generator case, gives the following formula, corresponding to the numerator:

$$\sum_{x_1=1}^{m} \sum_{x_0=0}^{2(m-x_1)} x_1 = \frac{1}{6} m(m+1)(2m+1) = \sum_{x_1 x_0=1}^{m} \sum_{x_1=0}^{m-x_1 x_0} x_1 x_0 + \sum_{x_1 x_0=1}^{m-1} \sum_{x_0=1}^{m-x_1 x_0} x_1 x_0.$$

Figure 1 shows the geometrical link between these two generators (on the top, the set of monomials ; on the bottom, the collection of polynomials). When working with more polynomials, the situation is identical. Moreover, when $a$ and $b$ are unknown, the polynomial used to build the collection in the inversive generator case is similar to those used in the quadratic generator's one. The obtained bound is also 1/12 and with more consecutive outputs, it tends to 1/3, similarly to the quadratic generator.

| | 2 outputs | 3 outputs | 4 outputs | 5 outputs | 6 outputs | 7 outputs |
|---|---|---|---|---|---|---|
| previous bound | 0.25 | 0.286 | 0.3 | 0.308 | 0.313 | 0.316 |
| new achievable bound | 0.321 $m = 13$ | 0.39 $m = 9$ | 0.40 $m = 8$ | 0.401 $m = 8$ | 0.401 $m = 8$ | 0.401 $m = 8$ |
| new asymptotic bound | 1/3 | 3/7 | 7/15 | 15/31 | 31/63 | 63/127 |

**Table 1.** Some bounds for the inversive generator, $a$,$b$ known

## 4 The Pollard generator

The recursive sequence of the Pollard generator is defined as $v_i = v_{i-1}^2 + b \mod p$ with $b \in \mathbb{Z}_p$ (*i.e.* it is a particular instance of the quadratic generator where the constant $a$ is equal to 1). As a consequence, the attack scenario is exactly the same as in the previous section when $b$ is known to the attacker. However, if one takes advantage of the fact that $a$ is fixed to 1, a specific analysis can be made and a better bound can be obtained. To reach such a result, we use a novel technique, called unravelled linearization whose description is provided below.

**Unravelled linearization.** In 2009, Hermann and May introduced a new technique called *unravelled linearization* [12] that allows to work with smaller lattices by optimizing the way the initial polynomial is written. It consists in improving the bounds, see Equation (2), by reducing the number of monomials in $M$ while keeping the same amount of powers of $N$ in the right hand side of the equation.

Let us show what happens on a toy example, say $f(x, y) = x^2 + x + y$ having a root $(x_0, y_0)$ modulo $N$ where $|x_0| < X$ and $|y_0| < Y$ with $X = Y$. The idea is to find the better way of linearizing $f$ before proceeding to Coppersmith's construction. If we fix $u = x^2$, the polynomial $f$ becomes $g(u, x, y) = u+x+y$ and the bounds on the root can be determined by the following formula $UXY < N$. Knowing that $U = X^2$, this leads to $X = N^{1/4}$. Now, let us take another smarter linearization, say $u = x^2+y$, leading to the polynomial $g(u, x) = u+x$. This time, the formula becomes $UX < N$, what leads to the improved bound $X = N^{1/3}$. In this case, the "weight" of $y$ is hidden in $u$ by the weight of $x^2$.

One need to use another tricky manipulation to conclude. Let us go back to our toy example $g(u, x) = u + x$ and construct the original matrix defined by Coppersmith taking the collection $\{g, g^2\}$. This leads to the matrix $\mathcal{M}$ that follows, thus reaching the asymptotic bound $U^4X^4 < N^3$, what gives $X < N^{1/4}$.

$$\underbrace{\left(\begin{array}{ccccc|cc} 1/U &&&&& 1 & 0 \\ & 1/X &&&& 1 & 0 \\ && 1/X^2 &&& 0 & 1 \\ &&& 1/UX && 0 & 2 \\ &&&& 1/U^2 & 0 & 1 \\ \hline && 0 &&& N & \\ &&&&& & N^2 \end{array}\right)}_{\mathcal{M}} \begin{array}{c} u \\ x \\ x^2 \\ ux \\ u^2 \\ \\ \\ \end{array} \qquad \underbrace{\left(\begin{array}{ccccc|cc} 1/U &&&&& 1 & 1 \\ & 1/X &&&& 1 & 0 \\ && \mathbf{1/Y} &&& 0 & -1 \\ &&& 1/UX && 0 & 2 \\ &&&& 1/U^2 & 0 & 1 \\ \hline && 0 &&& N & \\ &&&&& & N^2 \end{array}\right)}_{\mathcal{M}'} \begin{array}{c} u \\ x \\ \boldsymbol{y} \\ ux \\ u^2 \\ \\ \\ \end{array}$$

