New Asymptotic Results on Predicting Polynomial Congruential Generators

Yansong Feng[1,2], Abderrahmane Nitaj[3], Yanbin Pan[1,2], and Mengce Zheng[4,✉]

[1] State Key Laboratory of Mathematical Sciences, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
[2] School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China
{fengyansong,panyanbin}@amss.ac.cn
[3] LMNO, University of Caen Normandy, Caen, France
abderrahmane.nitaj@unicaen.fr
[4] Zhejiang Wanli University, Ningbo, China
mczheng@zwu.edu.cn

Abstract. We investigate cryptanalytic attacks for predicting polynomial congruential generators (PCGs) from arbitrarily long sequences of consecutive truncated outputs. Such attacks naturally yield systems of modular polynomial equations, which can be solved using Coppersmith's method.
However, deriving the corresponding success conditions by hand requires substantial combinatorial summation, which is typically both time-consuming and tedious. Existing automated Coppersmith methods assist with this computation, but they generally provide only numerical bounds for fixed systems, whereas in our setting the number of equations is itself a parameter. Inspired by the Newton-polytope framework of Feng et al. (Crypto 2025), we express the success condition as the volume of a high-dimensional polytope and compute it symbolically as a function of the number of outputs.
We improve existing asymptotic bounds for the Pollard generator and for linear congruential generators. We also obtain new attacks on quadratic congruential generators (QCGs) with partially known coefficients and on perturbed Power Generators with an unknown masking constant, matching bounds previously achieved only under stronger assumptions on the map $F(x)$.

Keywords: Asymptotic bound, Coppersmith's method, Lattice reduction, Polynomial congruential generator

1 Introduction

Polynomial congruential generators (PCGs) are among the most classical algebraic constructions for producing long deterministic sequences from a short secret seed. Starting from a state $v_0 \in \mathbb{Z}_p$, a PCG iterates

$$v_{i+1} \equiv F(v_i) \pmod{p},$$

where $F \in \mathbb{Z}_p[x]$ is a low-degree polynomial, and releases either the full state or a truncated part of it, typically the most significant bits. The update rule is simple and efficient, but this algebraic structure also makes PCGs natural targets for lattice-based cryptanalysis.

The Linear Congruential Generator (LCG), with $F(x) = ax+b$, has long been known to be predictable from sufficiently many outputs [5, 17, 23, 25]. Boyar [5] showed that, given the map F, a sufficiently long run of full outputs allows one to recover the seed in time polynomial in the bit-size of p. Stern [25] showed that predictability can remain possible even when only the most significant bits of the states are revealed. These results motivated the study of nonlinear congruential generators, including Quadratic Congruential Generators (QCGs), Pollard generators, and Power Generators. Nevertheless, a substantial line of work has shown that the algebraic structure of such generators can still be exploited, in particular through Coppersmith-type attacks on systems of modular polynomial equations arising from truncated outputs [1, 3, 4, 10–12, 14].

This paper studies PCGs in the setting where the adversary is given an arbitrarily long sequence of consecutive truncated outputs. We also allow some coefficients of the update polynomial F to be unknown. If the released part of v_i is denoted by w_i, then we can write v_i as

$$v_i = w_i + u_i, \qquad 0 \leq u_i < p^\delta.$$

The parameter δ measures the size of the hidden part of each state: the larger δ is, the fewer bits are revealed to the adversary. Our goal is therefore to determine the largest δ for which one can still recover the hidden value u_i, and hence predict the generator.

For example, consider a QCG

$$v_{i+1} \equiv av_i^2 + b \pmod{p}$$

in the case where a is known and b is unknown. From $n+1$ consecutive truncated outputs, subtracting consecutive recurrence relations eliminates b and gives, for $j = 1, \ldots, n-1$,

$$v_j - v_{j+1} \equiv a(v_{j-1}^2 - v_j^2) \pmod{p}.$$

Substituting $v_i = w_i + u_i$ yields a system

$$f_j(u_{j-1}, u_j, u_{j+1}) \equiv 0 \pmod{p}, \qquad j = 1, \ldots, n-1,$$

whose number of variables and equations grows with n.

Such systems can be attacked using multivariate Coppersmith's methods [6, 16]. However, deriving the corresponding success conditions by hand requires substantial combinatorial summation, which is typically both time-consuming and tedious. Recent automated Coppersmith frameworks provided an automated approach to these calculations [9, 22, 24]. However, these methods are typically tailored to a fixed polynomial system. In our setting, the numbers of variables and equations depend on n. Consequently, while these automated Coppersmith

methods can be applied to several concrete values of n, they do not yield a closed-form expression for δ_n or its limiting value as $n \to \infty$.

We overcome this difficulty by using the Newton polytope based framework of Feng et al. [9]. In this viewpoint, the quantities in the Coppersmith's method become volumes of convex polytopes. However, in our setting, some of these polytopes are not simplices, making direct volume computations difficult. Moreover, these polytopes have dimension $n+2$ with $n \geq 2$, and therefore cannot be visualized directly. Our main observation is that such polytopes can be related to simpler geometric objects through a sequence of transformations: we rescale the polytope, project it, complete it into a larger shape, and then subtract the remaining part. Fig. 1 provides a three-dimensional schematic illustrating the main idea, although the polytope arising in our setting is substantially more complicated.

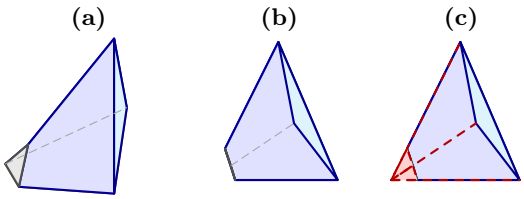


Fig. 1: A three-dimensional schematic for the volume computation. We first stretch and rescale the irregular shape (a) into a more standard truncated tetrahedron (b), and then complete the missing corner to obtain a standard tetrahedron (c) while subtracting the added corner.

Our contribution. We give new asymptotic attacks on several polynomial congruential generators. Tables 1 and 2 summarize the resulting asymptotic bounds. In both tables, a bound δ means that the hidden part of each state may have size up to p^δ.

First, we improve two bounds in previously studied settings. For Pollard generators, our asymptotic bound improves the result of Bauer et al. [1, Section 4.2] from 2/5 to 1/2, matching their conjectured threshold. For LCGs with unknown multiplier and known increment, we improve the bound of Zhang et al. [27] from 1/3 to 1/2.

Table 1: Improved asymptotic bounds.

Setting	Previous bound	This work
Pollard Generator $x^2 + c$, unknown c	2/5 [1]	1/2
LCG $ax + b$, unknown a, known b	1/3 [27]	1/2

Second, we analyze several settings that were not covered in previous works. Table 2 compares our results in these new settings with the closest existing results. For QCGs, we first consider the case where a is known while b is unknown, and obtain a bound matching the previously known result for the setting in which both a and b are known. For perturbed Power Generators of the form $F(x) = x^e + c$ with unknown c, we obtain the bound $1/e$, matching the result of Herrmann and May [14] for the standard Power Generator with $F(x) = x^e$. We also study the setting in which a is unknown and b is known for QCGs. However, in this case, we obtain no improvement over the setting where both a and b are unknown [1].

Table 2: New settings of QCG and Power Generator.

Generator	Bound	Previous setting	This work
QCG $ax^2 + b$	$1/2$	known a, b [1]	known a, unknown b
Power Generator $x^e + c$	$1/e$	$c = 0$ [14]	unknown c

Related work. Ding et al. [7] recently proposed a linear programming approach to computing asymptotic bounds. In Section 4.4 of their paper, they derived the same result as that stated in Theorem 5 for linear congruential generators with an unknown parameter a.

Organization. Section 2 introduces lattice reduction, Coppersmith's method, and the Newton polytope formulation of automated Coppersmith analysis. Section 3 gives the attacks on QCGs with partially known coefficients and includes Pollard generators as a special case. Section 4 gives the attacks perturbed Power Generators with an unknown masking constant. Section 5 gives the improved attack on LCGs with an unknown multiplier. Section 6 concludes. The code for our experiments is available at `https://github.com/MengceZheng/PCGs`. All experiments were conducted on a personal computer equipped with an 11th-generation Intel Core i7-1165G7 processor (2.80 GHz) and 16 GB of RAM, running Ubuntu 22.04 under the Windows Subsystem for Linux (WSL).

2 Preliminaries

This section reviews the tools underlying our lattice-based attacks. We recall the LLL lattice reduction algorithm [19], Coppersmith's method [6] together with the refinement of Howgrave-Graham [15], and the recent framework of Feng et al. [9]. For further details, we refer the reader to [20, 21, 28].

2.1 Polynomials

Let $f \in \mathbb{Z}[x_1, \ldots, x_k]$ be a polynomial in k variables, written as

$$f(x_1, \ldots, x_k) = \sum_{i_1, \ldots, i_k \in \mathbb{N}} \alpha_{i_1, \ldots, i_k} x_1^{i_1} \cdots x_k^{i_k},$$

where $\alpha_{i_1, \ldots, i_k} \in \mathbb{Z}$. A term $x_1^{i_1} \cdots x_k^{i_k}$ with $\alpha_{i_1, \ldots, i_k} \neq 0$ is called a *monomial* of f.

A *monomial order* $\prec$ on the set of monomials $\mathcal{M}$ is a total order such that $1 \prec \lambda$ for all $\lambda \in \mathcal{M}$, and if $\lambda_1 \prec \lambda_2$, then $\lambda\lambda_1 \prec \lambda\lambda_2$ for all $\lambda \in \mathcal{M}$. Given $\prec$, the *leading monomial* $\mathrm{LM}(f)$ is the maximal monomial of f under $\prec$, and the *leading coefficient* $\mathrm{LC}(f)$ is its coefficient.

