An Improved Method for Predicting Truncated Fibonacci LFSRs over Integer Residue Rings

Han-Bing Yu[1], Qun-Xiong Zheng[1*], Wen-Feng Qi[1]

[1]Information Engineering University, P.O. Box 407, 62 Kexue Road, Zhengzhou, Henan, 450001, China.

*Corresponding author(s). E-mail(s): qunxiong_zheng@163.com;
Contributing authors: hbing_yu@163.com; wenfeng.qi@263.net;

Abstract

Sequences over the residue ring of integers modulo $\boldsymbol{m}$ generated by linear feedback shift registers (LFSRs) exhibit ring-level linearity and bit-level nonlinearity, making such kind of LFSRs (denoted as $\mathbb{Z}/(\boldsymbol{m})$-LFSRs) a key component of stream ciphers. Beyond fundamental cryptographic properties, the unpredictability of truncated $\mathbb{Z}/(\boldsymbol{m})$-LFSRs has attracted considerable attention as a critical security consideration in the design of stream cipher components. This paper investigates the unpredictability of truncated Fibonacci $\mathbb{Z}/(\boldsymbol{m})$-LFSRs under various scenarios. First, we provide a practical heuristic estimation for the values of two key parameters based on lattice theory, thereby avoiding previous blind search. This estimation is subsequently applied to determine the number of truncated digits required in different scenarios. Next, we develop a lattice-based method for finding annihilating polynomials over $\mathbb{Z}/(\boldsymbol{m})$ by the high-order truncated digits when the modulus $\boldsymbol{m}$ is known but coefficients are unknown, filling a methodological gap for this specific case. Finally, we demonstrate that when both the modulus $\boldsymbol{m}$ and coefficients are unknown but $\boldsymbol{m}$ is close to a power of 2, our lattice constructed by the high-order truncated digits can yield annihilating polynomials over $\mathbb{Z}/(\boldsymbol{m})$ rather than $\mathbb{Z}$ as in [1], achieving a 41% reduction in digits and a 4x speedup for the recovery of ZUC's driving sequence with 17 high-order truncated digits. Experimental results confirm the efficacy of our methods.

Keywords: Fibonacci $\mathbb{Z}/(\boldsymbol{m})$-LFSR, Truncated sequence, Lattice reduction, Unpredictability, Stream cipher

1 Introduction

Throughout the paper, for an integer $m > 1$, let $\mathbb{Z}/(m)$ denote the residue ring of integers modulo m. We choose $\{0, 1, \ldots, m-1\}$ as the set of representative elements of the ring $\mathbb{Z}/(m)$. Let log denote the base 2 logarithm. For any integer a and any positive integer $b \geq 2$, let $\lceil a \rfloor_{\bmod b}$ represent the absolutely smallest residue of $a \bmod b$, which takes values from the set $\{-\lfloor b/2 \rfloor, \ldots, -1, 0, 1, \ldots, \lceil b/2 \rceil - 1\}$. For any integer vector $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ and any positive integer $b \geq 2$, let $\lceil \mathbf{a} \rfloor_{\bmod b} = (\lceil a_0 \rfloor_{\bmod b}, \lceil a_1 \rfloor_{\bmod b}, \ldots, \lceil a_{n-1} \rfloor_{\bmod b})$.

The linear feedback shift register (LFSR) is a critical component in the design of stream ciphers and is widely employed in cryptographic systems such as A5 [2], E0 [3], ZUC [4], and the SNOW family [5, 6]. Early stream ciphers primarily utilized maximal-length LFSRs over the binary field $\mathbb{F}_2$ to generate pseudorandom sequences, known as m-sequences. These sequences possess favorable cryptographic properties, including a controllable period, balanced element distribution, and good run distribution [7]. However, their inherent linearity has made them vulnerable to a number of cryptanalytic attacks, such as correlation attacks [8] and algebraic attacks [9]. Consequently, designing nonlinear sequences with controllable cryptographic properties—such as period, element distribution, and run distribution—has become a key research challenge in the field of stream ciphers. In this context, linear recurring sequences generated by maximal-length LFSRs over integer residue rings (i.e., maximal-length $\mathbb{Z}/(m)$-LFSRs), commonly referred to as sequences over rings, have attracted significant interest. These sequences maintain linearity at the ring level, thereby preserving control over period and distribution, while also exhibiting rich nonlinear structures at the bit level due to the carry effect [4]. As a result, they have emerged as an important class of driving sequences in stream cipher research and have been adopted in the design of ZUC, a stream cipher which serves as the core of the 3GPP mobile standards 128-EEA3 (for encryption) and 128-EIA3 (for message integrity).

Fibonacci $\mathbb{Z}/(m)$-LFSRs represent the predominant architecture for generating sequences over rings. A Fibonacci $\mathbb{Z}/(m)$-LFSR of order n consists of n registers that shift their values with each clock interval and a linear recurrence function that feeds new values into the last register, as shown in Figure 1.

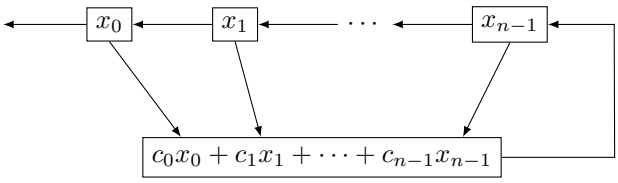

Fig. 1: A Fibonacci $\mathbb{Z}/(m)$-LFSR of order n

A Fibonacci $\mathbb{Z}/(m)$-LFSR of order n generates a sequence $\underline{a} = (a_i)_{i \geq 0}$ satisfying the linear recurrence relation

$$a_{i+n} \equiv c_{n-1}a_{i+n-1} + c_{n-2}a_{i+n-2} + \cdots + c_0 a_i \bmod m, i = 0, 1, 2, \ldots,$$

where the coefficients $c_0, c_1, \ldots, c_{n-1} \in \mathbb{Z}/(m)$ are not all zero. The initial state can be any n integers $a_0, a_1, \ldots, a_{n-1} \in \mathbb{Z}/(m)$, subject to the condition that for every prime divisor p_i of m, not all a_j are congruent to 0 modulo p_i. If the modulus m is a prime number, denoted as $m = p$, the maximum period of the output sequences of a Fibonacci $\mathbb{Z}/(m)$-LFSR of order n is $p^n - 1$ [10]. For a general modulus m, let $m = p_1^{e_1} p_2^{e_2} \cdots p_s^{e_s}$ be the canonical factorization of m, then the maximum period of the output sequences of a Fibonacci $\mathbb{Z}/(m)$-LFSR of order n is $\operatorname{lcm}\left(p_1^{e_1-1}\left(p_1^n-1\right), p_2^{e_2-1}\left(p_2^n-1\right), \ldots, p_s^{e_s-1}\left(p_s^n-1\right)\right)$. A Fibonacci $\mathbb{Z}/(m)$-LFSR reaches the maximum period if its characteristic polynomial

$$f(x) = x^n - c_{n-1}x^{n-1} - \cdots - c_0$$

is a primitive polynomial over $\mathbb{Z}/(m)$ [11, 12] (the definition of primitive polynomials over $\mathbb{Z}/(m)$ can be found in Definition 3). The aforementioned ZUC employs a Fibonacci $\mathbb{Z}/(m)$-LFSR of order 16 whose characteristic polynomial is a primitive polynomial over $\mathbb{Z}/(2^{31}-1)$, defined as

$$f(x) = x^{16} - (2^{15}x^{15} + 2^{17}x^{13} + 2^{21}x^{10} + 2^{20}x^4 + 2^8 + 1).$$

The $\mathbb{Z}/(m)$-LFSR, as an important component of stream ciphers, has been extensively studied with regard to the properties of its output sequences, including period, element distribution, run distribution, linear complexity, and unpredictability [1, 13–30]. Here, unpredictability refers to the property that a cryptanalyst is unable to predict with advantage any other segment of the $\mathbb{Z}/(m)$-LFSR's output within feasible time and memory complexity bounds, even after obtaining a sufficiently long segment of its output. The Berlekamp-Massey algorithm [13, 14] and its variants [15–17] demonstrate that when a sufficiently long segment of the full-bit output sequence is known, the entire sequence can be successfully predicted. This implies that outputting all bits of the sequence is no longer secure. Therefore, a natural question arises: if only partial bits of the sequence are output, can the truncated $\mathbb{Z}/(m)$-LFSR be considered unpredictable? More precisely, from the perspective of cryptanalysis, for a fixed number of truncated bits, what is the maximum output length that can be exposed without revealing the entire sequence? These problems essentially call for a criterion to evaluate the bit security of sequences over rings, which have been partially investigated in [1, 23–30].

For truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs, when only outputting the highest bit sequence, Huang et al. [23] and Kuzmin et al. [24] independently demonstrated that these sequences are pairwise distinct if the modulus is 2^e and the characteristic polynomial is primitive. This implies that the original primitive sequences can theoretically be reconstructed from their highest bits, provided the modulus and characteristic polynomial are known. Based on this theoretical groundwork, numerous practical prediction algorithms have been proposed [23, 25–27]. On the other hand, researchers have explored whether effective prediction remains feasible when multiple bits are truncated. For the case where all parameters (the modulus m and the coefficients $c_0, c_1, \ldots, c_{n-1}$) are known, Yang et al. [28] gave a method based on Coppersmith's method [31] to recover the initial state by the truncated sequences. For the case where

all parameters are unknown, Sun et al. [29] extended Stern's algorithm to predict truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs. Yu et al. [1] further significantly improved the method presented in [29]. They first apply the resultant to recover the modulus m, then use the Chinese Remainder Theorem and the idea of recovering p-adic coordinates of the coefficients layer by layer to recover the coefficients $c_0, c_1, \ldots, c_{n-1}$, and finally employ Kannan's embedding technique to recover the initial state $a_0, a_1, \ldots, a_{n-1}$. Galois $\mathbb{Z}/(m)$-LFSRs constitute another major class of $\mathbb{Z}/(m)$-LFSRs, and their unpredictability has been studied in [30].