But here is the point: by definition of $u$, the monomial $x^2$ can easily be written as $u - y$, thus allowing to express the polynomial $g^2$ as $g^2 = u^2 + 2ux + u - y$. Such a manipulation leads to the matrix on the right hand side, say $\mathcal{M}'$. In this case, the obtained bounds on the root can be reformulated as $U^4X^2Y < N^3$ what gives the improved result $X < N^{3/11}$. This benefit can be understood by the fact that we have managed to decrease the weight of the monomials in the set $M$ by 1 while keeping the exact number of powers of $N$ appearing in the right hand side of Equation (2). Such manipulations are quite hard to proceed, they strongly rely on the linearization chosen for the initial polynomial $f$ (a more detailed discussion on the importance of the choice of the linearization can be found in the full version of the paper).

### 4.1 Case $F$ known

**Attack with two consecutive outputs.** Let us first assume that the adversary is given access to two consecutive outputs of the generator, namely $w_0$ and $w_1$. Knowing that $v_0 = 2^{\pi-k}w_0 + x_0$ and $v_1 = 2^{\pi-k}w_1 + x_1$, we reach the same relation as those previously obtained for the quadratic case:

$$x_1 - 2^{\pi-k+1}w_0x_0 - x_0^2 - b + 2^{\pi-k}w_1 - 4^{\pi-k}w_0^2 = 0 \mod p$$

Let us denote by $f(y_0, y_1)$ the polynomial $y_1 - c_0y_0 - y_0^2 + d_0$ where the coefficients $c_0 = 2^{\pi-k+1}w_0$ and $d_0 = -b + 2^{\pi-k}w_1 - 4^{\pi-k}w_0^2$ are known to the attacker. As usual, its goal consists in recovering the small modular root $(x_0, x_1)$ of $f(y_0, y_1)$.

To solve this problem, we use the unravelled linearization technique. As already stated, the first step consists in choosing a good linearization for $f$. In this particular case, we set $u = y_1 - y_0^2$, what leads to the following polynomial $g(y_0, u) = u - c_0y_0 + c \mod p$. In that case, the bound on $u$ can thus be expressed as $U = X_0^2$.

Let us now consider the collection of polynomials defined as $y_0^j g^i(y_0, u)$ with $i + j \leq m$ and $i > 0$. The list of monomials appearing in that collection can be described as $M = \left\{ y_0^j u^i \quad | \quad i + j \leq m \right\}$. Initially, we use this set of polynomials to construct the matrix defined by Coppersmith, as in Section 2.2. In that case, the right-hand side (*resp.* the left-hand side of) of (2) can easily be expressed as $p$ to the power

$$\sum_{i=1}^{m}\sum_{j=0}^{m-i} i = \frac{1}{6}m^3 + o(m^3) \qquad \left( \textit{resp. } \delta \sum_{i=0}^{m}\sum_{j=0}^{m-i} 2i + j = \frac{\delta}{2}m^3 + o(m^3) \right)$$

The idea of the unravelled linearization technique is to improve the bound on $\delta$ by decreasing the weight of the monomials. To do so, one should proceed to a "back-substitution" in the constructed matrix, as explained in the previous section. In that particular case, knowing that $y_0^2 = y_1 - u$, the following replacement is done (for all monomials $\mu$ such that $\mu \cdot y_0^2 \in M$): $\mu \cdot y_0^2 \to \mu \cdot y_1 - \mu \cdot u$. It is obvious that the presence of $\mu \cdot y_0^2$ in the set $M$ implies those of $\mu \cdot u$. As a consequence, doing such a manipulation allows to replace the quantity $\mu \cdot y_0^2$ by $\mu \cdot y_1$ thus decreasing by "1" the weight on the monomials. If we express the collection $M$ as $M = \left\{y_0^{2b+a}u^i \quad | \quad a \in \{0,1\} \wedge a + 2b + i \leq m\right\}$, after the back-substitution, we obtain the set $M' = \left\{y_1^b y_0^a u^i \quad | \quad a \in \{0,1\} \wedge a + 2b + i \leq m\right\}$. In that case, the *new* left-hand side in Equation (2) becomes $p$ raised to the power:

$$\delta \sum_{a=0}^{1} \sum_{i=0}^{m-a} \sum_{b=0}^{\lfloor \frac{m-i-a}{2} \rfloor} (a + b + 2i) = \delta \frac{5}{12} m^3 + o(m^3)$$

Thus, the corresponding asymptotic bound on $\delta$ becomes:

$$\delta < \frac{1/6m^3 + o(m^3)}{5/12m^3 + o(m^3)} \xrightarrow[m \to +\infty]{} \frac{2}{5}.$$

This bound is better than $\delta < 5/14$, previously obtained by Blackburn et al. [11] when working with one polynomial. One can also notice that $2/5$ is exactly the bound obtained in [12] for attacking the Blum-Blum Shub generator.