Throughout, we assume $x_1 \succ \cdots \succ x_k$ and use two standard monomial orders. The lexicographic order ($\prec_{\mathrm{lex}}$) compares exponents of x_1, then x_2, and so on; for example, $1 \prec x_2 \prec x_2^2 \prec x_1 \prec x_1 x_2 \prec x_1^2$. The graded reverse lexicographic order first compares total degree and then reverses the lexicographic comparison; for example, $x_3^2 \prec x_2 x_3 \prec x_1 x_3 \prec x_2^2 \prec x_1 x_2 \prec x_1^2$.

2.2 Lattices and Coppersmith's Method

Let n and ω be positive integers with $\omega \leq n$. Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_\omega$ in $\mathbb{R}^n$ be ω linearly independent vectors. The *lattice* Λ spanned by $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_\omega$ is the set of all their integer linear combinations, namely

$$\Lambda = \left\{ \sum_{i=1}^{\omega} z_i \mathbf{b}_i \mid z_i \in \mathbb{Z} \right\}.$$

The *determinant* of the lattice, denoted $\det(\Lambda)$, is given by $\sqrt{\det(BB^{\mathsf{T}})}$, where B is the matrix with rows $\mathbf{b}_i$. When the lattice has full rank ($\omega = n$), the determinant simplifies to $\det(\Lambda) = |\det(B)|$.

The following LLL algorithm [19] efficiently computes a reduced lattice basis in polynomial time.

Lemma 1. *Let Λ be an ω-dimensional lattice generated by $\mathbf{b}_1, \ldots, \mathbf{b}_\omega$. The LLL algorithm returns a reduced basis $(\mathbf{v}_1, \ldots, \mathbf{v}_\omega)$ satisfying*

$$\|\mathbf{v}_i\| \leq 2^{\frac{\omega(\omega-1)}{4(\omega+1-i)}} \cdot \det(\Lambda)^{\frac{1}{\omega+1-i}}, \text{ for } i = 1, \ldots, \omega.$$

Howgrave-Graham's Lemma [15] gives a sufficient condition under which a root modulo R is also a root over the integers. Consider an integer polynomial

$$h(x_1, \ldots, x_n) = \sum c_{i_1, \ldots, i_n} x_1^{i_1} \cdots x_n^{i_n},$$

with its Euclidean norm defined by

$$\|h\| = \sqrt{\sum |c_{i_1, \ldots, i_n}|^2}.$$

Lemma 2. *Let $h(x_1, \ldots, x_n) \in \mathbb{Z}[x_1, \ldots, x_n]$ be a polynomial with at most ω monomials. Let R be a modulus, and let $X_1, \ldots, X_n$ be bounds on the absolute values of the variables. If there exists a solution $(x_1^\star, \ldots, x_n^\star)$ with $|x_i^\star| \leq X_i$ such that $h(x_1^\star, \ldots, x_n^\star) \equiv 0 \bmod R$ and $\|h(X_1 x_1, \ldots, X_n x_n)\| < R/\sqrt{\omega}$, then $h(x_1^\star, \ldots, x_n^\star) = 0$ over the integers.*

The workflow of Coppersmith's method is as follows. First, let $f(x_1, \ldots, x_n)$ be a polynomial such that $f(x_1^\star, \ldots, x_n^\star) \equiv 0 \bmod M$ with bounds $|x_i^\star| \leq X_i$. Next, choose an integer m and construct shift polynomials satisfying $\mathfrak{s}_k(x_1^\star, \ldots, x_n^\star) \equiv 0 \bmod M^m$, typically of the form $x_1^{i_{k_1}} \cdots x_n^{i_{k_n}} f^{j_k} M^{m-j_k}$. Then scale each polynomial $\mathfrak{s}_k$ by the bounds X_i and form a matrix B from their coefficient vectors; this matrix generates a lattice Λ. Applying LLL to B produces short vectors $\mathbf{v}_1, \ldots, \mathbf{v}_\ell$, which correspond to new polynomials that vanish at $(x_1^\star, \ldots, x_n^\star)$ under the above condition. Finally, assuming these new polynomials are algebraically independent, one computes the solution using resultants or a Gröbner basis computation [2]. In general, Gröbner basis computation is doubly exponential in the degree of the polynomials.

Combining Lemma 1 and Lemma 2 yields the usual Coppersmith criterion. Suppose that we compute the first ℓ reduced vectors $(\mathbf{v}_1, \ldots, \mathbf{v}_\ell)$ of a lattice Λ of dimension ω. To ensure that one of the associated polynomials vanishes at the solution over $\mathbb{Z}$, it suffices to satisfy

$$2^{\frac{\omega(\omega-1)}{4(\omega+1-\ell)}} \cdot \det(\Lambda)^{\frac{1}{\omega+1-\ell}} < \frac{R}{\sqrt{\omega}}.$$

For asymptotic analyses using large-dimensional lattices, this condition is often simplified to

$$\det(\Lambda) < R^{\omega-\varepsilon}, \tag{1}$$

where ε denotes a term that is negligible compared to ω.

Multivariate Coppersmith methods are heuristic, since there is no general guarantee that the constructed polynomials are algebraically independent. Nevertheless, this assumption holds in most practical instances and is standard in cryptanalytic applications. We adopt it throughout.

Assumption 1. *The lattice-based Coppersmith's method yields algebraically independent polynomials. The common roots of these polynomials can be computed efficiently using Gröbner basis computation.*

2.3 Automated Coppersmith Framework

An effective lattice-basis construction is essential. For a single polynomial equation, Jochemsz and May [16] proposed a systematic strategy for selecting shift polynomials. Under a suitable monomial order, their construction yields a triangular basis matrix, so $\det(\Lambda)$ is simply the product of the diagonal entries.

More recently, Meers and Nowakowski [22] and Ryan [24] studied automated Coppersmith methods, including generalization to a system of polynomial equations. Feng et al. [9] further incorporated techniques from additive combinatorics

to compute the lattice dimension and determinant. Their analysis relies on results of Khovanskii [18]. For a polynomial $f(x_1, \ldots, x_k)$, define

$$A(f) = \left\{ (i_1, \ldots, i_k) \in \mathbb{Z}_{\geq 0}^k \;\middle|\; x_1^{i_1} \cdots x_k^{i_k} \text{ is a monomial of } f \right\}.$$

For any finite set $A \subseteq \mathbb{Z}_{\geq 0}^k$, we write

$$N(A) = \operatorname{conv}(A) = \left\{ \sum_{a \in A} \lambda_a a \;\middle|\; \lambda_a \geq 0,\ \sum_{a \in A} \lambda_a = 1 \right\}$$

for its convex hull. For simplicity, we write $N(f)$ instead of $N(A(f))$, and refer to it as the Newton polytope of f.

Feng et al. [9] also formulated the lattice construction for a system of polynomial equations $\mathcal{F} = \{f_1, \ldots, f_s\}$ modulo known p with k variables $x_1, \ldots, x_k$. Let $A(\mathcal{F}) = \bigcup_{r=1}^s A(f_r)$. If b_r is the exponent vector of the leading monomial of f_r, we write $B = \{b_1, \ldots, b_s\}$.

Fix a positive integer m. Define the sets

$$S_\ell = \{x_1^{i_1} \cdot \ldots \cdot x_k^{i_k} \mid (i_1, \ldots, i_k) \in (m - \ell)A(\mathcal{F}) + \ell B\}, \tag{2}$$

where ℓB denotes the ℓ-fold sumset of B for $0 \leq \ell \leq m$, and we set $S_{m+1} = \emptyset$, so that $S_m \subset \ldots \subset S_0$. The strategy constructs basis polynomials for every monomial $\lambda = x_1^{i_1} \cdot \ldots \cdot x_k^{i_k} \in S_\ell \setminus S_{\ell+1}$:

Step 1: Choose $\ell_1, \ldots, \ell_s$ with $\sum_{r=1}^s \ell_r = \ell$ such that

$$(i_1, \ldots, i_k) \in (m - \ell)A(\mathcal{F}) + \sum_{r=1}^s \ell_r b_r.$$

Step 2: Construct $g_\lambda(x_1, \ldots, x_k)$ as follows:

$$g_\lambda(x_1, \ldots, x_k) = \frac{\lambda}{\prod_{r=1}^s \operatorname{LM}(f_r)^{\ell_r}} \prod_{r=1}^s f_r^{\ell_r} p^{m-\ell}. \tag{3}$$

The coefficient vectors of all scaled shifts $g_\lambda(x_1 X_1, \ldots, x_k X_k)$ form the basis of the lattice Λ. The basis matrix of Λ can be made diagonal in the cases considered below, and the solvability condition from (1) becomes

$$\prod_{j=1}^k X_j^{\sigma_j} p^{\sigma_p} < p^{\sigma_m - \varepsilon_0},$$

where σ_m, σ_j, and σ_p are the corresponding sums of exponents over all shift polynomials.

Following [9], we use the following two lemmas of Khovanskii [18] to compute the leading terms of these sums as functions of m.

Lemma 3 (Corollary 2, [18]). *Suppose* $\mathbf{0} \in A \subseteq \mathbb{Z}^k$ *and* A *has finite index* $\operatorname{ind} A$ *in* $\mathbb{Z}^k$*. Then there exists* m_0 *such that for all* $m \geq m_0$*, the number of integer points in the scaled polytope* mA *is a polynomial in* m *of degree* k*. The leading coefficient equals the integral of* 1 *over* $N(A)$ *divided by* $\operatorname{ind} A$*, given by*

$$\lim_{m\to\infty} \frac{|mA|}{m^k} = \frac{\int_{N(A)} 1 \, \mathrm{d}V}{\operatorname{ind} A}.$$

For the lattice determinant, one also needs sums of homogeneous polynomials ϕ, such as the projection polynomial ϕ_{j_0}:

$$\begin{aligned} \phi_{j_0} : \mathbb{Z}^k &\longrightarrow \mathbb{Z}, \\ (i_1, \ldots, i_k) &\longmapsto i_{j_0}. \end{aligned}$$

For such sums, we use the following lemma.