As highlighted in both [1] and [30], a valid heuristic analysis for choosing the critical parameters r and t—which are essential for recovering the modulus and coefficients—is still lacking. In practice, identifying correct values for r and t relied on a computationally expensive trial-and-error process. Beyond this issue, an important scenario of truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs remains unexplored: the case where the modulus of the component is known but the specific tap positions are unknown.

To address these challenges and limitations, this paper focuses on Fibonacci $\mathbb{Z}/(m)$-LFSRs to investigate the aforementioned issues (which can also be extended to Galois $\mathbb{Z}/(m)$-LFSRs according to [30]). First, we provide a practical heuristic estimation for the values of r and t based on lattice theory, thereby avoiding previous blind search. This estimation is subsequently applied to determine the number of truncated digits required in different scenarios. Next, we develop a lattice-based method for finding annihilating polynomials over $\mathbb{Z}/(m)$ by the high-order truncated digits when the modulus m is known but coefficients are unknown, filling a methodological gap for this specific case. Finally, we demonstrate that when both the modulus m and coefficients are unknown but m is close to a power of 2, our lattice constructed by the high-order truncated digits can yield annihilating polynomials over $\mathbb{Z}/(m)$ rather than $\mathbb{Z}$ as in [1], achieving significant reductions in runtime and truncated digits. Extensive experiments have verified the correctness and effectiveness of the above methods. To be more specific,

- **We provide a practical heuristic estimation for the values of r and t.** When predicting a truncated $\mathbb{Z}/(m)$-LFSR, the selection of parameters r and t is critical for two main reasons. On the one hand, it determines the construction of the annihilating polynomials of the sequence, which directly affects whether the modulus and coefficients can be successfully recovered. On the other hand, it explicitly determines the number of truncated digits required, namely $r + t - 1$. Therefore, a practical estimation can significantly reduce the time consumed by trial and error. In prior works, only [29] provided a rough criterion for selecting these values, but it deviates considerably from their experimental results and is impractical for real experiments. The core of our approach involves constructing a lattice formed by the coefficient vectors of the annihilating polynomials of the sequence over $\mathbb{Z}/(m)$. By applying the Gaussian heuristic to estimate the length of the shortest nonzero vector in this lattice, we derive a minimal possible length for the coefficient vector of the annihilating polynomial. Next, under various truncation and parameter scenarios, we use this length to estimate the target vector length, thereby yielding a lower bound for the values of r and t and avoiding previous blind search. Experiments confirm that the theoretical estimation is attainable for some small-scale instances.

For large-scale instances such as ZUC's driving sequence, the actual values of r and t approach the estimation as more powerful SVP algorithms are employed. Although practical constraints of time and memory currently prevent these large instances from fully reaching the estimation, the observed trend provides strong evidence for the quality of this estimation. To be more specific, the detailed experimental results are as follows.

- For the Fibonacci $\mathbb{Z}/(2^{13}-1)$-LFSR of order 8 with 9 high-order truncated digits, annihilating polynomials are found exactly at the theoretical estimation $r = t = 24$ in 2 out of 30,000 instances, demonstrating the theoretical estimation is practically achievable for some small-scale instances. Moreover, 98.63% of all instances require values of r and t no larger than 27, which are in close proximity to the estimated values of 24.
- For ZUC's driving sequence with 17 high-order truncated digits, the minimal values of r and t decrease progressively with more powerful SVP algorithms: from 71 (BKZ-20) to 69 and 68 (BKZ-30 and BKZ-32) and finally to 64 (BGJ1 sieve). Subject to time and memory constraints, a slight discrepancy exists between the theoretical estimation ($r = t = 59$) and the best experimental result ($r = t = 64$); nevertheless, the descending trend demonstrates both the soundness of the estimation and its practical value for experimental guidance.

- **We propose a method for predicting high-order truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs when the modulus m is known but coefficients are unknown.**
  In previous work, such as [29] and [1], the scenario where both the modulus m and the coefficients $c_0, c_1, \ldots, c_{n-1}$ are unknown was considered, but no dedicated method was proposed for the scenario where the modulus m is known while $c_0, c_1, \ldots, c_{n-1}$ remain unknown. Although the methods in [29] and [1] can be directly applied to the latter scenario, this approach of not fully leveraging the known modulus is clearly not reasonable. A natural question then arises: If the modulus m is utilized, can the prediction be achieved with fewer truncated digits and shorter runtime? To address this issue, we propose an approach specifically designed for this scenario. The core of this approach lies in finding a sufficient number of annihilating polynomials over $\mathbb{Z}/(m)$. Once these polynomials are obtained, the coefficients and initial state can be recovered using the greatest common divisor method and lattice-based approach from [1], respectively. To find the annihilating polynomials, we use m, the segments of the high-order truncated sequences, and the bit-length β of the unknown low-order part to construct a lattice, whose reduced basis is shown to yield numerous annihilating polynomials over $\mathbb{Z}/(m)$. By combining the estimated target vector length with the Gaussian heuristic for this lattice, we derive a lower bound on the required number of truncated digits that is significantly more practical than the one in [29]. Experiments further confirm that our approach has advantages in both truncated digits and runtime, and the provided lower bound is reliable. The detailed experimental results are as follows.
  - For ZUC's driving sequence, assuming the modulus m is known, while the coefficients and the initial state are unknown, we use 141 truncated digits of 17

high-order bits to find 71 annihilating polynomials over $\mathbb{Z}/(m)$. This reduces the number of required truncated digits by 98 compared to the method in [1]. (Note that the method in [1] is designed for predicting $\mathbb{Z}/(m)$-LFSRs with unknown modulus.) Moreover, as demonstrated by the experiments on parameters r and t, at least 127 truncated digits are required in this experiment. Although the estimated lower bound of 117 shows a slight discrepancy from the actual requirement, it remains reliable compared to the impractical estimation provided in [29].
  - For $\mathbb{Z}/(m)$-LFSRs with a general modulus, experiments show that recovering the coefficients demands significantly more truncated digits and longer runtime when the modulus m is unknown compared to when it is known, thus justifying our focused investigation on the known modulus scenario. For details, see Section 5.

- **We present a method for predicting high-order truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs when all parameters are unknown, but m is close to a power of 2.**
  When predicting truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs with unknown parameters, the annihilating polynomials found by [1] are over $\mathbb{Z}$ due to the unknown modulus m. We show, however, that if $2^k - m < 2^\beta$ or $m - 2^{k-1} < 2^\beta$, where $k = \lceil \log m \rceil$ and β denotes the bit-length of the unknown low-order part, then annihilating polynomials over $\mathbb{Z}/(m)$ (as opposed to $\mathbb{Z}$) can still be found by a lattice-based method, even without precise knowledge of the value of m. Once these polynomials are obtained, the modulus m can be recovered using the resultant as described in [1], after which the coefficients and the initial state can be determined by the aforementioned method. To find the annihilating polynomials, we use the segments of the high-order truncated sequences and the bit-length α of the known high-order part to construct a lattice, whose reduced basis contains multiple annihilating polynomials over $\mathbb{Z}/(m)$. Additionally, we also derive a lower bound on the required number of truncated digits whose reliability is confirmed by experimental results. To evaluate the effectiveness of our proposed approach against the method in [1], we conduct comparative experiments as follows.
  - For ZUC's driving sequence, assuming all parameters are unknown, we recover them by only 141 truncated digits of 17 high-order bits in 0.92 min, compared to 239 digits and 3.83 min in [1]. This represents a 41% reduction in digits and a 4x speedup.
  - For $\mathbb{Z}/(m)$-LFSRs with a modulus m approximating a power of 2, our approach yields a substantial reduction in both runtime and the number of truncated digits required compared to the method in [1]. For details, see Section 5.

  Experiments demonstrate that, compared to [1], our approach requires fewer truncated digits and yields lower-degree annihilating polynomials, thereby significantly accelerating the recovery of m and $c_0, c_1, \ldots, c_{n-1}$.

The rest of the paper is organized as follows. Section 2 introduces the fundamental definitions and results used in subsequent sections. Section 3 first proposes a method for predicting truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs with a known modulus but unknown coefficients, and subsequently refines the approach established in [1] for the case where

the modulus and coefficients are unknown, but the modulus is near a power of 2. A key contribution here is a practical heuristic estimation for selecting r and t. Section 4 provides experimental validation of our methods' effectiveness. Section 5 concludes the paper and outlines an open problem for further investigation.

2 Preliminaries

In this section, we will introduce some basic definitions and results about sequences over rings, lattice reduction, and the resultant.