**More consecutive outputs.** In that case, one can easily generalize the method explained before in the same way as what has been done for the Blum-Blum Shub generator, thus reaching the bound $\delta < 1/2$. Details are left to the reader.

### 4.2 Case $F$ unknown

**Attack with three consecutive outputs.** Let us now consider the case of an adversary having access to three consecutive outputs of the generator. In that case, writing two consecutive recurrence relations and subtracting both of them leads to:

$$\begin{aligned} -x_1^2 + x_0^2 + x_2 + \underbrace{2^{\pi-k+1} w_0}_{c_0} x_0 - \underbrace{(2^{\pi-k+1} w_1 + 1)}_{c_1} x_1 \\ + \underbrace{2^{\pi-k} w_2 - 2^{\pi-k} w_1 + 4^{\pi-k} w_0^2 - 4^{\pi-k} w_1^2}_{c} = 0 \mod p. \end{aligned}$$

The adversary wants to recover the small modular root $(x_0, x_1, x_2)$ of the polynomial $f(y_0, y_1, y_2) = -y_1^2 + y_0^2 + y_2 + c_0 y_0 - c_1 y_1 + c$. To do so, we use again the unravelled linearization technique. To linearize the polynomial $f$, we set $u = -y_1^2 + y_0^2 + y_2$, reaching the new following expression $g(u, y_0, y_1) = u + c_0 y_0 - c_1 y_1 + c$. Let us now consider the collection of polynomials defined as $y_0^k y_1^j g^i$ with $i + j + k \leq m$ and $i > 0$. In that case, the list of involved monomials can easily be expressed as $M = \left\{u^i y_1^j y_0^k \quad | \quad i + j + k \leq m\right\}$. Thus, the right-hand side of Coppersmith's Equation (2) is given by $p$ to the power:

$$\sum_{i=1}^{m} \sum_{j=0}^{m-i} \sum_{k=0}^{m-i-j} i = \frac{1}{24} m^4 + o(m^4).$$

Before evaluating the weight of the monomials in $M$, we perform some back-substitutions. In this case, the rule given by the linearization is (for all monomials $\mu$ such that $\mu \cdot y_1^2 \in M$): $\mu \cdot y_1^2 \to \mu \cdot y_0^2 + \mu \cdot y_2 - \mu \cdot u$. One can notice that the presence of the monomial $\mu \cdot y_1^2$ in the set $M$ automatically implies those of $\mu \cdot y_0^2$ and $\mu \cdot u$. Thus, each monomial of the form $\mu \cdot y_1^2$ can be replaced by one of those $\mu \cdot y_2$ in the constructed matrix, again decreasing by "1" the weight on the involved monomials. The shape of the *new* constructed set $M$ is then:

$$\left\{ u^i y_2^b y_1^a y_0^k \quad | \quad a \in \{0,1\} \wedge i+k+a+2b \le m \right\}$$

In that case, the *new* left hand side of Equation (2) becomes:

$$\delta \sum_{a=0}^{1} \sum_{i=0}^{m-a} \sum_{b=0}^{\lfloor \frac{m-i-a}{2} \rfloor} \sum_{k=0}^{m-i-a-2b} (a+b+2i+k) = \frac{7\delta}{48} m^4 + o(m^4)$$

which leads to the following bound on $\delta$:

$$\delta < (1/24 m^4 + o(m^4))/(7/48 m^4 + o(m^4)) \xrightarrow[m \to +\infty]{} 2/7.$$

**More consecutive outputs.** Let us assume that the attacker knows $n+2$ consecutive outputs, for $n \ge 2$. We denote $f_i$ the relation between two outputs:

$$f_i = 2^{\pi-k} w_i + y_i - (2^{\pi-k} w_{i-1} + y_{i-1})^2 - b \mod p \quad i \in \{1, \ldots, n\}$$

These polynomials have $(x_i, x_{i-1})$ as a root modulo $p$ and denoting $g_i = f_{i+1} - f_i$ for $i \in \{1, \ldots, n\}$, we have $g_i = -y_i^2 + y_{i-1}^2 + y_{i+1} + c_i y_{i-1} - d_i y_i + e_i \mod p$ for some constants $c_i, d_i, e_i$ known to the adversary.