Lemma 4 (Lemma 10, [9]). *Suppose* $\mathbf{0} \in A \subseteq \mathbb{Z}^k$ *and* A *has finite index* $\operatorname{ind} A$ *in* $\mathbb{Z}^k$*. Then there exists* m_0 *such that for all* $m \geq m_0$*, the sum*

$$\sum_{(i_1,\ldots,i_k)\in mA\cap\mathbb{Z}^k} i_{j_0}$$

is a polynomial in m *of degree* $k+1$*. The leading coefficient equals the integral of* x_{j_0} *over* $N(A)$ *divided by* $\operatorname{ind} A$*, given by*

$$\lim_{m\to\infty} \frac{\sum_{(i_1,\ldots,i_k)\in mA} i_{j_0}}{m^{k+1}} = \frac{\int_{N(A)} x_{j_0} \, \mathrm{d}V}{\operatorname{ind} A}.$$

In our applications, $A(\mathcal{F})$ may be replaced by any finite set A satisfying $A(\mathcal{F}) \subseteq A$. Let N be the convex hull of A. Since $\dim(\Lambda)$ is governed by the size of mA, we have

$$\lim_{m\to\infty} \frac{\sigma_m}{m^{k+1}} = \lim_{m\to\infty} \frac{|mA|}{m^k} = \frac{\int_N 1 \, \mathrm{d}V}{\operatorname{ind} A}.$$

For σ_{j_0}, we have

$$\lim_{m\to\infty} \frac{\sigma_{j_0}}{m^{k+1}} = \lim_{m\to\infty} \frac{\sum_{(i_1,\ldots,i_k)\in mA} i_{j_0}}{m^{k+1}} = \frac{\int_N x_{j_0} \, \mathrm{d}V}{\operatorname{ind} A}.$$

For σ_p, given the sets A and B above, Feng et al. defined $\widetilde{A}$ by

$$\begin{aligned} \widetilde{A} = \{(i_1, \ldots, i_k, 1) \mid (i_1, \ldots, i_k) \in A\} \\ \cup \{(i_1, \ldots, i_k, 0) \mid (i_1, \ldots, i_k) \in B\} \subseteq \mathbb{Z}^{k+1}, \end{aligned}$$

and let $\widetilde{N}$ be the convex hull of $\widetilde{A}$. Hence, we have

$$\lim_{m\to\infty} \frac{\sigma_p}{m^{k+1}} = \frac{\int_N 1 \, \mathrm{d}V}{\operatorname{ind} A} - \frac{\int_{\widetilde{N}} 1 \, \mathrm{d}V}{\operatorname{ind} \widetilde{A}}.$$

Feng et al. [9] also considered the Jochemsz-May Extended Strategy and generalized it to systems of polynomial equations; further details appear in [8]. An open-source implementation for computing these asymptotic bounds is available at `https://github.com/fffmath/AsymptoticBounds`. We refer to the approach without additional shifts as the *Basic Strategy*. For many polynomial systems, it is useful to include additional shifts in one variable; we refer to this variant as the *Extended Strategy*.

Remark 1. *We study asymptotic PCG predictability using the asymptotic solvability condition of Coppersmith's method. To avoid repetition, we use the term "asymptotic bound" for our main results and omit the term ε when no ambiguity arises.*

2.4 Experimental Environment

To validate the attacks, we performed numerical experiments with randomly generated parameters for small values of n. All experiments were carried out using `SageMath` [26] on a machine running 64-bit Windows 10 with Ubuntu 22.04 on WSL 2, equipped with a 2.80 GHz CPU and 16 GB of RAM.

3 Attacks on Quadratic Congruential Generators

Let p be a prime number of bit-length π. A QCG is defined by the recurrence

$$v_{i+1} = F(v_i) \bmod p,$$

where $F(x) = ax^2 + b$ is a quadratic polynomial over $\mathbb{Z}_p$, and v_0 is the secret seed.

We assume that the generator reveals κ most significant bits of each state v_i, with a known output pattern. Thus the internal state can be written as

$$v_i = w_i + u_i,$$

where w_i is known and $u_i \leq 2^{\pi-\kappa} - 1 = p^\delta$ is the unknown least significant part. Our goal is to recover the sequence u_i, assuming access to infinitely many consecutive outputs and that either a or b is known, while maximizing the admissible value of δ.

3.1 The Case of Known a and Unknown b

We use $n+1$ consecutive outputs, which yield n relations

$$v_j = av_{j-1}^2 + b \bmod p, \qquad j = 1, \ldots, n.$$

Subtracting the j-th and $(j+1)$-th relations, for $j = 1, \ldots, n-1$, eliminates b:

$$v_j - v_{j+1} \equiv a(v_{j-1}^2 - v_j^2) \pmod p.$$

Substituting $v_k = w_k + u_k$ for $k = j-1, j, j+1$, multiplying by $a^{-1} \bmod p$, and rearranging yields

$$\begin{aligned} a^{-1}&(w_j - w_{j+1} + u_j - u_{j+1}) \\ &\equiv (w_{j-1}^2 - w_j^2) + 2w_{j-1}u_{j-1} + u_{j-1}^2 - 2w_j u_j - u_j^2 \pmod p. \end{aligned}$$

Letting $x_k = u_{k-1}$ for $k = 1, \ldots, n+1$, we obtain, for each $j = 1, \ldots, n-1$, the modular polynomial equation

$$f_j(x_j, x_{j+1}, x_{j+2}) = x_j^2 + c_{j1}x_{j+1}^2 + c_{j2}x_j + c_{j3}x_{j+1} + c_{j4}x_{j+2} + c_{j0} \equiv 0 \pmod p,$$

where the coefficients c_{jk} are known. Thus we obtain a system of $n-1$ modular equations in $n+1$ variables:

$$\begin{cases} f_1(x_1, x_2, x_3) \equiv 0 \pmod p, \\ f_2(x_2, x_3, x_4) \equiv 0 \pmod p, \\ \qquad \vdots \\ f_{n-1}(x_{n-1}, x_n, x_{n+1}) \equiv 0 \pmod p. \end{cases}$$

The root of interest is

$$(x_1^\star, \ldots, x_{n+1}^\star) = (u_0, \ldots, u_n),$$

where each u_i is the unknown least significant part of v_i.

We apply the Basic Strategy to solve this system.

Theorem 1. *Let a QCG be defined by $F(x) = ax^2 + b$ over $\mathbb{Z}_p$ with known a and unknown b. Given $n+1$ consecutive truncated outputs for fixed $n \geq 2$, the entire sequence can be predicted, provided that Assumption 1 holds and that the single continuous unknown part of size p^δ for each iteration satisfies*

$$\delta < \frac{n-1}{2n+1} - \varepsilon,$$

for some negligible $\varepsilon > 0$. The time complexity is polynomial in $\log p$. As $n \to \infty$, the asymptotic bound is $\delta < 1/2$.

Proof. Let m be a large positive integer. We write

$$\alpha_j^{(k)} = \underbrace{(0, \ldots, 1, \ldots, 0)}_{\text{the } j\text{-th position is } 1} \in \mathbb{Z}^k, \tag{4}$$

for the j-th unit vector in $\mathbb{Z}^k$, and

$$\mathbf{0}^{(k)} = \underbrace{(0, \ldots, 0)}_{\text{all entries are } 0} \in \mathbb{Z}^k$$

for the origin vector. The monomial set for the shift polynomials is determined by the Newton polytopes associated with the system. Let

$$A = \{\mathbf{0}^{(n+1)}, \alpha_1^{(n+1)}, \ldots, \alpha_{n+1}^{(n+1)}, 2\alpha_1^{(n+1)}, \ldots, 2\alpha_n^{(n+1)}\} \subseteq \mathbb{Z}^{n+1},$$
$$B = \{2\alpha_1^{(n+1)}, \ldots, 2\alpha_{n-1}^{(n+1)}\} \subseteq \mathbb{Z}^{n+1}.$$

Then A has index 1. The set B consists of the leading monomials $x_1^2, \ldots, x_{n-1}^2$ of $f_1, \ldots, f_{n-1}$ and is used to compute σ_p.

For each ℓ, we then define the corresponding monomial set

$$S_\ell = \left\{ x_1^{i_1} \cdots x_{n+1}^{i_{n+1}} \mid (i_1, \ldots, i_{n+1}) \in (m-\ell)A + \ell B \right\}.$$

We construct a lattice Λ by the general method of Section 2. For each monomial $\lambda = x_1^{i_1} \cdots x_{n+1}^{i_{n+1}} \in S_\ell \setminus S_{\ell+1}$, define the corresponding shift polynomial $g_\lambda(x_1, \ldots, x_{n+1})$ as follows.

Step 1. Choose $\ell_1, \ldots, \ell_{n-1}$ such that $\sum_{j=1}^{n-1} \ell_j = \ell$ and

$$(i_1, \ldots, i_{n+1}) \in (m-\ell)A + \sum_{j=1}^{n-1} \ell_j \alpha_j^{(n+1)}.$$

Step 2. Define $g_\lambda(x_1, \ldots, x_{n+1})$ by

$$g_\lambda(x_1, \ldots, x_{n+1}) = \frac{\lambda}{\prod_{j=1}^{n-1} \mathrm{LM}(f_j)^{\ell_j}} \cdot \prod_{j=1}^{n-1} f_j^{\ell_j} \cdot p^{m-\ell},$$

The coefficient vectors of $g_\lambda(x_1 X_1, \ldots, x_{n+1} X_{n+1})$ form the basis of a lattice Λ. The basis matrix of Λ can be made diagonal, and the solvability condition from (1) becomes

$$\prod_{j=1}^{n+1} X_j^{\sigma_j} p^{\sigma_p} < p^{\sigma_m - \varepsilon_0}, \tag{5}$$

where $\sigma_m, \sigma_j, \sigma_p$ are sums of exponents over all shift polynomials. Let N be the convex hull of A. By Lemma 3 and Lemma 4, we compute the leading terms of these sums as functions of m.