2.1 Characteristic Polynomials, minimal polynomials, and primitive polynomials

Definition 1 Let m be an integer greater than 1. If a sequence $\underline{a} = (a_i)_{i\geq 0}$ over $\mathbb{Z}/(m)$ satisfies the linear recurrence relation

$$a_{i+n} \equiv c_{n-1}a_{i+n-1} + c_{n-2}a_{i+n-2} + \cdots + c_0 a_i \bmod m, i = 0, 1, 2, \ldots,$$

where $c_0, c_1, \ldots, c_{n-1} \in \mathbb{Z}/(m)$ are not all zero and n is a positive integer, then the sequence $\underline{a}$ is called a linear recurring sequence of order n over $\mathbb{Z}/(m)$ generated by $f(x) = x^n - c_{n-1}x^{n-1} - \cdots - c_0$, and $f(x)$ is called a characteristic polynomial of $\underline{a}$. A characteristic polynomial of $\underline{a}$ with the minimal degree is called a minimal polynomial of $\underline{a}$. A polynomial $g(x) = d_s x^s + d_{s-1}x^{s-1} + \cdots + d_1 x + d_0$ over $\mathbb{Z}/(m)$ is called an annihilating polynomial of $\underline{a}$ if

$$d_s a_{i+s} + d_{s-1}a_{i+s-1} + \cdots + d_1 a_{i+1} + d_0 a_i \equiv 0 \bmod m, i = 0, 1, 2, \ldots.$$

Let $G(f(x), m)$ denote the set of all linear recurring sequences generated by $f(x)$ over $\mathbb{Z}/(m)$, and let $I_m(\underline{a})$ denote the set of annihilating polynomials of $\underline{a}$ over $\mathbb{Z}/(m)$. It can be easily verified that $I_m(\underline{a})$ forms an ideal of $\mathbb{Z}/(m)[x]$.

For any integer a and any positive integer $b \geq 2$, the least nonnegative residue of a modulo b is denoted by $a \bmod b$ or $[a] \bmod b$. For a sequence $\underline{a} = (a_i)_{i\geq 0}$ over $\mathbb{Z}/(m)$, let $\underline{a} \bmod b = ([a_i] \bmod b)_{i\geq 0}$, where $[a_i] \bmod b$ denotes a_i first modulo m then modulo b. Similarly, for a polynomial $f(x) = c_n x^n + \cdots + c_1 x + c_0$ over $\mathbb{Z}/(m)$, let $f(x) \bmod b = [c_n] \bmod b\, x^n + \cdots + [c_1] \bmod b\, x + [c_0] \bmod b$.

Let $f(x)$ be a monic polynomial of degree n over $\mathbb{Z}/(m)$. If $\gcd(f(0), m) = 1$, there exists a positive integer T such that $x^T - 1$ is divisible by $f(x)$ in $\mathbb{Z}/(m)[x]$. The minimum such T is called the period of $f(x)$ over $\mathbb{Z}/(m)$ and denoted by $\mathrm{per}(f(x), m)$.

Definition 2 [32] Let p^e be a prime power and let $f(x)$ be a monic polynomial of degree n over $\mathbb{Z}/(p^e)$ with $\gcd(f(0), p) = 1$. If $\mathrm{per}(f(x), p^e) = p^{e-1}(p^n - 1)$, then $f(x)$ is called a primitive polynomial of degree n over $\mathbb{Z}/(p^e)$. A sequence $\underline{a}$ is called a primitive sequence of order n over $\mathbb{Z}/(p^e)$ if $\underline{a}$ is generated by a primitive polynomial of degree n over $\mathbb{Z}/(p^e)$ and $\underline{a} \bmod p \neq \underline{0}$, where $\underline{0}$ is the all-zero sequence.

If $f(x)$ is a primitive polynomial of degree n over $\mathbb{Z}/(p^e)$, then for every $i \in \{1, 2, \ldots, e-1\}$, $f(x) \bmod p^i$ is also a primitive polynomial of degree n over $\mathbb{Z}/(p^i)$. In particular, $f(x) \bmod p$ is a primitive polynomial of degree n over the finite field $\mathbb{F}_p$.

Definition 3 [12] Let $m = p_1^{e_1} p_2^{e_2} \cdots p_s^{e_s}$ be the canonical factorization of m. A monic polynomial $f(x)$ with $\gcd(f(0), m) = 1$ is called a primitive polynomial of degree n over $\mathbb{Z}/(m)$ if for every $i \in \{1, 2, \ldots, s\}$, $f(x) \bmod p_i^{e_i}$ is a primitive polynomial of degree n over $\mathbb{Z}/(p_i^{e_i})$. A sequence $\underline{a}$ is called a primitive sequence of order n over $\mathbb{Z}/(m)$ if $\underline{a}$ is generated by a primitive polynomial of degree n over $\mathbb{Z}/(m)$ and $\underline{a} \bmod p_i \neq \underline{0}$ for every $i \in \{1, 2, \ldots, s\}$, i.e., $\underline{a} \bmod p_i^{e_i}$ is a primitive sequence of order n over $\mathbb{Z}/(p_i^{e_i})$.

Let $G'(f(x), m)$ denote the set of all primitive sequences generated by the primitive polynomial $f(x)$ over $\mathbb{Z}/(m)$.

2.2 Lattices and lattice reduction algorithms

In this subsection, we will introduce some results about lattices and lattice reduction algorithms.

Definition 4 A lattice is a discrete additive subgroup of $\mathbb{R}^n$. Specifically, given s linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_s \in \mathbb{R}^n$, the set

$$L = \{\mathbf{v} \mid \mathbf{v} = c_1\mathbf{b}_1 + c_2\mathbf{b}_2 + \cdots + c_s\mathbf{b}_s, c_1, c_2, \ldots, c_s \in \mathbb{Z}\}$$

is called an n-dimensional lattice. The set of linearly independent vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_s$ is called a basis of the lattice L and the integer s is called the rank of L. A lattice L is called a full-rank lattice if $s = n$.

Let B be the $s \times n$ matrix consisting of row vectors $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_s$. The determinant of the lattice L is defined by

$$\det(L) = \sqrt{\det(BB^T)},$$

where B^T is the transpose of B. In the subsequent sections, without causing confusion, we will use the matrix composed of lattice basis vectors to represent the corresponding lattice. The ith successive minima of L denoted by $\lambda_i(L)$ is the radius of the smallest zero-centered ball containing at least i linearly independent vectors of L. In particular, $\lambda_1(L)$ is the length of the shortest nonzero vector in L.

Definition 5 (Gaussian Heuristic) [33] Let L be a full-rank lattice in $\mathbb{R}^n$, and C a measurable subset of $\mathbb{R}^n$ with volume $\mathrm{vol}(C)$. The Gaussian Heuristic "predicts" that the number of points of $L \cap C$ is roughly $\mathrm{vol}(C)/\det(L)$.

A lattice is called a random lattice if it follows from Haar measures of classical groups [34], which give rise to a natural probability distribution on the set of lattices. Nguyen [35] suggested that random lattices have the following property: with overwhelming probability, the successive minima of a random n-dimensional lattice L are all asymptotically close to the Gaussian heuristic. In particular, $\lambda_1(L)$ can be estimated as follows.

Lemma 1 [36] *Let L be an n-dimensional lattice. If the Gaussian heuristic holds for L, then*

$$\lambda_1(L) \approx \sqrt{\frac{n}{2\pi e}} (\det L)^{1/n}.$$

For a vector $\mathbf{v}$, let $\|\mathbf{v}\|$ denote its Euclidean norm. Next, we introduce two basic computational lattice problems which are very useful in cryptography: the shortest vector problem (SVP) and the closest vector problem (CVP).

Definition 6 (SVP) Given a basis of an n-dimensional lattice L, find a $\mathbf{v} \in L$ such that $\|\mathbf{v}\| = \lambda_1(L)$.

Definition 7 (CVP) Given a basis of an n-dimensional lattice L and a point $\mathbf{t} \in \mathbb{R}^n$, find a $\mathbf{v} \in L$ such that $\|\mathbf{v} - \mathbf{t}\| \leq \|\mathbf{u} - \mathbf{t}\|$ for any vector $\mathbf{u} \in L$.

The main algorithms for solving SVPs can be divided into two types: approximation algorithms, such as the LLL algorithm [37] and the BKZ algorithm [38], and exact algorithms, such as Kannan's enumeration [39, 40] and the sieve of AKS [41]. In [37], Lenstra et al. put forward the famous LLL algorithm, which can solve an approximate-SVP in polynomial time. In [38], Schnorr and Euchner proposed the BKZ algorithm, which is a blockwise generalization of LLL with potentially superexponential complexity and better practical performance.

2.3 The resultant

The resultant is an important tool in algebra, widely used in determining whether two polynomials share common factors. It is defined as follows.

Definition 8 [42] Let R be an arbitrary commutative ring and $R[x]$ the polynomial ring over R. Let $g(x) = a_u x^u + a_{u-1} x^{u-1} + \cdots + a_0 \in R[x]$ and $h(x) = b_v x^v + b_{v-1} x^{v-1} + \cdots + b_0 \in R[x]$ be two polynomials of formal degrees u resp. v with $u, v \in \mathbb{N}$. Then the resultant $R(g, h)$ of the two polynomials is defined by the determinant

$$R(g,h) = \begin{vmatrix} a_u & a_{u-1} & \cdots & a_0 & 0 & & \cdots & 0 \\ 0 & a_u & a_{u-1} & \cdots & a_0 & 0 & \cdots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & \cdots & 0 & a_u & a_{u-1} & & \cdots & a_0 \\ b_v & b_{v-1} & \cdots & & b_0 & 0 & \cdots & 0 \\ 0 & b_v & b_{v-1} & \cdots & & b_0 & \cdots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & \cdots & 0 & b_v & b_{v-1} & & \cdots & b_0 \end{vmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\} v \\ \left.\vphantom{\begin{matrix}a\\a\\a\\a\end{matrix}}\right\} u \end{matrix}$$

of order $u + v$.