Knowing the set of polynomials $\{g_1, \ldots, g_n\}$, the attacker wants to recover the unknown values $x_i$. To do so, we use again the unravelled linearization technique by choosing $u_i = -y_i^2 + y_{i-1}^2 + y_{i+1}$, what leads to: $g_i = u_i + c_i y_{i-1} - d_i y_i + e_i$. Such polynomials allows us to reach the asymptotic following bound $\delta < \frac{2}{5}$ (details will be given in the full version). In that particular case, we think this bound could be improved to $\delta < 1/2$, following the discussion from Remark 1.

| Number of outputs | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|
| Previous bound [3] | 0.261 | 0.286 | 0.3 | 0.308 | 0.313 | 0.316 |
| Our achievable bound | 0.278 | 0.319 | 0.324 | 0.324 | 0.324 | 0.324 |
| Our asymptotic bound | 2/7 | 6/17 | 14/37 | 30/77 | 62/157 | 126/317 |

**Table 2.** Theoretical and experimental bounds for the Pollard generator, $b$ unknown

## References

1. A. Bauer et A. Joux – “Toward a rigorous variation of Coppersmith’s algorithm on three variables”, *Advances in Cryptology – EUROCRYPT 2007* (M. Naor, éd.), Lect. Notes Comput. Sci., vol. 4515, Springer, 2007, p. 361–378.
2. S. R. Blackburn, D. Gomez-Perez, J. Gutierrez et I. Shparlinski – “Predicting the inversive generator”, *9th IMA International Conference on Cryptography and Coding* (K. G. Paterson, éd.), Lect. Notes Comput. Sci., vol. 2898, Springer, 2003, p. 264–275.
3. — , “Predicting nonlinear pseudorandom number generators”, *Math. Comput.* **74** (2005), no. 251, p. 1471–1494.
4. — , “Reconstructing noisy polynomial evaluation in residue rings”, *J. Algorithms* **61** (2006), no. 2, p. 47–59.
5. J. Blömer et A. May – “A tool kit for finding small roots of bivariate polynomials over the integers”, *Advances in Cryptology – EUROCRYPT 2005* (R. Cramer, éd.), Lect. Notes Comput. Sci., vol. 3494, Springer, 2005, p. 251–267.
6. J. Boyar – “Inferring sequences produced by a linear congruential generator missing low-order bits”, *Journal of Cryptology* **1** (1989), no. 3, p. 177–184.
7. — , “Inferring sequences produced by pseudo-random number generators”, *J. ACM* **36** (1989), no. 1, p. 129–141.
8. D. Coppersmith – “Finding a small root of a bivariate integer equation; factoring with high bits known”, *Advances in Cryptology – EUROCRYPT’96* (U. M. Maurer, éd.), Lect. Notes Comput. Sci., vol. 1070, Springer, 1996, p. 178–189.
9. — , “Finding a small root of a univariate modular equation”, *Advances in Cryptology – EUROCRYPT’96* (U. M. Maurer, éd.), Lect. Notes Comput. Sci., vol. 1070, Springer, 1996, p. 155–165.
10. D. Gómez, J. Gutierrez et Á. Ibeas – “Cryptanalysis of the quadratic generator”, *Progress in Cryptology - INDOCRYPT 2005: 6th International Conference in Cryptology in India* (S. Maitra, C. E. V. Madhavan et R. Venkatesan, éds.), Lect. Notes Comput. Sci., vol. 3797, Springer, 2005, p. 118–129.
11. — , “Attacking the Pollard generator”, *IEEE Transactions on Information Theory* **52** (2006), no. 12, p. 5518–5523.
12. M. Herrmann et A. May – “Attacking power generators using unravelled linearization: When do we output too much?”, *Advances in Cryptology – ASIACRYPT 2009* (M. Matsui, éd.), Lect. Notes Comput. Sci., vol. 5912, Springer, 2009, p. 487–504.
13. N. Howgrave-Graham – “Finding small roots of univariate modular equations revisited”, *6th IMA International Conference on Cryptography and Coding* (M. Darnell, éd.), Lect. Notes Comput. Sci., vol. 1355, Springer, 1997, p. 131–142.
14. E. Jochemsz et A. May – “A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants”, *Advances in Cryptology – ASIACRYPT 2006* (X. Lai et K. Chen, éds.), Lect. Notes Comput. Sci., vol. 4284, Springer, 2006, p. 267–282.
15. A. Joux et J. Stern – “Lattice reduction: A toolbox for the cryptanalyst”, *Journal of Cryptology* **11** (1998), no. 3, p. 161–185.
16. A. K. Lenstra, H. W. j. Lenstra et L. Lovász – “Factoring polynomials with rational coefficients.”, *Math. Ann.* **261** (1982), no. 4, p. 515–534.
17. J. Stern – “Secret linear congruential generators are not cryptographically secure”, *FOCS*, IEEE, 1987, p. 421–426.