Computing σ_m. The term σ_m is determined by the lattice dimension and hence by the volume of N:

$$\lim_{m\to\infty} \frac{\sigma_m}{m^{n+2}} = \lim_{m\to\infty} \frac{|mA|}{m^{n+1}} = \int_N 1 \,\mathrm{d}V.$$

Let N_0 be the standard $(n+1)$-dimensional simplex (the convex hull of $\mathbf{0}^{(n+1)}$ and $\alpha_1^{(n+1)}, \ldots, \alpha_{n+1}^{(n+1)}$). Then we have

$$\int_N 1 \,\mathrm{d}V = 2^n \cdot \int_{N_0} 1 \,\mathrm{d}V.$$

Moreover, the volume of N_0 is

$$\int_{N_0} 1\,\mathrm{d}V = \int_{x_1=0}^{1}\int_{x_2=0}^{1-x_1}\int_{x_3=0}^{1-x_1-x_2}\cdots\int_{x_{n+1}=0}^{1-x_1-x_2-\cdots-x_n} 1\,\mathrm{d}x_{n+1}\cdots\mathrm{d}x_1 = \frac{1}{(n+1)!}.$$

Since N is obtained from N_0 by scaling the first n coordinates by 2, the volume of N is 2^n times the volume of N_0.

$$\lim_{m\to\infty}\frac{\sigma_m}{m^{n+2}} = \int_N 1\ \mathrm{d}V = 2^n\cdot\int_{N_0} 1\ \mathrm{d}V = \frac{2^n}{(n+1)!}.$$

Therefore, we conclude that

$$\sigma_m = \frac{2^n}{(n+1)!}m^{n+2} + o(m^{n+2}).$$

Computing σ_j. The term σ_j is the sum of the exponents of x_j over the shift polynomials. By Lemma 4,

$$\lim_{m\to\infty}\frac{\sigma_j}{m^{n+2}} = \int_N x_j\,\mathrm{d}V = \begin{cases} 2^{n+1}\int_{N_0} x_j\,\mathrm{d}V & \text{for } j=1,\ldots,n,\\ 2^n\int_{N_0} x_j\,\mathrm{d}V & \text{for } j=n+1.\end{cases}$$

By symmetry, each coordinate contributes equally, and hence

$$\int_{N_0} x_1\,\mathrm{d}V = \cdots = \int_{N_0} x_{n+1}\,\mathrm{d}V.$$

It follows that

$$\int_{N_0} x_j\,\mathrm{d}V = \frac{1}{n+1}\int_{N_0}\left(\sum_{k=1}^{n+1} x_k\right)\mathrm{d}V, \quad \text{for } j=1,\ldots,n+1.$$

Recall that the standard n-simplex is defined as

$$\Delta^n = \left\{(x_1,\ldots,x_{n+1})\in\mathbb{R}^{n+1} \mid x_i\geq 0,\ \sum_{i=1}^{n+1} x_i = 1\right\},$$

whose volume is $\int_{\Delta^n} 1\,\mathrm{d}V = 1/n!$. Using this identity, we get

$$\int_{N_0}\left(\sum x_k\right)\mathrm{d}V = \int_0^1 s\cdot\mathrm{Vol}(s\Delta^n)\,\mathrm{d}s = \int_0^1 s\cdot\frac{s^n}{n!}\,\mathrm{d}s = \frac{1}{(n+2)\cdot n!}.$$

Therefore, we have

$$\int_{N_0} x_j\,\mathrm{d}V = \frac{1}{n+1}\cdot\frac{1}{(n+2)\cdot n!} = \frac{1}{(n+2)!}.$$

Combining the above results, we conclude that

$$\sigma_1 = \sigma_2 = \cdots = \sigma_n = 2\sigma_{n+1} = \frac{2^{n+1}}{(n+2)!}m^{n+2} + o(m^{n+2}).$$

Computing σ_p. The term σ_p is the sum of the powers of p in the lattice construction. It can be computed as the difference between the volumes of two related polytopes. Following [9], define $\widetilde{A}$ by

$$\widetilde{A} = \{(i_1, \ldots, i_{n+1}, 1) \mid (i_1, \ldots, i_{n+1}) \in A\} \\ \cup \{(i_1, \ldots, i_{n+1}, 0) \mid (i_1, \ldots, i_{n+1}) \in B\} \subseteq \mathbb{Z}^{n+2},$$

and let $\widetilde{N}$ be the convex hull of $\widetilde{A}$. Hence, we have

$$\lim_{m\to\infty} \frac{\sigma_p}{m^{n+2}} = \int_N 1 \, \mathrm{d}V - \int_{\widetilde{N}} 1 \, \mathrm{d}V.$$

It remains to evaluate $\int_{\widetilde{N}} 1 \, \mathrm{d}V$. As before, introduce a simpler polytope $\widetilde{N}_0$, defined as the convex hull of the point set

$$\widetilde{A}_0 = \{(i_1, \ldots, i_{n+1}, 1) \mid (i_1, \ldots, i_{n+1}) \in \{\mathbf{0}^{(n+1)}, \alpha_1^{(n+1)}, \ldots, \alpha_{n+1}^{(n+1)}\}\} \\ \cup \{(i_1, \ldots, i_{n+1}, 0) \mid (i_1, \ldots, i_{n+1}) \in \{\alpha_1^{(n+1)}, \ldots, \alpha_{n-1}^{(n+1)}\}\} \subseteq \mathbb{Z}^{n+2}.$$

Accordingly, we have

$$\int_{\widetilde{N}} 1 \, \mathrm{d}V = 2^n \cdot \int_{\widetilde{N}_0} 1 \, \mathrm{d}V.$$

Directly computing the integral over $\widetilde{N}_0$ is inconvenient. We instead consider a larger, more symmetric polytope S, defined as the convex hull of $\widetilde{A}_0 \cup \{\mathbf{0}^{(n+2)}\} \subseteq \mathbb{Z}^{n+2}$.

A key observation is that, if T denotes the polytope generated by the points

$$\{\mathbf{0}^{(n+2)}, \alpha_1^{(n+2)}, \ldots, \alpha_{n-1}^{(n+2)}, \alpha_{n+2}^{(n+2)}, (0, \ldots, 0, 1, 0, 1), (0, \ldots, 0, 1, 1)\} \subseteq \mathbb{Z}^{n+2},$$

that is

$$\{(0, \ldots, 0), (1, 0, \ldots, 0), \ldots, (0, \ldots, 1, 0, 0, 0), \\ (0, \ldots, 0, 1), (0, \ldots, 0, 1, 0, 1), (0, \ldots, 0, 1, 1)\}.$$

then

$$\int_{\widetilde{N}_0} 1 \, \mathrm{d}V = \int_S 1 \, \mathrm{d}V - \int_T 1 \, \mathrm{d}V.$$

The volume of T is

$$\int_T 1 \, \mathrm{d}V = \frac{1}{(n+2)!}.$$

We now compute $\int_S 1 \, \mathrm{d}V$. The projection of S onto the first $n-1$ coordinates is generated by

$$\{\mathbf{0}^{(n-1)}, \alpha_1^{(n-1)}, \ldots, \alpha_{n-1}^{(n-1)}\} \subseteq \mathbb{Z}^{n-1},$$

whose volume is $1/(n-1)!$.

Now consider the projection of S onto the first $n-1$ coordinates together with the $(n+2)$-th coordinate. Define

$$\begin{aligned}\psi : \mathbb{Z}^{n+2} &\longrightarrow \mathbb{Z}^n,\\ (i_1,\ldots,i_{n+2}) &\longmapsto (i_1,\ldots,i_{n-1},i_{n+2}).\end{aligned}$$

Then $\psi(S)$ is an n-dimensional prism of height 1 whose base is the polytope generated by

$$\{\mathbf{0}^{(n-1)}, \alpha_1^{(n-1)}, \ldots, \alpha_{n-1}^{(n-1)}\} \subseteq \mathbb{Z}^{n-1}.$$

Thus $\psi(S)$ has volume $1/(n-1)!$, and

$$\int_S 1\,\mathrm{d}V = \int_{x_{n+1}=0}^{1}\int_{x_n=0}^{1-x_{n+1}} \frac{(1-x_n-x_{n+1})^n}{(n-1)!}\,\mathrm{d}x_n\,\mathrm{d}x_{n+1} = \frac{n}{(n+2)!}.$$

It follows that

$$\begin{aligned}\lim_{m\to\infty}\frac{\sigma_p}{m^{n+2}} &= \int_N 1\,\mathrm{d}V - \int_{\tilde{N}} 1\,\mathrm{d}V = \int_N 1\,\mathrm{d}V - 2^n\cdot\left(\int_S 1\,\mathrm{d}V - \int_T 1\,\mathrm{d}V\right)\\ &= \frac{2^n}{(n+1)!} - \frac{2^n\cdot(n-1)}{(n+2)!}\\ &= \frac{3\cdot 2^n}{(n+2)!}.\end{aligned}$$

Therefore, we conclude that

$$\sigma_p = \frac{3\cdot 2^n}{(n+2)!}m^{n+2} + o(m^{n+2}).$$

Substituting these values into (5) with $X_j = p^\delta$, taking logarithms base p, and keeping only the leading terms in m gives

$$\delta\left(n\cdot\frac{2^{n+1}}{(n+2)!} + \frac{2^n}{(n+2)!}\right) + \frac{3\cdot 2^n}{(n+2)!} < \frac{2^n}{(n+1)!} - \varepsilon_1.$$

Multiplying by $(n+2)!$ and dividing by 2^n simplifies this to

$$\delta\big(2n+1\big) + 3 < (n+2) - \varepsilon_2,$$

which gives

$$\delta < \frac{n-1}{2n+1} - \varepsilon.$$

The overall time complexity is dominated by LLL reduction and root extraction. For fixed m and n, it is polynomial in $\log p$. □

Remark 2. *In this paper, we only consider the asymptotic setting where $n \to \infty$. Therefore, we do not apply the unravelled linearization technique proposed in [13] and used in [1]. For fixed n, incorporating this technique may lead to improved bounds.*

Table 3: Experimental results of attacks on QCGs with known a and unknown b

π (bits)	n	κ (bits)	δ (expt.)	m	ω	Time (seconds)
256	2	203	0.207	2	19	0.396
256	3	190	0.258	2	33	0.593
256	4	184	0.281	2	51	1.883

Experimental Validation. We validate the attack experimentally for $\pi = \log_2(p) = 256$. The results are shown in Table 3.