The following is an important theorem about the resultant, which will be used later to recover the modulus.

Theorem 1 [1] *Let m be an integer greater than 1 and $m = p_1^{e_1} p_2^{e_2} \cdots p_s^{e_s}$ the canonical factorization of m. Let $\underline{a}$ be a linear recurring sequence over $\mathbb{Z}/(m)$, and let $f(x)$ be a minimal polynomial of $\underline{a}$ over $\mathbb{Z}/(m)$ of degree n. If for every $i \in \{1, 2, \ldots, s\}$, $f(x) \bmod p_i$ is the minimal polynomial of $\underline{a} \bmod p_i$ over $\mathbb{F}_{p_i}$, then*

$$m^n \mid R\left(F^{(1)}(x), F^{(2)}(x)\right)$$

over $\mathbb{Z}$ for any two annihilating polynomials $F^{(1)}(x), F^{(2)}(x)$ of $\underline{a}$.

3 Main results

This section first introduces a prediction method for truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs with a known modulus but unknown coefficients, enabling the recovery of both coefficients and the initial state by the high-order truncated digits. Subsequently, we refine the method of [1] for the case where the modulus and coefficients are unknown, but the modulus is near a power of 2, presenting a more efficient and data-saving algorithm to recover all parameters—the modulus, coefficients, and initial state—by the high-order truncated digits.

First, we introduce some necessary notations and the problem to be studied.

Let m be an integer greater than 1 and let $f(x) = x^n - c_{n-1}x^{n-1} - \cdots - c_0 \in \mathbb{Z}/(m)[x]$. Let $\underline{a} = (a_i)_{i \geq 0} \in G(f(x), m)$, i.e., the sequence $\underline{a}$ satisfies

$$a_{i+n} \equiv c_{n-1}a_{i+n-1} + c_{n-2}a_{i+n-2} + \cdots + c_0 a_i \bmod m, i = 0, 1, 2, \ldots.$$

Let $k = \lceil \log m \rceil$ denote the bit-length of $m-1$. Assume that y_i consists of the α high-order bits of a_i and z_i consists of the β low-order bits of a_i, then a_i can be expressed as

$$a_i = 2^\beta y_i + z_i, i = 0, 1, 2, ..., \tag{1}$$

where $0 < \alpha < k$ and $\beta = k - \alpha$.

The problem of predicting truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs with a known modulus but unknown coefficients can be formulated as follows: given the modulus m and the first N truncated digits $y_0, y_1, \ldots, y_{N-1}$, how to recover the unknown coefficients $c_0, c_1, \ldots, c_{n-1}$ and the unknown initial state $a_0, a_1, \ldots, a_{n-1}$. The prediction algorithm is detailed in Subsection 3.1.

Similarly, the problem of predicting truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs with unknown parameters can be stated as follows: given the first N truncated digits $y_0, y_1, \ldots, y_{N-1}$, how to recover the unknown modulus m, the unknown coefficients $c_0, c_1, \ldots, c_{n-1}$, and the unknown initial state $a_0, a_1, \ldots, a_{n-1}$. The detailed prediction algorithm is provided in Subsection 3.2.

3.1 Predicting truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs with a known modulus but unknown coefficients

The method in [1] predicts truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs with unknown parameters through four steps: searching linear relations, recovering the modulus, recovering the coefficients, and recovering the initial state. While applicable to the case of a known

modulus but unknown coefficients, the approach does not specifically address or optimize for this scenario. To fill this gap, this subsection presents a dedicated method tailored to predicting truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs under a known modulus.

As a preliminary step for recovering the unknown coefficients, we need to find a set of annihilating polynomials for $\underline{a}^{(N)} = (a_0, a_1, \ldots, a_{N-1})$, a segment of length N of the sequence $\underline{a}$. However, since $\underline{a}^{(N)}$ is unknown, such polynomials cannot be directly determined. [1] proposed a solution: using the known truncated sequence $\underline{y}^{(N)}$ to construct a lattice, whose reduced basis yields annihilating polynomials with small coefficients. A key distinction arises from the modulus: [1] operates with an unknown modulus m, resulting in polynomials over $\mathbb{Z}$. In our known-modulus scenario, the approach diverges, as we need to find annihilating polynomials over $\mathbb{Z}/(m)$ using $\underline{y}^{(N)}$.

First, we introduce the following Lemma 2 to estimate the norm of the coefficient vector of an annihilating polynomial over $\mathbb{Z}/(m)$.

Lemma 2 *Let $\underline{a} = (a_i)_{i\geq 0}$ be a sequence generated by a Fibonacci $\mathbb{Z}/(m)$-LFSR of order n with modulus m and characteristic polynomial $f(x) = x^n - c_{n-1}x^{n-1} - \cdots - c_0 \in \mathbb{Z}/(m)[x]$. Let r and t be positive integers satisfying $r \geq t > n$, and let $\mathbf{A}_i = (a_i, a_{i+1}, \ldots, a_{i+t-1}) \in \mathbb{Z}^t$ for $i = 0, 1, \ldots, r-1$. If the matrix*

$$M_a = \begin{pmatrix} a_0 & a_1 & \cdots & a_{n-1} \\ a_1 & a_2 & \cdots & a_n \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & a_n & \cdots & a_{2n-2} \end{pmatrix}$$

is invertible over $\mathbb{Z}/(m)$, let M_a^{-1} denote the inverse matrix of M_a over $\mathbb{Z}/(m)$. Let the lattice

$$L_{\boldsymbol{\eta}} = \begin{pmatrix} mI_n & 0_{n\times(r-n)} \\ -A^T_{n\times(r-n)} & I_{r-n} \end{pmatrix},$$

where

$$A_{n\times(r-n)} = M_a^{-1} \cdot \begin{pmatrix} a_n & a_{n+1} & \cdots & a_{r-1} \\ a_{n+1} & a_{n+2} & \cdots & a_r \\ \vdots & \vdots & \ddots & \vdots \\ a_{2n-1} & a_{2n} & \cdots & a_{r+n-2} \end{pmatrix},$$

$0_{n\times(r-n)}$ is the $n \times (r-n)$ zero matrix, I_n and I_{r-n} are the identity matrices of order n and $r-n$, respectively. Then an integer vector $\boldsymbol{\eta} = (\eta_0, \eta_1, \ldots, \eta_{r-1})$ belongs to the lattice $L_{\boldsymbol{\eta}}$ if and only if the vector satisfies

$$\sum_{i=0}^{r-1} \eta_i \mathbf{A}_i \equiv \mathbf{0} \bmod m, \tag{2}$$

and the norm of the shortest vector satisfying (2) is approximately equal to $\sqrt{r/(2\pi e)} \cdot m^{\frac{n}{r}}$.

Proof According to the given conditions, the sequence $\underline{a}$ satisfies

$$a_{i+n} \equiv c_{n-1}a_{i+n-1} + c_{n-2}a_{i+n-2} + \cdots + c_0 a_i \bmod m, i = 0, 1, 2, \ldots.$$

Therefore, for any integer vector $\boldsymbol{\zeta} = (\zeta_0, \zeta_1, \ldots, \zeta_{r-1})$ that satisfies

$$\sum_{i=0}^{r-1} \zeta_i a_i \equiv 0 \bmod m, \sum_{i=0}^{r-1} \zeta_i a_{i+1} \equiv 0 \bmod m, \ldots, \sum_{i=0}^{r-1} \zeta_i a_{i+n-1} \equiv 0 \bmod m,$$

it naturally follows that

$$\begin{aligned}\sum_{i=0}^{r-1}\zeta_i\mathbf{A}_i &\equiv \left(\sum_{i=0}^{r-1}\zeta_i a_i, \sum_{i=0}^{r-1}\zeta_i a_{i+1}, \ldots, \sum_{i=0}^{r-1}\zeta_i a_{i+t-1}\right)\\ &\equiv \left(\sum_{i=0}^{r-1}\zeta_i a_i, \sum_{i=0}^{r-1}\zeta_i a_{i+1}, \ldots, \sum_{i=0}^{r-1}\zeta_i a_{i+n-1}, \sum_{i=0}^{r-1}\zeta_i \sum_{j=0}^{n-1} c_j a_{i+j}, \ldots, \sum_{i=0}^{r-1}\zeta_i \sum_{j=0}^{n-1} c_j a_{i+j+t-n-1}\right)\\ &\equiv \left(\sum_{i=0}^{r-1}\zeta_i a_i, \sum_{i=0}^{r-1}\zeta_i a_{i+1}, \ldots, \sum_{i=0}^{r-1}\zeta_i a_{i+n-1}, \sum_{j=0}^{n-1} c_j \sum_{i=0}^{r-1}\zeta_i a_{i+j}, \ldots, \sum_{j=0}^{n-1} c_j \sum_{i=0}^{r-1}\zeta_i a_{i+j+t-n-1}\right)\\ &\equiv \mathbf{0} \bmod m.\end{aligned}$$