Attacks on Pollard Generators A Pollard Generator is defined by the recurrence $v_{i+1} = v_i^2 + c \bmod p$. We consider the case where the parameter c is unknown. Given $n+1$ consecutive outputs, we have n relations $v_j = v_{j-1}^2 + c \bmod p$. Subtracting consecutive relations eliminates c:

$$v_j - v_{j+1} = v_{j-1}^2 - v_j^2 \bmod p,$$

which can be rearranged to

$$v_{j-1}^2 - v_j^2 - v_j + v_{j+1} \equiv 0 \bmod p.$$

This is the same relation as for QCGs with known $a = 1$ and unknown b. Substituting $v_k = w_k + u_k$ and letting $x_k = u_{k-1}$ gives a system of $n-1$ modular equations of the form

$$f_j(x_j, x_{j+1}, x_{j+2}) = x_j^2 + c_{j1}x_j + c_{j2}x_{j+1}^2 + c_{j3}x_{j+1} + c_{j4}x_{j+2} + c_{j0} \equiv 0 \bmod p.$$

This system has the same structure as the one in Section 3.1. Therefore, the same analysis applies, from which we derive the following result.

Theorem 2. *Let a Pollard generator be defined by $F(x) = x^2 + c$ over $\mathbb{Z}_p$ with unknown c. Given $n+1$ consecutive truncated outputs for fixed $n \geq 2$, the entire sequence can be predicted, provided that Assumption 1 holds and that the single continuous unknown part of size p^δ for each iteration satisfies*

$$\delta < \frac{n-1}{2n+1} - \varepsilon,$$

for some negligible $\varepsilon > 0$. The time complexity is polynomial in $\log p$. As $n \to \infty$, the asymptotic bound is $\delta < 1/2$.

Experimental Validation. We validate the attack experimentally for $\pi = \log_2(p) = 256$. The results are shown in Table 4.

Table 4: Experimental results of attacks on Pollard Generators with unknown c

π (bits)	n	κ (bits)	δ (expt.)	m	ω	Time (seconds)
256	2	202	0.211	3	44	0.714
256	3	187	0.270	3	96	5.118
256	4	182	0.289	3	180	42.076

3.2 The Case of Unknown a and Known b

We now consider the case where a is unknown and b is known in the QCG defined by $F(x) = ax^2 + b$ over $\mathbb{Z}_p$. We start from the equations

$$v_j = av_{j-1}^2 + b \bmod p, \qquad j = 1, \ldots, n.$$

We eliminate the unknown a by taking ratios: for $j = 1, \ldots, n-1$,

$$\frac{v_j - b}{v_{j+1} - b} = \frac{v_{j-1}^2}{v_j^2} \pmod p,$$

which can be rearranged as

$$v_j^2(v_j - b) - v_{j-1}^2(v_{j+1} - b) \equiv 0 \pmod p.$$

Substituting $v_k = w_k + u_k$ leads to

$$(w_j + u_j)^2(w_j + u_j - b) - (w_{j-1} + u_{j-1})^2(w_{j+1} + u_{j+1} - b) \equiv 0 \bmod p.$$

Letting $x_k = u_{k-1}$ for $k = 1, \ldots, n+1$, we obtain, for each $j = 1, \ldots, n-1$, a modular polynomial equation

$$f_j(x_j, x_{j+1}, x_{j+2}) \equiv 0 \pmod p.$$

The leading monomial is x_{j+1}^3, and the polynomial can be written as

$$\begin{aligned} f_j(x_j, x_{j+1}, x_{j+2}) = x_{j+1}^3 &+ c_{j1}x_j^2x_{j+2} + c_{j2}x_j^2 + c_{j3}x_{j+1}^2 + c_{j4}x_jx_{j+2} \\ &+ c_{j5}x_j + c_{j6}x_{j+1} + c_{j7}x_{j+2} + c_{j0} \equiv 0 \pmod p, \end{aligned}$$

where the coefficients c_{jk} are known. Thus we obtain a system of $n-1$ modular equations in $n+1$ variables:

$$\begin{cases} f_1(x_1, x_2, x_3) \equiv 0 \pmod p, \\ f_2(x_2, x_3, x_4) \equiv 0 \pmod p, \\ \qquad\vdots \\ f_{n-1}(x_{n-1}, x_n, x_{n+1}) \equiv 0 \pmod p. \end{cases}$$

The root of interest is

$$(x_1^\star, \ldots, x_{n+1}^\star) = (u_0, \ldots, u_n),$$

where each u_i denotes the unknown least significant part of v_i.

We apply the Basic Strategy. The analysis is analogous to that in Section 3.1, but the Newton polytopes now reflect the cubic leading terms. Unlike Theorem 1, where the relevant monomials have degree at most 2, the present polynomials contain cubic terms. We therefore define A using points scaled by a factor of 3 and add auxiliary points so that the resulting lattice remains diagonal.

Theorem 3. *Let a QCG be defined by $F(x) = ax^2 + b$ over $\mathbb{Z}_p$ with unknown a and known b. Given $n+1$ consecutive truncated outputs for fixed $n \geq 2$, the entire sequence can be predicted, provided that Assumption 1 holds and that the single continuous unknown part of size p^δ for each iteration satisfies*

$$\delta < \frac{n-1}{3n+3} - \varepsilon,$$

for some negligible ε. The time complexity is polynomial in $\log p$. As $n \to \infty$, the asymptotic bound is $\delta < 1/3$.

Proof. The leading monomial of each f_j in the graded reverse lexicographic order is x_{j+1}^3. Let m be a large positive integer, and define the same elementary vectors $\alpha_j^{(n+1)} \in \mathbb{Z}^{n+1}$ as (4). The sets A and B are defined as

$$A = \{\mathbf{0}^{(n+1)}, \alpha_1^{(n+1)}, \ldots, \alpha_{n+1}^{(n+1)}, 3\alpha_1^{(n+1)}, \ldots, 3\alpha_{n+1}^{(n+1)}\} \subseteq \mathbb{Z}^{n+1},$$
$$B = \{3\alpha_2^{(n+1)}, \ldots, 3\alpha_n^{(n+1)}\} \subseteq \mathbb{Z}^{n+1}.$$

We include $3\alpha_1^{(n+1)}$ and $3\alpha_{n+1}^{(n+1)}$ in A to simplify the volume calculations, since the polynomials f_j have degree 3 in all variables. As in the proof of Theorem 1, we use A and B to construct the shift polynomials $g_\lambda(x_1, \ldots, x_{n+1})$ for each monomial λ, and form the lattice Λ using the coefficient vectors of $g_\lambda(x_1X_1, \ldots, x_{n+1}X_{n+1})$ as its basis. The basis matrix of Λ is diagonal. Hence, the solvability condition from (1) becomes

$$\prod_{j=1}^{n+1} X_j^{\sigma_j} p^{\sigma_p} < p^{\sigma_m - \varepsilon_0}.$$

It remains to compute the values of σ_m, $\sigma_1, \ldots, \sigma_{n+1}$, and σ_p.

Computing σ_m. To evaluate σ_m, let N be the convex hull of A and apply Lemma 3:

$$\lim_{m\to\infty} \frac{\sigma_m}{m^{n+2}} = \lim_{m\to\infty} \frac{|mA|}{m^{n+1}} = \int_N 1 \,\mathrm{d}V.$$

Let N_0 denote the convex hull of $\{\mathbf{0}^{(n+1)}, \alpha_1^{(n+1)}, \ldots, \alpha_{n+1}^{(n+1)}\}$. It follows that

$$\int_N 1 \,\mathrm{d}V = 3^{n+1} \cdot \int_{N_0} 1 \,\mathrm{d}V = \frac{3^{n+1}}{(n+1)!}.$$

Therefore, we obtain

$$\sigma_m = \frac{3^{n+1}}{(n+1)!} m^{n+2} + o(m^{n+2}).$$

Computing σ_j. For each σ_j, Lemma 4 gives

$$\lim_{m\to\infty} \frac{\sigma_j}{m^{n+2}} = \int_N x_j \,\mathrm{d}V.$$

We have

$$\int_N x_j \,\mathrm{d}V = 3^{n+2} \cdot \int_{N_0} x_j \,\mathrm{d}V = \frac{3^{n+2}}{(n+2)!},$$

where N_0 is the convex hull of $\{\mathbf{0}^{(n+1)}, \alpha_1^{(n+1)}, \ldots, \alpha_{n+1}^{(n+1)}\}$ as before. Hence,

$$\sigma_1 = \cdots = \sigma_{n+1} = \frac{3^{n+2}}{(n+2)!} m^{n+2} + o(m^{n+2}).$$

Computing σ_p. To compute σ_p, define $\widetilde{A}$ as in the proof of Theorem 1:

$$\widetilde{A} = \{(i_1, \ldots, i_{n+1}, 1) \mid (i_1, \ldots, i_{n+1}) \in A\} \\ \cup \{(i_1, \ldots, i_{n+1}, 0) \mid (i_1, \ldots, i_{n+1}) \in B\} \subseteq \mathbb{Z}^{n+2}.$$

Let $\widetilde{N}$ be the convex hull of $\widetilde{A}$. Then, we have

$$\lim_{m\to\infty} \frac{\sigma_p}{m^{n+2}} = \int_N 1 \,\mathrm{d}V - \int_{\widetilde{N}} 1 \,\mathrm{d}V.$$