If the matrix M_a is invertible over $\mathbb{Z}/(m)$, every vector $\boldsymbol{\eta} = (\eta_0, \eta_1, \ldots, \eta_{r-1})$ in the lattice $L_{\boldsymbol{\eta}}$ satisfies (2). In fact, for any vector $\boldsymbol{\eta} = (\eta_0, \eta_1, \ldots, \eta_{r-1}) \in L_{\boldsymbol{\eta}}$, there exists an n-dimensional integer vector $(k_0, k_1, \ldots, k_{n-1})$ such that

$$(k_0, k_1, \ldots, k_{n-1}, \eta_n, \eta_{n+1}, \ldots, \eta_{r-1}) \cdot L_{\boldsymbol{\eta}} = (\eta_0, \eta_1, \ldots, \eta_{r-1}).$$

It follows that

$$(k_0, k_1, \ldots, k_{n-1}) \cdot mI_n - (\eta_n, \eta_{n+1}, \ldots, \eta_{r-1}) \cdot A^T_{n\times(r-n)} = (\eta_0, \eta_1, \ldots, \eta_{n-1}),$$

and thus

$$\begin{pmatrix} I_n & A_{n\times(r-n)}\end{pmatrix} \cdot \begin{pmatrix}\eta_0\\ \eta_1\\ \vdots\\ \eta_{r-1}\end{pmatrix} \equiv \mathbf{0} \bmod m.$$

Left-multiply both sides of the above equation by M_a, we have

$$\begin{pmatrix} a_0 & a_1 & \cdots & a_{r-1}\\ a_1 & a_2 & \cdots & a_r\\ \vdots & \vdots & \ddots & \vdots\\ a_{n-1} & a_n & \cdots & a_{r+n-2}\end{pmatrix} \cdot \begin{pmatrix}\eta_0\\ \eta_1\\ \vdots\\ \eta_{r-1}\end{pmatrix} \equiv \mathbf{0} \bmod m,$$

which means

$$\sum_{i=0}^{r-1}\eta_i a_i \equiv 0 \bmod m, \sum_{i=0}^{r-1}\eta_i a_{i+1} \equiv 0 \bmod m, \ldots, \sum_{i=0}^{r-1}\eta_i a_{i+n-1} \equiv 0 \bmod m,$$

or equivalently,

$$\sum_{i=0}^{r-1}\eta_i \mathbf{A}_i \equiv \mathbf{0} \bmod m.$$

Conversely, it is straightforward to verify that any vector $\boldsymbol{\eta}$ satisfying (2) belongs to the lattice $L_{\boldsymbol{\eta}}$.

Consider the shortest vector in the lattice $L_{\boldsymbol{\eta}}$, which naturally validates (2). The norm of this vector can be approximated by Lemma 1 as

$$\lambda_1(L_{\boldsymbol{\eta}}) \approx \sqrt{\frac{r}{2\pi e}} m^{\frac{n}{r}}.$$

Therefore, if the matrix M_a is invertible over $\mathbb{Z}/(m)$, then the norm of the shortest vector satisfying (2) is approximately equal to $\sqrt{r/(2\pi e)} \cdot m^{\frac{n}{r}}$. □

Next, we give the specific process to find $\boldsymbol{\eta} = (\eta_0, \eta_1, \ldots, \eta_{r-1})$ that validates (2) using the known truncated sequence $\underline{y}^{(N)}$.

Choose suitable positive integers r and t satisfying $r \geq t > n$ and $r+t-1 \leq N$, let

$$\mathbf{Y}_i = (y_i, y_{i+1}, \ldots, y_{i+t-1}) \in \mathbb{Z}^t, i = 0, 1, \ldots, r-1,$$

and

$$\mathbf{Z}_i = (z_i, z_{i+1}, \ldots, z_{i+t-1}) \in \mathbb{Z}^t, i = 0, 1, \ldots, r-1.$$

If the integer vector $\boldsymbol{\eta} = (\eta_0, \eta_1, \ldots, \eta_{r-1})$ satisfies (2), it follows from (1) that

$$2^\beta \sum_{i=0}^{r-1} \eta_i \mathbf{Y}_i + \sum_{i=0}^{r-1} \eta_i \mathbf{Z}_i \equiv \mathbf{0} \bmod m, \tag{3}$$

and so

$$2^\beta \sum_{i=0}^{r-1} \eta_i \mathbf{Y}_i \equiv -\sum_{i=0}^{r-1} \eta_i \mathbf{Z}_i \bmod m.$$

Construct the following lattice

$$L_{m,y} = \begin{pmatrix} mI_t & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ 2^\beta \mathbf{Y}_0 & 2^\beta & 0 & \cdots & 0 \\ 2^\beta \mathbf{Y}_1 & 0 & 2^\beta & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2^\beta \mathbf{Y}_{r-1} & 0 & 0 & \cdots & 2^\beta \end{pmatrix},$$

and obviously, the $(r+t)$-dimensional vector

$$\mathbf{v}_{m,y} = \left(\left\lceil 2^\beta \sum_{i=0}^{r-1} \eta_i \mathbf{Y}_i \right\rfloor_{\bmod m}, 2^\beta \eta_0, 2^\beta \eta_1, \ldots, 2^\beta \eta_{r-1} \right) \in L_{m,y}.$$

Heuristically, if $\boldsymbol{\eta} = (\eta_0, \eta_1, \ldots, \eta_{r-1})$ is a short vector that satisfies (2), then we may expect

$$\left\lceil 2^\beta \sum_{i=0}^{r-1} \eta_i \mathbf{Y}_i \right\rfloor_{\bmod m} = \left\lceil -\sum_{i=0}^{r-1} \eta_i \mathbf{Z}_i \right\rfloor_{\bmod m} = -\sum_{i=0}^{r-1} \eta_i \mathbf{Z}_i$$

to be correspondingly short, which aligns with our experimental observations. This, in turn, suggests that $\mathbf{v}_{m,y}$ is also a short vector. Therefore, the desired vector $\boldsymbol{\eta}$ that validates (2) can be obtained by applying lattice reduction algorithms to the lattice $L_{m,y}$.

For a long time, the selection of parameters r and t in truncated Fibonacci $\mathbb{Z}/(m)$-LFSR prediction has lacked a practical heuristic, typically relying on extensive experimental search. To address this gap and support experimental validation, we present a heuristic analysis below for selecting r and t based on Lemmas 1 and 2.

According to Lemma 2, if the matrix M_a is invertible over $\mathbb{Z}/(m)$, then there exists an integer vector $\boldsymbol{\eta} = (\eta_0, \eta_1, \ldots, \eta_{r-1})$ that validates (2), and its norm is

approximately $\|\boldsymbol{\eta}\| \approx \sqrt{r/(2\pi e)} \cdot m^{\frac{n}{r}}$. Heuristically, since the vectors in the reduced basis of the lattice $L_{m,y}$ should be balanced in each coordinate, the corresponding vector

$$\|\mathbf{v}_{m,y}\| \approx \sqrt{\frac{r+t}{r}} \cdot 2^{\beta} \cdot \|\boldsymbol{\eta}\| \approx \sqrt{\frac{r+t}{2\pi e}} \cdot 2^{\beta} \cdot m^{\frac{n}{r}}.$$

We note that by Lemma 1,

$$\lambda_1(L_{m,y}) \approx \sqrt{\frac{r+t}{2\pi e}} \cdot m^{\frac{t}{r+t}} 2^{\frac{\beta r}{r+t}}.$$

If

$$\sqrt{\frac{r+t}{2\pi e}} \cdot 2^{\beta} \cdot m^{\frac{n}{r}} \leq \sqrt{\frac{r+t}{2\pi e}} \cdot m^{\frac{t}{r+t}} 2^{\frac{\beta r}{r+t}},$$

i.e.,

$$\frac{1}{r} + \frac{1}{t} \leq \frac{\log m - \beta}{n \log m}, \tag{4}$$

then the desired vector $\boldsymbol{\eta}$ that validates (2) may be found in the reduced basis of the lattice $L_{m,y}$. By (4)

$$\begin{aligned} r + t &\geq r + \frac{n \log m \cdot r}{(\log m - \beta) r - n \log m} \\ &= \frac{(\log m - \beta) r^2}{(\log m - \beta) r - n \log m} \\ &= \frac{\log m - \beta}{-n \log m \cdot \left(\frac{1}{r} - \frac{\log m - \beta}{2n \log m}\right)^2 + \frac{(\log m - \beta)^2}{4n \log m}} \\ &\geq \frac{4n \log m}{\log m - \beta}, \end{aligned} \tag{5}$$

where equality holds if and only if

$$r = t = \frac{2n \log m}{\log m - \beta}. \tag{6}$$

Since constructing the lattice $L_{m,y}$ requires $r+t-1$ truncated digits $y_0, y_1, \ldots, y_{r+t-2}$, at least $2 \times \lceil 2n \log m/(\log m - \beta) \rceil - 1$ truncated digits are required to potentially find a vector $\boldsymbol{\eta}$ that satisfies (2). In practical experiments, we can incrementally increase the values of r and t from this lower bound $\lceil 2n \log m/(\log m - \beta) \rceil$ until a substantial number of $\boldsymbol{\eta}$ that satisfy (2) are found. Experiments confirm that (6) is attainable for some small-scale instances. For large-scale instances such as ZUC's driving sequence, the actual values of r and t approach the estimation as more powerful SVP algorithms are employed. Although practical constraints of time and memory currently prevent these large instances from fully reaching (6), the observed trend provides strong evidence for the quality of (6). It marks a significant advance over the previous strategy of blindly searching the parameter space.

Remark 1 If the values of r and t are chosen appropriately, a single call to a lattice reduction algorithm can generate multiple sets of $\eta_0, \eta_1, \ldots, \eta_{r-1}$ that satisfy (2). In our experiments, we typically require no more than 3 sets of integer coefficients $\eta_0, \eta_1, \ldots, \eta_{r-1}$ to successfully recover the modulus and 2 sets to recover the coefficients in most cases.

As discussed above, by appropriately selecting the values of r and t, we can get $\eta_0, \eta_1, \ldots, \eta_{r-1}$ that validates (2) by lattice reduction algorithms, i.e.,

$$\begin{cases} \eta_0 a_0 + \eta_1 a_1 + \cdots + \eta_{r-1} a_{r-1} \equiv 0 \bmod m \\ \eta_0 a_1 + \eta_1 a_2 + \cdots + \eta_{r-1} a_r \equiv 0 \bmod m \\ \vdots \\ \eta_0 a_{t-1} + \eta_1 a_t + \cdots + \eta_{r-1} a_{r+t-2} \equiv 0 \bmod m \end{cases}.$$

Obviously, this implies that we have got a polynomial

$$F(x) = \eta_{r-1} x^{r-1} + \cdots + \eta_1 x + \eta_0,$$

which annihilates the sequence $\underline{a}^{(r+t-1)} = (a_0, a_1, \ldots, a_{r+t-2})$ over $\mathbb{Z}/(m)$. If we can obtain a sufficient number of such polynomials, the coefficients can be recovered by computing the greatest common divisor of these polynomials. To be specific, let $m = p_1^{e_1} p_2^{e_2} \cdots p_s^{e_s}$ be the canonical factorization of the modulus m and let $F^{(1)}(x), F^{(2)}(x), \ldots, F^{(w)}(x)$ be the obtained annihilating polynomials. To recover the characteristic polynomial $f(x) = x^n - c_{n-1}x^{n-1} - \cdots - c_0$, we can first get $f(x) \bmod p_j$ by computing the greatest common divisor of $F^{(i)}(x) \bmod p_j$ for every $j \in \{1, 2, \ldots, s\}$, then get $f(x) \bmod p_j^{e_j}$ using the Extended Euclidean algorithm over finite fields layer by layer, and finally derive $f(x)$ using the Chinese Remainder Theorem. In practice, the coefficients $c_0, c_1, \ldots, c_{n-1}$ can be successfully recovered using only 2 annihilating polynomials in most cases. See [1] for more details.

After recovering the unknown coefficients, the next step is to recover the initial state $a_0, a_1, \ldots, a_{n-1}$. Since the truncated digits $y_0, y_1, \ldots, y_{n-1}$ are known, by (1), we only need to recover $z_0, z_1, \ldots, z_{n-1}$, which has been proven to be equivalent to finding small integer solutions to systems of linear congruences, i.e., the SIS problem [28]. To solve this, we employ Kannan's embedding technique [40] to transform this SIS problem into an SVP. By applying lattice reduction algorithms to solve the SVP, we can recover $z_0, z_1, \ldots, z_{n-1}$. See [1] for more details.

Remark 2 Combined with the techniques in [30], this method can also be extended to predict truncated Galois $\mathbb{Z}/(m)$-LFSRs with a known modulus but unknown coefficients by appropriately adapting the lattice $L_{m,y}$.

3.2 Predicting truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs with unknown parameters

As discussed in Subsection 3.1, when predicting truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs with unknown parameters, the annihilating polynomials found by [1] are over $\mathbb{Z}$ because the modulus m is unknown. In this subsection, we prove that if the difference between 2^k and the modulus m satisfies $\Delta_m \triangleq 2^k - m < 2^\beta$ (or m satisfies $m - 2^{k-1} < 2^\beta$), then the annihilating polynomials over $\mathbb{Z}/(m)$ (rather than $\mathbb{Z}$) can still be found by a lattice-based method, even when m is unknown.

By (3), if the vector $\boldsymbol{\eta} = (\eta_0, \eta_1, \ldots, \eta_{r-1})$ satisfies (2), then there exists a t-dimensional integer vector $\mathbf{K}$ such that

$$2^\beta \sum_{i=0}^{r-1} \eta_i \mathbf{Y}_i + \sum_{i=0}^{r-1} \eta_i \mathbf{Z}_i = \mathbf{K}m = \mathbf{K}(2^k - \Delta_m) = \mathbf{K}(2^{\alpha+\beta} - \Delta_m).$$

Divide both sides of the equation by 2^β, we get

$$\sum_{i=0}^{r-1} \eta_i \mathbf{Y}_i + \frac{\sum_{i=0}^{r-1} \eta_i \mathbf{Z}_i + \mathbf{K}\Delta_m}{2^\beta} = \mathbf{K}2^\alpha,$$

and thus

$$\sum_{i=0}^{r-1} \eta_i \mathbf{Y}_i \equiv -\frac{\sum_{i=0}^{r-1} \eta_i \mathbf{Z}_i + \mathbf{K}\Delta_m}{2^\beta} \bmod 2^\alpha.$$

Construct the following lattice

$$L_{\alpha,y} = \begin{pmatrix} 2^\alpha I_t & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{Y}_0 & 1 & 0 & \cdots & 0 \\ \mathbf{Y}_1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{Y}_{r-1} & 0 & 0 & \cdots & 1 \end{pmatrix},$$

and obviously, the $(r+t)$-dimensional vector

$$\mathbf{v}_{\alpha,y} = \left(\left\lceil \sum_{i=0}^{r-1} \eta_i \mathbf{Y}_i \right\rfloor_{\bmod 2^\alpha}, \eta_0, \eta_1, \ldots, \eta_{r-1} \right) \in L_{\alpha,y}.$$

Note that $0 \le z_i < 2^\beta$ for each $i \in \{0, 1, \ldots, N-1\}$. Heuristically, if $\Delta_m < 2^\beta$ and $\boldsymbol{\eta} = (\eta_0, \eta_1, \ldots, \eta_{r-1})$ is a short vector that satisfies (2), then we may expect

$$\left\lceil \sum_{i=0}^{r-1} \eta_i \mathbf{Y}_i \right\rfloor_{\bmod 2^\alpha} = \left\lceil -\frac{\sum_{i=0}^{r-1} \eta_i \mathbf{Z}_i + \mathbf{K}\Delta_m}{2^\beta} \right\rfloor_{\bmod 2^\alpha} = -\frac{\sum_{i=0}^{r-1} \eta_i \mathbf{Z}_i + \mathbf{K}\Delta_m}{2^\beta}$$

to be correspondingly short, which aligns with our experimental observations. This, in turn, suggests that $\mathbf{v}_{\alpha,y}$ is also a short vector. Therefore, the desired vector $\boldsymbol{\eta}$ that validates (2) can be obtained by applying lattice reduction algorithms to the lattice $L_{\alpha,y}$.

Similar to Subsection 3.1, we provide a heuristic analysis below for selecting the values of r and t by Lemmas 1 and 2.

According to Lemma 2, if the matrix M_a is invertible over $\mathbb{Z}/(m)$, then there exists an integer vector $\boldsymbol{\eta} = (\eta_0, \eta_1, \ldots, \eta_{r-1})$ that validates (2), and its norm is approximately $\|\boldsymbol{\eta}\| \approx \sqrt{r/(2\pi e)} \cdot m^{\frac{n}{r}}$. Heuristically, since the vectors in the reduced basis of the lattice $L_{\alpha,y}$ should be balanced in each coordinate, the corresponding vector

$$\|\mathbf{v}_{\alpha,y}\| \approx \sqrt{\frac{r+t}{r}} \cdot \|\boldsymbol{\eta}\| \approx \sqrt{\frac{r+t}{2\pi e}} \cdot m^{\frac{n}{r}}.$$

We note that by Lemma 1,

$$\lambda_1(L_{\alpha,y}) \approx \sqrt{\frac{r+t}{2\pi e}} \cdot 2^{\frac{\alpha t}{r+t}}.$$

If

$$\sqrt{\frac{r+t}{2\pi e}} \cdot m^{\frac{n}{r}} \leq \sqrt{\frac{r+t}{2\pi e}} \cdot 2^{\frac{\alpha t}{r+t}},$$

i.e.,

$$\frac{1}{r} + \frac{1}{t} \leq \frac{\alpha}{n \log m},$$

then the desired vector $\boldsymbol{\eta}$ that validates (2) may be found in the reduced basis of the lattice $L_{\alpha,y}$. Similarly, at least $2 \times \lceil 2n \log m/\alpha \rceil - 1$ truncated digits are required to potentially find a vector $\boldsymbol{\eta}$ that satisfies (2), where $r = t = \lceil 2n \log m/\alpha \rceil$. This implies that we have obtained an annihilating polynomial $F(x) = \eta_{r-1}x^{r-1} + \cdots + \eta_1 x + \eta_0$ of the sequence $\underline{a}^{(r+t-1)} = (a_0, a_1, \ldots, a_{r+t-2})$ over $\mathbb{Z}/(m)$.