It remains to evaluate $\int_{\widetilde{N}} 1 \,\mathrm{d}V$. Introduce the simplified polytope $\widetilde{N}_0$, defined as the convex hull of the point set

$$\widetilde{A}_0 = \{(i_1, \ldots, i_{n+1}, 1) \mid (i_1, \ldots, i_{n+1}) \in \{\mathbf{0}^{(n+1)}, \alpha_1^{(n+1)}, \ldots, \alpha_{n+1}^{(n+1)}\}\} \\ \cup \{(i_1, \ldots, i_{n+1}, 0) \mid (i_1, \ldots, i_{n+1}) \in \{\alpha_2^{(n+1)}, \ldots, \alpha_n^{(n+1)}\}\} \subseteq \mathbb{Z}^{n+2}.$$

This gives

$$\int_{\widetilde{N}} 1 \,\mathrm{d}V = 3^{n+1} \cdot \int_{\widetilde{N}_0} 1 \,\mathrm{d}V.$$

Following the same idea as in Theorem 1, let S denote the convex hull of $\widetilde{A}_0 \cup \{\mathbf{0}^{(n+2)}\} \subseteq \mathbb{Z}^{n+2}$, and define T as the simplex generated by the points

$$\{\mathbf{0}^{(n+2)}, \alpha_2^{(n+2)}, \ldots, \alpha_n^{(n+2)}, \alpha_{n+2}^{(n+2)}, (1, 0, \ldots, 0, 1), (0, \ldots, 0, 1, 1)\} \subseteq \mathbb{Z}^{n+2},$$

which has volume $1/(n+2)!$. Then

$$\int_{\widetilde{N}_0} 1 \,\mathrm{d}V = \int_S 1 \,\mathrm{d}V - \int_T 1 \,\mathrm{d}V.$$

To compute $\int_S 1 \,\mathrm{d}V$, define the projection map ψ by taking the components from the second to the n-th together with the $(n+2)$-nd component:

$$\psi : \mathbb{Z}^{n+2} \longrightarrow \mathbb{Z}^n, \\ (i_1, \ldots, i_{n+2}) \longmapsto (i_2, \ldots, i_n, i_{n+2}).$$

Then $\psi(S)$ is an n-dimensional prism of height 1 whose base is the polytope generated by

$$\{\mathbf{0}^{(n-1)}, \alpha_1^{(n-1)}, \ldots, \alpha_{n-1}^{(n-1)}\} \subseteq \mathbb{Z}^{n-1}.$$

Thus $\psi(S)$ has volume $1/(n-1)!$, and

$$\int_S 1 \, \mathrm{d}V = \int_{x_{n+1}=0}^{1} \int_{x_1=0}^{1-x_{n+1}} \frac{(1-x_1-x_{n+1})^{n-1}}{(n-1)!} \, \mathrm{d}x_1 \, \mathrm{d}x_{n+1} = \frac{n}{(n+2)!}.$$

Combining the above results, we conclude that

$$\sigma_p = \frac{3^{n+2}}{(n+2)!} m^{n+2} + o(m^{n+2}).$$

Substituting these values into the solvability condition gives

$$\delta \cdot (n+1) \cdot \frac{3^{n+2}}{(n+2)!} + \frac{3^{n+2}}{(n+2)!} < \frac{3^{n+1}}{(n+1)!} - \varepsilon_1.$$

Multiplying by $(n+2)!$ and dividing by 3^{n+1} simplifies this to

$$\delta \cdot 3(n+1) + 3 < (n+2) - \varepsilon_2,$$

which yields the bound

$$\delta < \frac{n-1}{3n+3} - \varepsilon.$$

The complexity analysis is the same as in Theorem 1. Thus the time complexity is polynomial in $\log p$. □

Experimental Validation. We validate the attack experimentally for $\pi = \log_2(p) = 256$. The results are shown in Table 5.

Table 5: Experimental results of attacks on QCGs with unknown a and known b

π (bits)	n	κ (bits)	δ (expt.)	m	ω	Time (seconds)
256	2	234	0.086	2	36	0.507
256	3	230	0.102	2	82	3.401
256	4	228	0.109	2	148	17.372

4 Attacks on Power Generators

Let p be a prime number of bit-length π, and let $\mathbb{Z}_p$ denote the ring of integers modulo p. A Power Generator is defined by the recurrence $v_{i+1} = F(v_i) \bmod p$, where $F(x) = x^e$ is a polynomial of a fixed public exponent e over $\mathbb{Z}_p[x]$, and v_0

is the secret seed. We assume that the generator reveals the κ most significant bits of each state v_i with a known output pattern. Thus the internal state can be written as

$$v_i = w_i + u_i,$$

where w_i is the known partial output and $u_i \leq 2^{\pi-\kappa} - 1 = p^\delta$ is the unknown least significant part. Our goal is to recover u_i from infinitely many consecutive outputs while maximizing the admissible value of δ. It was shown in [14] that the attack succeeds if the unknown part is smaller than $p^{1/e}$, which corresponds to revealing a $(1 - 1/e)$ fraction of the MSBs.

In this section, we show that the same bound $1/e$ is achieved even when the Power Generator map $F(x) = x^e$ is perturbed by an unknown masking constant c. Consider the perturbed Power Generator defined by $v_j = v_{j-1}^e + c \bmod p$ for $j = 1, \ldots, n$, where c is unknown. Subtracting consecutive relations eliminates c:

$$v_j - v_{j+1} = v_{j-1}^e - v_j^e \bmod p.$$

This can be rewritten as

$$v_{j-1}^e - v_j^e - v_j + v_{j+1} \equiv 0 \bmod p.$$

Substituting $v_k = w_k + u_k$ and letting $x_k = u_{k-1}$ gives a system of $n - 1$ modular equations whose leading monomial is x_j^e. Hence

$$\begin{aligned} f_j(x_j, x_{j+1}, x_{j+2}) = x_j^e + c_{j,1}x_j^{e-1} + \cdots + c_{j,e-1}x_j \\ + c_{j,e}x_{j+1}^e + \cdots + c_{j,2e-1}x_{j+1} + c_{j,2e}x_{j+2} + c_{j,0} \equiv 0 \bmod p, \end{aligned}$$

where the coefficients $c_{j,k}$ are known explicitly. We therefore solve the modular equation system

$$\begin{cases} f_1(x_1, x_2, x_3) \equiv 0 \bmod p, \\ f_2(x_2, x_3, x_4) \equiv 0 \bmod p, \\ \qquad\qquad \vdots \\ f_{n-1}(x_{n-1}, x_n, x_{n+1}) \equiv 0 \bmod p. \end{cases}$$

The corresponding root is

$$(x_1^\star, x_2^\star, \ldots, x_{n+1}^\star) = (u_0, u_1, \ldots, u_n),$$

with upper bounds $X_j = p^\delta$. We apply the Basic Strategy together with the asymptotic computation framework. The resulting attack is as follows.

Theorem 4. *Let a perturbed Power Generator be defined by $F(x) = x^e + c$ over $\mathbb{Z}_p$ with unknown c. Given $n + 1$ consecutive truncated outputs for fixed $n \geq 2$, the entire sequence can be predicted, provided that Assumption 1 holds and that the single continuous unknown part of size p^δ for each iteration satisfies*

$$\delta < \frac{n-1}{en+1} - \varepsilon,$$

for some negligible $\varepsilon > 0$. The time complexity is polynomial in $\log p$. As $n \to \infty$, the asymptotic bound is $\delta < 1/e$.

Proof. The proof is analogous to that of Theorem 1. The leading monomial of each derived polynomial f_j is x_j^e. Let m be a large positive integer, and define the same elementary vectors $\alpha_j^{(n+1)} \in \mathbb{Z}^{n+1}$ as (4). The sets A and B are defined as

$$A = \{\mathbf{0}, \alpha_1^{(n+1)}, \ldots, \alpha_{n+1}^{(n+1)}, e\alpha_1^{(n+1)}, \ldots, e\alpha_n^{(n+1)}\} \subseteq \mathbb{Z}^{n+1},$$
$$B = \{e\alpha_1^{(n+1)}, \ldots, e\alpha_{n-1}^{(n+1)}\} \subseteq \mathbb{Z}^{n+1}.$$

As in the proof of Theorem 1, we use A and B to construct the shift polynomials $g_\lambda(x_1, \ldots, x_{n+1})$ for each monomial λ, and form the lattice Λ using the coefficient vectors of $g_\lambda(x_1X_1, \ldots, x_{n+1}X_{n+1})$ as its basis. The basis matrix of Λ is diagonal. Hence, the solvability condition from (1) becomes

$$\prod_{j=1}^{n+1} X_j^{\sigma_j} p^{\sigma_p} < p^{\sigma_m - \varepsilon_0}.$$

It remains to compute the values of σ_m, $\sigma_1, \ldots, \sigma_{n+1}$, and σ_p.

The only difference from Theorem 1 is that the vector is scaled by e instead of 2.

For the computation of σ_m and σ_j with $j = 1, \ldots, n+1$, we have

$$\sigma_m = \frac{e^n}{(n+1)!} m^{n+2} + o(m^{n+2}),$$
$$\sigma_1 = \sigma_2 = \cdots = \sigma_n = e\sigma_{n+1} = \frac{e^{n+1}}{(n+2)!} m^{n+2} + o(m^{n+2}).$$

For the computation of σ_p, we first define $\widetilde{A}$ as

$$\widetilde{A} = \{(i_1, \ldots, i_{n+1}, 1) \mid (i_1, \ldots, i_{n+1}) \in A\} \cup \{(i_1, \ldots, i_{n+1}, 0) \mid (i_1, \ldots, i_{n+1}) \in B\} \subseteq \mathbb{Z}^{n+2},$$

and let $\widetilde{N}$ be the convex hull of $\widetilde{A}$. Hence, we have

$$\lim_{m\to\infty} \frac{\sigma_p}{m^{n+2}} = \int_N 1 \, \mathrm{d}V - \int_{\widetilde{N}} 1 \, \mathrm{d}V.$$

Thus, we introduce a simpler polytope $\widetilde{N}_0$, defined as the convex hull of the point set

$$\widetilde{A}_0 = \{(i_1, \ldots, i_{n+1}, 1) \mid (i_1, \ldots, i_{n+1}) \in \{\mathbf{0}, \alpha_1^{(n+1)}, \ldots, \alpha_{n+1}^{(n+1)}\}\} \cup \{(i_1, \ldots, i_{n+1}, 0) \mid (i_1, \ldots, i_{n+1}) \in \{\alpha_1^{(n+1)}, \ldots, \alpha_{n-1}^{(n+1)}\}\} \subseteq \mathbb{Z}^{n+2}.$$

Accordingly, we have

$$\int_{\widetilde{N}} 1 \, \mathrm{d}V = e^n \cdot \int_{\widetilde{N}_0} 1 \, \mathrm{d}V.$$

The polytope $\widetilde{N}_0$ is the same as in the proof of Theorem 1. Therefore,

$$\sigma_p = \frac{3 \cdot e^n}{(n+2)!} m^{n+2} + o(m^{n+2}).$$

Substituting these values into the solvability condition with $X_j = p^\delta$ gives

$$\delta \left(n \cdot \frac{e^{n+1}}{(n+2)!} + \frac{e^n}{(n+2)!} \right) + \frac{3e^n}{(n+2)!} < \frac{e^n}{(n+1)!} - \varepsilon_1,$$

leading to the bound

$$\delta < \frac{n-1}{en+1} - \varepsilon.$$

The time complexity is polynomial in $\log p$. □

Experimental Validation. We validate the attack experimentally for $\pi = \log_2(p) = 256$. The results are shown in Table 6.

Table 6: Experimental results of attacks on perturbed Power Generators

π (bits)	e	n	κ (bits)	δ (expt.)	m	ω	Time (seconds)
256	3	2	230	0.102	2	30	0.323
256	3	3	224	0.125	2	57	2.374
256	3	4	221	0.137	2	93	6.624

5 Attacks on Linear Congruential Generators

Let p be a prime number of bit-length π. An LCG is defined by the recurrence

$$v_{i+1} = F(v_i) \bmod p,$$

where $F(x) = ax + b$ is a linear polynomial over $\mathbb{Z}_p$, and v_0 is the secret seed. We consider the setting where a is unknown and b is known.

We assume that the generator reveals κ most significant bits of each state v_i, with a known output pattern. Thus the internal state can be written as

$$v_i = w_i + u_i,$$

where w_i is known and $u_i \leq 2^{\pi-\kappa} - 1 = p^\delta$ is the unknown least significant part. Our goal is to recover the sequence u_i, assuming access to infinitely many consecutive outputs, while maximizing the admissible value of δ.

We use $n+1$ consecutive outputs, which yield n relations

$$v_j = av_{j-1} + b \bmod p, \qquad j = 1, \ldots, n.$$

We eliminate the unknown a by taking ratios: for $j = 1, \ldots, n-1$,

$$\frac{v_j - b}{v_{j+1} - b} = \frac{v_{j-1}}{v_j} \pmod{p},$$

which can be rearranged as

$$v_j(v_j - b) - v_{j-1}(v_{j+1} - b) \equiv 0 \pmod{p}.$$

Substituting $v_k = w_k + u_k$ leads to

$$(w_j + u_j)(w_j + u_j - b) - (w_{j-1} + u_{j-1})(w_{j+1} + u_{j+1} - b) \equiv 0 \bmod p.$$

Letting $x_k = u_{k-1}$ for $k = 1, \ldots, n+1$, we obtain, for each $j = 1, \ldots, n-1$, a modular polynomial equation

$$f_j(x_j, x_{j+1}, x_{j+2}) \equiv 0 \pmod{p}.$$

Under graded reverse lexicographic order, the leading monomial is x_{j+1}^2, and the polynomial can be written as

$$f_j(x_j, x_{j+1}, x_{j+2}) = x_{j+1}^2 + c_{j1}x_jx_{j+2} + c_{j2}x_j + c_{j3}x_{j+1} + c_{j4}x_{j+2} + c_{j0} \equiv 0 \pmod{p}$$

where the coefficients c_{jk} are known. This gives a system of $n-1$ modular equations in $n+1$ variables:

$$\begin{cases} f_1(x_1, x_2, x_3) \equiv 0 \pmod{p}, \\ f_2(x_2, x_3, x_4) \equiv 0 \pmod{p}, \\ \qquad \vdots \\ f_{n-1}(x_{n-1}, x_n, x_{n+1}) \equiv 0 \pmod{p}. \end{cases}$$

The root of interest is

$$(x_1^\star, \ldots, x_{n+1}^\star) = (u_0, \ldots, u_n),$$

where each u_i denotes the unknown least significant part of v_i.

We apply the Basic Strategy. The analysis is analogous to that in Section 3.1.

Theorem 5. *Let a LCG be defined by $F(x) = ax + b$ over $\mathbb{Z}_p$ with unknown a and known b. Given $n+1$ consecutive truncated outputs for fixed $n \geq 2$, the entire sequence can be predicted, provided that Assumption 1 holds and that the single continuous unknown part of size p^δ for each iteration satisfies*

$$\delta < \frac{n-1}{2n+2} - \varepsilon,$$

for some negligible $\varepsilon > 0$. The time complexity is polynomial in $\log p$. As $n \to \infty$, the asymptotic bound is $\delta < 1/2$.

Proof. The proof structure follows that of Theorem 3. The leading monomial of each f_j in the graded reverse lexicographic order is x_{j+1}^2. Let m be a large positive integer, and define the elementary vectors $\alpha_j^{(n+1)} \in \mathbb{Z}^{n+1}$ as in (4). The sets A and B are defined as

$$A = \{\mathbf{0}, \alpha_1^{(n+1)}, \ldots, \alpha_{n+1}^{(n+1)}, 2\alpha_1^{(n+1)}, \ldots, 2\alpha_{n+1}^{(n+1)}\} \subseteq \mathbb{Z}^{n+1},$$
$$B = \{2\alpha_2^{(n+1)}, \ldots, 2\alpha_n^{(n+1)}\} \subseteq \mathbb{Z}^{n+1}.$$

We use A and B to construct the shift polynomials $g_\lambda(x_1, \ldots, x_{n+1})$ for each monomial λ. After scaling by the bounds $X_1, \ldots, X_{n+1}$, their coefficient vectors generate Λ. Its basis matrix is diagonal. Hence, the solvability condition from (1) becomes

$$\prod_{j=1}^{n+1} X_j^{\sigma_j} p^{\sigma_p} < p^{\sigma_m - \varepsilon_0}.$$

It remains to compute the values of σ_m, $\sigma_1, \ldots, \sigma_{n+1}$, and σ_p.

For the computation of σ_m and σ_j with $j = 1, \ldots, n+1$, we have

$$\sigma_m = \frac{2^{n+1}}{(n+1)!} m^{n+2} + o(m^{n+2}),$$
$$\sigma_1 = \cdots = \sigma_{n+1} = \frac{2^{n+2}}{(n+2)!} m^{n+2} + o(m^{n+2}).$$

To compute σ_p, we define $\widetilde{A}$ similarly to the proof of Theorem 3 as follows:

$$\widetilde{A} = \{(i_1, \ldots, i_{n+1}, 1) \mid (i_1, \ldots, i_{n+1}) \in A\} \cup \{(i_1, \ldots, i_{n+1}, 0) \mid (i_1, \ldots, i_{n+1}) \in B\} \subseteq \mathbb{Z}^{n+2}.$$

Let $\widetilde{N}$ be the convex hull of $\widetilde{A}$. Then, we have

$$\lim_{m \to \infty} \frac{\sigma_p}{m^{n+2}} = \int_N 1 \, \mathrm{d}V - \int_{\widetilde{N}} 1 \, \mathrm{d}V.$$

It remains to evaluate $\int_{\widetilde{N}} 1 \, \mathrm{d}V$. Introduce the simplified polytope $\widetilde{N}_0$, defined as the convex hull of the point set

$$\widetilde{A}_0 = \{(i_1, \ldots, i_{n+1}, 1) \mid (i_1, \ldots, i_{n+1}) \in \{\mathbf{0}, \alpha_1^{(n+1)}, \ldots, \alpha_{n+1}^{(n+1)}\}\} \cup \{(i_1, \ldots, i_{n+1}, 0) \mid (i_1, \ldots, i_{n+1}) \in \{\alpha_2^{(n+1)}, \ldots, \alpha_n^{(n+1)}\}\} \subseteq \mathbb{Z}^{n+2}.$$

This gives

$$\int_{\widetilde{N}} 1 \, \mathrm{d}V = 2^{n+1} \cdot \int_{\widetilde{N}_0} 1 \, \mathrm{d}V.$$

Let S denote the convex hull of $\widetilde{A}_0 \cup \{\mathbf{0}\} \subseteq \mathbb{Z}^{n+2}$, and define T as the simplex generated by the points

$$\{\mathbf{0}, \alpha_2^{(n+2)}, \ldots, \alpha_n^{(n+2)}, \alpha_{n+2}^{(n+2)}, (1, 0, \ldots, 0, 1), (0, \ldots, 0, 1, 1)\} \subseteq \mathbb{Z}^{n+2},$$

which has volume $1/(n+2)!$. Then

$$\int_{\widetilde{N}_0} 1 \, \mathrm{d}V = \int_S 1 \, \mathrm{d}V - \int_T 1 \, \mathrm{d}V.$$

Following the same idea as in Theorem 3, we have

$$\int_S 1 \, \mathrm{d}V = \int_{x_{n+1}=0}^{1} \int_{x_1=0}^{1-x_{n+1}} \frac{(1-x_1-x_{n+1})^{n-1}}{(n-1)!} \, \mathrm{d}x_1 \, \mathrm{d}x_{n+1} = \frac{n}{(n+2)!}.$$

Thus, we conclude that

$$\sigma_p = \frac{3 \cdot 2^{n+1}}{(n+2)!} m^{n+2} + o(m^{n+2}).$$

Substituting these values into the solvability condition gives

$$\delta \cdot (n+1) \cdot \frac{2^{n+2}}{(n+2)!} + \frac{3 \cdot 2^{n+1}}{(n+2)!} < \frac{2^{n+1}}{(n+1)!} - \varepsilon_1,$$

which yields the bound

$$\delta < \frac{n-1}{2n+2} - \varepsilon.$$

The complexity analysis is the same as in Theorem 1. Thus the time complexity is polynomial in $\log p$. □

Experimental Validation. We validate the attack experimentally for $\pi = \log_2(p) = 256$. The results are shown in Table 7.