By Theorem 1, if a sufficient number of annihilating polynomials are obtained, the modulus m can be recovered by computing the greatest common divisor of the resultants of these polynomials, provided that the resultants are nonzero. As discussed in [1], the proportion that the resultant of any two annihilating polynomials equals 0 is negligible, ensuring that we have ample options to obtain nonzero resultants. In practice, no more than 3 nonzero resultants typically suffice to recover m in most cases. Once m is successfully recovered, we can sequentially employ the greatest common factor method and Kannan's embedding technique in turn to recover the coefficients $c_0, c_1, \ldots, c_{n-1}$ and the initial state $a_0, a_1, \ldots, a_{n-1}$ as described in Subsection 3.1.

Remark 3 Combined with the techniques in [30], this method can also be extended to predict truncated Galois $\mathbb{Z}/(m)$-LFSRs with unknown parameters by appropriately adapting the lattice $L_{\alpha,y}$.

4 Experimental results

We have conducted extensive experiments to verify the correctness and effectiveness of our method. All experiments related to the BKZ algorithm in this section are conducted on a general-purpose computer (Windows 10, Processor Intel(R) Xeon(R) W-2255 CPU @ 3.70 GHz, 10 Core(s) (20 logical processors), RAM: 160 GB (160 GB usable)). Experiments involving the BGJ1 sieve [43] are performed on an Ubuntu 22.04.5 LTS virtual machine hosted on the same physical hardware, configured with 16 virtual CPUs and 62 GB of memory. Unless otherwise specified, the BKZ algorithm is used from the NTL 11.4.3 library [44] with a reduction parameter of $\delta = 0.99$ and a blocksize of 20. The BGJ1 sieve is utilized from the G6K library [45]. The source code is available at: https://pastebin.com/nvtFXmHw.

In this section, we present three experiments to validate different aspects of our method. The first experiment evaluates the estimation of parameters r and t given in (6). Tests are performed on both the ZUC's driving sequence and a Fibonacci $\mathbb{Z}/(2^{13}-1)$-LFSR, aiming to determine the minimal values of r and t required to find annihilating polynomials by the high-order truncated digits when the modulus m is known. The second and third experiments focus on recovering the unknown parameters and the initial state by the high-order truncated digits under the following two situations, respectively: (1) when the modulus is known but the coefficients are unknown, and (2) when both the modulus and coefficients are unknown, but the modulus is close to a power of 2. To ensure a fair comparison with the method presented in [1] and to verify the applicability of our method to general truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs, each experiment comprises two parts: (1) the ZUC's driving sequence, and (2) random Fibonacci $\mathbb{Z}/(m)$-LFSRs of order $n = 16$ with a 31-bit modulus.

In the first experiment, we assume that the coefficients $c_0, c_1, \ldots, c_{n-1}$ and the initial state $a_0, a_1, \ldots, a_{n-1}$ are unknown. Given the modulus m and the first N truncated digits $y_0, y_1, \ldots, y_{N-1}$, our objective is to determine the minimal values of r and t for finding the annihilating polynomials over $\mathbb{Z}/(m)$, which in turn defines the minimal truncated sequence length $N = r + t - 1$.

Exp. 1.1 (ZUC): The characteristic polynomial of ZUC's driving sequence is a primitive polynomial of degree $n = 16$ over $\mathbb{Z}/(2^{31}-1)$, i.e.,

$$f(x) = x^{16} - (2^{15}x^{15} + 2^{17}x^{13} + 2^{21}x^{10} + 2^{20}x^4 + 2^8 + 1).$$

We choose the same initial state as in [1] that

$$\begin{gathered} a_0 = 5113033, a_1 = 101513454, a_2 = 342517303, a_3 = 602278014, \\ a_4 = 545182365, a_5 = 2109832480, a_6 = 83899935, a_7 = 1353806595, \\ a_8 = 56274808, a_9 = 220365776, a_{10} = 1900047307, a_{11} = 1287703311, \\ a_{12} = 754976298, a_{13} = 344592060, a_{14} = 2072123851, a_{15} = 1906585150, \end{gathered}$$

and then generate the sequence $\underline{a} = (a_i)_{i\geq 0} \in G(f(x), 2^{31}-1)$. Let $a_i = 2^{14}y_i + z_i$, where y_i denotes the 17 high-order bits of a_i. Table 1 summarizes the minimal values of r and t required to identify annihilating polynomials using different SVP algorithms.

Table 1: ZUC: The minimal values of r and t

SVP algorithm	r	t	N	Wall time	CPU time
BKZ-20	71	71	141	73.8 s	73.8 s
BKZ-30	69	69	137	2 h	2 h
BKZ-32	68	68	135	13.4 h	13.4 h
BGJ1	64	64	127	5.5 h	38.2 h

By (5), finding an annihilating polynomial requires at least 117 truncated digits, where $r = t = 59$. There is a slight discrepancy between this theoretical estimation ($r = t = 59$) and the best experimental result ($r = t = 64$). This discrepancy may arise from the following two reasons:

1. We heuristically consider that the components of $\mathbf{v}_{m,y}$ are relatively balanced, which makes the estimation of $\|\mathbf{v}_{m,y}\|$ somewhat rough and affects the accuracy of the inequality (4).

2. Table 1 shows that the minimal values of r and t are progressively reduced as more powerful SVP algorithms are employed: from 71 with BKZ-20, to 69 and 68 with larger blocksizes, and further to 64 with the BGJ1 sieve. Due to time and memory constraints, we do not explore whether even smaller values of r and t are feasible. Nonetheless, the trend strongly suggests that with more advanced SVP solvers and sufficient computational resources, it may be possible to reach the theoretical bound of $r = t = 59$.

Despite this discrepancy, the descending trend confirms the soundness of the estimation and its practical value for experimental guidance by providing a starting point for determining suitable values of r and t when predicting Fibonacci $\mathbb{Z}/(m)$-LFSRs. This fills a gap in the previous lack of effective analysis regarding the selection of r and t.

Exp. 1.2 (Fibonacci $\mathbb{Z}/(2^{13}-1)$-LFSR): To further validate the reasonableness of our estimation, we also test it on a small-scale Fibonacci $\mathbb{Z}/(m)$-LFSR. Its characteristic polynomial is a randomly chosen primitive polynomial of degree $n = 8$ over $\mathbb{Z}/(2^{13}-1)$, i.e.,

$$f(x) = x^8 + 3295x^7 + 1574x^6 + 1676x^5 + 4719x^4 + 3842x^3 + 1450x^2 + 908x + 1267.$$

Let $a_i = 2^4 y_i + z_i$, where y_i denotes the 9 high-order bits of a_i. We randomly select 30,000 initial states and employ the method described in Subsection 3.1, along with the BKZ-20 algorithm, to determine the minimal values of r and t needed to find annihilating polynomials in each instance. The results are detailed in Table 2.

Table 2: Fibonacci $\mathbb{Z}/(2^{13}-1)$-LFSR: The minimal values of r and t

r, t	24	25	26	27	28	29
Number of instances	2	1523	16980	11084	410	1

By (5), finding an annihilating polynomial requires at least 47 truncated digits, where $r = t = 24$. In 2 of the 30,000 instances, annihilating polynomials are successfully found exactly at $r = t = 24$. Moreover, 98.63% of all instances require values of r and t no larger than 27.

In the second experiment, we assume that the coefficients $c_0, c_1, \ldots, c_{15}$ and the initial state $a_0, a_1, \ldots, a_{15}$ are unknown. Given the modulus m and the first N truncated digits $y_0, y_1, \ldots, y_{N-1}$ (where each y_i denotes the 17 high-order bits of a_i), our goal is to recover the unknown information above.

Exp. 2.1 (ZUC): With the initial state of Exp. 1.1, we generate the sequence $\underline{a} = (a_i)_{i\geq 0} \in G(f(x), 2^{31} - 1)$. Given the modulus $m = 2^{31} - 1$ and the first N truncated digits $y_0, y_1, \ldots, y_{N-1}$, we aim to recover the coefficients and the initial state. A comparison of our method (Subsection 3.1) with the method presented in [1] is summarized in Table 3. The column labeled "[1]" presents the original data from [1], while "[1] Replication" column shows results from our replication on our own computer to ensure a fair comparison under identical computational conditions.

Table 3: ZUC: The comparison of our method with the method in [1]

		[1]	[1] Replication	Exp. 2.1 Subsec. 3.1	Exp. 3.1 Subsec. 3.2
Search annihilating polynomials	r, t	$175, 65$	$175, 65$	$71, 71$	$72, 70$
	Size of lattice	175×240	175×240	142×142	142×142
	Number of annihilating polynomials	110	110	71	72
	Total time	9.4 min	228.851 s	73.797 s	55.029 s
Recover the modulus	Number of annihilating polynomials	3	3	–	3
	Total time	2.625 s	0.606 s	–	0.060 s
Recover the coefficients	Number of annihilating polynomials	2	2	2	2
	Total time	0.075 s	0.029 s	0.007 s	0.015 s
Recover the initial state	Number of truncated digits	30	30	30	30
	Size of lattice	31×31	31×31	31×31	31×31
	Total time	0.260 s	0.111 s	0.110 s	0.114 s
Total time		9.45 min	3.83 min	1.23 min	0.92 min
Number of truncated digits		239	239	141	141

As shown in Table 3, we first select $r = 71$ and $t = 71$, and then construct a 142×142 lattice following the methodology described in Subsection 3.1. The BKZ-reduced basis of this lattice yields 71 annihilating polynomials over $\mathbb{Z}/(m)$. Next, we utilize the first 2 annihilating polynomials to compute their greatest common divisor to successfully recover the coefficients $c_0, c_1, \ldots, c_{15}$. Finally, we recover the initial

state $a_0, a_1, \ldots, a_{15}$ using the first 30 truncated digits. Notably, our method requires only 141 truncated digits to find annihilating polynomials over $\mathbb{Z}/(m)$, whereas the method in [1] requires 239 truncated digits to find annihilating polynomials over $\mathbb{Z}$. This demonstrates the clear superiority of our method for predicting truncated ZUC's driving sequence when m is known.