Table 7: Experimental results of attacks on LCGs with unknown a and known b

π (bits)	n	κ (bits)	δ (expt.)	m	ω	Time (seconds)
256	2	201	0.215	3	44	0.869
256	3	187	0.270	3	115	7.307
256	4	183	0.285	3	236	39.467

6 Conclusion

We presented new asymptotic attack bounds for several common polynomial congruential generators (PCGs), including QCGs, LCGs, Power Generators, and Pollard Generators. Our analyses cover several settings with partially known coefficients and refine the understanding of their security thresholds. Numerical experiments support the validity and effectiveness of the attacks.

We also developed a method for applying automated Coppersmith techniques to symbolic systems of equations. Specifically, we exploit the recursive algebraic structures that arise from $n+1$ consecutive generator outputs. By computing the associated high-dimensional polytope volumes symbolically in n, we derive general expressions for the attack bounds rather than only numerical estimates for fixed systems.

This symbolic analysis framework for structured polynomial systems may be of independent interest and may apply to other areas of public-key cryptanalysis where similar algebraic structures arise.

Acknowledgments

The author thanks the anonymous reviewers for their careful reading and helpful comments that improved both the clarity and quality of this paper. In particular, the author thank Reviewer C and Zhaopeng Ding for pointing out a typo in the integral calculation in Theorem 5 and also an earlier discussion with Yiming Gao. AI tools were used as typing assistants for grammar and basic editing. Yansong Feng and Yanbin Pan were supported by National KeyResearch and Development Project (No. 2025YFA1017200), National Natural Science Foundation of China (No. 62372445). Mengce Zheng was supproted by the National Natural Science Foundation of China (No. 62002335), and Ningbo Young Science and Technology Talent Cultivation Program (No. 2023QL007).

References

1. Bauer, A., Vergnaud, D., Zapalowicz, J.: Inferring sequences produced by nonlinear pseudorandom number generators using Coppersmith's methods. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7293, pp. 609–626. Springer (2012), https://doi.org/10.1007/978-3-642-30057-8_36
2. Becker, T., Weispfenning, V., Kredel, H.: Gröbner bases - a computational approach to commutative algebra, Graduate texts in mathematics, vol. 141. Springer (1993)
3. Blackburn, S.R., Gómez-Pérez, D., Gutierrez, J., Shparlinski, I.E.: Predicting the inversive generator. In: Paterson, K.G. (ed.) Cryptography and Coding, 9th IMA International Conference, Cirencester, UK, December 16-18, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2898, pp. 264–275. Springer (2003), https://doi.org/10.1007/978-3-540-40974-8_21
4. Blackburn, S.R., Gómez-Pérez, D., Gutierrez, J., Shparlinski, I.E.: Reconstructing noisy polynomial evaluation in residue rings. J. Algorithms 61(2), 47–59 (2006), https://doi.org/10.1016/j.jalgor.2004.07.002
5. Boyar, J.: Inferring sequences produced by pseudo-random number generators. J. ACM 36(1), 129–141 (1989), https://doi.org/10.1145/58562.59305

6. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. Cryptol. 10(4), 233–260 (1997), https://doi.org/10.1007/s001459900030
7. Ding, Z., Dai, Z., Wu, B., Zhang, Y., Zhang, K.: Computing asymptotic bounds for the automated coppersmith method via linear programming. Cryptology ePrint Archive (2026)
8. Feng, Y., Luo, H., Chen, Q., Nitaj, A., Pan, Y.: Computing asymptotic bounds for small roots in Coppersmith's method via sumset theory. Cryptology ePrint Archive, Paper 2024/1330 (2024), https://eprint.iacr.org/2024/1330
9. Feng, Y., Luo, H., Chen, Q., Nitaj, A., Pan, Y.: Computing asymptotic bounds for small roots in Coppersmith's method via sumset theory. In: Kalai, Y.T., Kamara, S.F. (eds.) Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part I. Lecture Notes in Computer Science, vol. 16000, pp. 3–32. Springer (2025), https://doi.org/10.1007/978-3-032-01855-7_1
10. Gao, J., Xu, J., Hu, L.: Inferring sequences produced by the quadratic generator. In: Deng, Y., Yung, M. (eds.) Information Security and Cryptology - 18th International Conference, Inscrypt 2022, Beijing, China, December 11-13, 2022, Revised Selected Papers. Lecture Notes in Computer Science, vol. 13837, pp. 483–494. Springer (2022), https://doi.org/10.1007/978-3-031-26553-2_26
11. Gómez, D., Gutierrez, J., Ibeas, Á.: Cryptanalysis of the quadratic generator. In: Maitra, S., Madhavan, C.E.V., Venkatesan, R. (eds.) Progress in Cryptology - INDOCRYPT 2005, 6th International Conference on Cryptology in India, Bangalore, India, December 10-12, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3797, pp. 118–129. Springer (2005), https://doi.org/10.1007/11596219_10
12. Gómez-Pérez, D., Gutierrez, J., Ibeas, Á.: Attacking the Pollard generator. IEEE Trans. Inf. Theory 52(12), 5518–5523 (2006), https://doi.org/10.1109/TIT.2006.885451
13. Herrmann, M., May, A.: Solving linear equations modulo divisors: On factoring given any bits. In: Pieprzyk, J. (ed.) Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5350, pp. 406–424. Springer (2008), https://doi.org/10.1007/978-3-540-89255-7_25
14. Herrmann, M., May, A.: Attacking power generators using unravelled linearization: When do we output too much? In: Matsui, M. (ed.) Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5912, pp. 487–504. Springer (2009), https://doi.org/10.1007/978-3-642-10366-7_29
15. Howgrave-Graham, N.: Finding small roots of univariate modular equations revisited. In: Darnell, M. (ed.) Cryptography and Coding, 6th IMA International Conference, Cirencester, UK, December 17-19, 1997, Proceedings. Lecture Notes in Computer Science, vol. 1355, pp. 131–142. Springer (1997), https://doi.org/10.1007/BFb0024458
16. Jochemsz, E., May, A.: A strategy for finding roots of multivariate polynomials with new applications in attacking RSA variants. In: Lai, X., Chen, K. (eds.) Advances in Cryptology - ASIACRYPT 2006, 12th International Conference on the Theory and Application of Cryptology and Information Security, Shanghai,

China, December 3-7, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4284, pp. 267–282. Springer (2006), https://doi.org/10.1007/11935230_18
17. Joux, A., Stern, J.: Lattice reduction: A toolbox for the cryptanalyst. J. Cryptol. 11(3), 161–185 (1998), https://doi.org/10.1007/s001459900042
18. Khovanskii, A.G.: Newton polyhedron, Hilbert polynomial, and sums of finite sets. Functional Analysis and Its Applications 26(4), 276–281 (1992)
19. Lenstra, A., Lenstra, H., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen 261(4), 515–534 (1982)
20. May, A.: New RSA vulnerabilities using lattice reduction methods. Ph.D. thesis, University of Paderborn (2003), https://ubdata.uni-paderborn.de/ediss/17/2003/may/disserta.pdf
21. May, A.: Lattice-based integer factorization - an introduction to Coppersmith's method. In: Bos, J.W., Stam, M. (eds.) Computational Cryptography: Algorithmic Aspects of Cryptology, pp. 78–105. London Mathematical Society Lecture Note Series, Cambridge University Press (2021), https://www.cits.ruhr-uni-bochum.de/imperia/md/content/may/paper/intro_to_coppersmiths_method.pdf
22. Meers, J., Nowakowski, J.: Solving the hidden number problem for CSIDH and CSURF via automated Coppersmith. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 14441, pp. 39–71. Springer (2023), https://doi.org/10.1007/978-981-99-8730-6_2
23. Plumstead, J.B.: Inferring a sequence generated by a linear congruence. In: 23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982. pp. 153–159. IEEE Computer Society (1982), https://doi.org/10.1109/SFCS.1982.73
24. Ryan, K.: Solving multivariate coppersmith problems with known moduli. In: Fehr, S., Fouque, P. (eds.) Advances in Cryptology - EUROCRYPT 2025 - 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4-8, 2025, Proceedings, Part VI. Lecture Notes in Computer Science, vol. 15606, pp. 355–384. Springer (2025), https://doi.org/10.1007/978-3-031-91095-1_13
25. Stern, J.: Secret linear congruential generators are not cryptographically secure. In: 28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987. pp. 421–426. IEEE Computer Society (1987), https://doi.org/10.1109/SFCS.1987.51
26. The Sage Developers: SageMath, the Sage Mathematics Software System (Version 10.3) (2025), https://www.sagemath.org
27. Zhang, R., Bi, J., Li, L., Peng, H.: Cryptanalysis on two kinds of number theoretic pseudo-random generators using Coppersmith method. IET Information Security 2025(1), 5569393 (2025)
28. Zheng, M., Kang, H.: Lattice-based cryptanalysis of RSA-type cryptosystems: a bibliometric analysis. Cybersecur. 7(1), 74 (2024), https://doi.org/10.1186/s42400-024-00289-7