Exp. 2.2 (General Fibonacci $\mathbb{Z}/(m)$-LFSRs): Assume the modulus m is a random 31-bit integer, the coefficients $c_0, c_1, \ldots, c_{15}$ and the initial state $a_0, a_1, \ldots, a_{15}$ are randomly chosen not-all-zero integers from $\mathbb{Z}/(m)$. The parameters are fixed as $t = 65$, $k = 31$ (bit-length of $m - 1$), $n = 16$ (order of $\mathbb{Z}/(m)$-LFSRs), and $\alpha = 17$ (bit-length of high-order digits). Given the modulus m and the first N truncated digits $y_0, y_1, \ldots, y_{N-1}$, we compare the success rates of our method (Subsection 3.1) with the method presented in [1] in searching annihilating polynomials. For each r, 100 random trials are conducted. Results are detailed in Table 4.

Table 4: Searching annihilating polynomials: unknown m vs. known m

	[1] (m is unknown)		Subsec. 3.1 (m is known)	
t	r	success rate	r	success rate
65	167	6%	74	1%
	177	39%	79	20%
	187	68%	84	64%
	197	90%	89	89%
	207	96%	94	99%
	217	98%	99	100%

As evidenced by Table 4, for general 31-bit moduli, if m is known, selecting $r = 84$ yields a success probability exceeding 60% in finding annihilating polynomials over $\mathbb{Z}/(m)$, and selecting $r = 99$ guarantees an extremely high probability of success. In contrast, if m is unknown, it becomes nearly impossible to find annihilating polynomials over $\mathbb{Z}$ when $r = 99$. To achieve a success rate of over 90%, r must be increased to around 197. Consequently, for general moduli, predicting Fibonacci $\mathbb{Z}/(m)$-LFSRs requires significantly fewer truncated digits when m is known.

In the third experiment, we assume that the modulus m, the coefficients $c_0, c_1, \ldots, c_{15}$, and the initial state $a_0, a_1, \ldots, a_{15}$ are all unknown. Given the first N truncated digits $y_0, y_1, \ldots, y_{N-1}$ (where each y_i denotes the 17 high-order bits of a_i), our goal is to recover the unknown information above.

Exp. 3.1 (ZUC): The initial state is set to be identical to that in Exp. 1.1. Given the first N truncated digits $y_0, y_1, \ldots, y_{N-1}$, we need to recover the modulus, the coefficients, and the initial state. A comparison of our method (Subsection 3.2) with the method presented in [1] is provided in Table 3.

As shown in Table 3, we first select $r = 72$ and $t = 70$, and then construct a lattice following Subsection 3.2. Its BKZ-reduced basis yields 72 annihilating polynomials over $\mathbb{Z}/(m)$, of which the first 3 annihilating polynomials suffice to recover the modulus m and the first 2 are adequate to determine the coefficients $c_0, c_1, \ldots, c_{15}$. Finally,

we use 30 truncated digits to recover the initial state $a_0, a_1, \ldots, a_{15}$. In summary, our method recovers all unknown parameters using only 141 truncated digits within 1 min. In contrast, the method in [1] requires 239 truncated digits and 3.83 min to finish the experiment. In other words, compared to the method in [1], our improved approach reduces the number of truncated digits N by 41% and cuts the running time to approximately 1/4 of that required by [1]. Specifically, the method in [1] reduces a 175×240 lattice to obtain 110 annihilating polynomials of degree 174 over $\mathbb{Z}$, whereas we only need to reduce a 142×142 lattice to derive 72 annihilating polynomials of degree 71 over $\mathbb{Z}/(m)$. The reduction in lattice size not only accelerates the process of finding a sufficient number of annihilating polynomials, but also lowers the degrees of the obtained annihilating polynomials. As a result, this further reduces the time spent on performing operations on these polynomials when recovering the modulus and coefficients.

Exp. 3.2 (General Fibonacci $\mathbb{Z}/(m)$-LFSRs): Assume the modulus m is a random 31-bit integer satisfying $\Delta_m < 2^\beta$. The coefficients $c_0, c_1, \ldots, c_{15}$ and the initial state $a_0, a_1, \ldots, a_{15}$ are randomly chosen not-all-zero integers from $\mathbb{Z}/(m)$. The parameters are fixed as $t = 65$, $k = 31$ (bit-length of $m - 1$), $n = 16$ (order of $\mathbb{Z}/(m)$-LFSRs), and $\alpha = 17$ (bit-length of high-order digits). Given the first N truncated digits $y_0, y_1, \ldots, y_{N-1}$, we compare the success rates of our method (Subsection 3.2) with the method presented in [1] in searching annihilating polynomials. For each r, 100 random trials are conducted. Results are detailed in Table 5.

Table 5: Searching annihilating polynomials: [1] vs. Subsection 3.2

	[1] (m is unknown and $\Delta_m < 2^\beta$)		Subsec. 3.2 (m is unknown and $\Delta_m < 2^\beta$)	
t	r	success rate	r	success rate
65	162	5%	74	1%
	167	27%	79	30%
	172	82%	84	52%
	177	97%	89	81%
	182	99%	94	98%
	187	100%	99	100%

As shown in Table 5, for moduli m satisfying $\Delta_m < 2^\beta$, our improved method can achieve a success rate of over 50% when $r = 84$, and it can essentially ensure the discovery of annihilating polynomials when $r = 99$. In contrast, the method in [1] requires at least $r = 172$ to guarantee a success rate of over 80%. Therefore, if m is unknown but an approximation is available (i.e., $\Delta_m < 2^\beta$), our improved method can effectively reduce the number of truncated digits required for predicting Fibonacci $\mathbb{Z}/(m)$-LFSRs.

5 Conclusion

In this paper, we study the unpredictability of truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs,

$$a_{i+n} \equiv c_{n-1}a_{i+n-1} + c_{n-2}a_{i+n-2} + \cdots + c_0 a_i \bmod m, i = 0, 1, 2, \ldots.$$

First, we provide a practical heuristic estimation for the values of r and t based on lattice theory, thereby avoiding previous blind search. Next, we develop a lattice-based method for finding annihilating polynomials over $\mathbb{Z}/(m)$ by the high-order truncated digits when the modulus m is known but coefficients are unknown, filling a methodological gap for this specific case. Finally, we demonstrate that when both the modulus m and coefficients are unknown but m is close to a power of 2, our lattice constructed by the high-order truncated digits can yield annihilating polynomials over $\mathbb{Z}/(m)$ rather than $\mathbb{Z}$, achieving significant reductions in runtime and truncated digits. Extensive experiments have verified the correctness and effectiveness of the above methods.

Current studies on the unpredictability of truncated Fibonacci $\mathbb{Z}/(m)$-LFSRs primarily focus on scenarios where either the high-order bits or the low-order bits are truncated. In terms of bit leakage, truncation can occur in middle or even multiple segments. Whether practically feasible prediction algorithms exist for cases involving the middle-segment truncated bits, or even multi-segment truncated bits, remains an open problem worthy of further investigation.

Acknowledgements. This work was supported by the National Key Research and Development Program of China (No. 2024YFB4504700) and the National Natural Science Foundation of China (No. 12371526, No. 62202494).

References

[1] Yu, H.-B., Zheng, Q.-X., Liu, Y.J., Bi, J.G., Duan, Y.F., Xue, J.W., Wu, Y., Cao, Y., Cheng, R., Wang, L., Sun, B.S.: An improved method for predicting truncated multiple recursive generators with unknown parameters. Des. Codes Cryptogr. **91**(5), 1713–1736 (2023)

[2] Briceno, M., Goldberg, I., Wagner, D.: A pedagogical implementation of A5/1 (1999). http://www.scard.org

[3] Bluetooth SIG: Specification of the bluetooth system (2001). http://www.bluetooth.com

[4] ETSI/SAGE: Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3. Document 2: ZUC Specification (2011)

[5] ETSI/SAGE: Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2. Document 2: SNOW3G Specification (2006)

[6] Ekdahl, P., Johansson, T., Maximov, A., Yang, J.: A new snow stream cipher called snow-v. IACR Trans. Symmetric Cryptol. **3**, 1–42 (2019)

[7] Golomb, S.W.: Shift Register Sequences. Aegean Park Press, Laguna Hills, CA (1981)

[8] Meier, W., Staffelbach, O.: Fast correlation attacks on certain stream ciphers. J. Cryptol. **1**, 159–176 (1989)

[9] Courtois, N.T.: Fast algebraic attacks on stream ciphers with linear feedback. In: Advances in Cryptology–CRYPTO 2003, pp. 176–194. Springer, Berlin, Heidelberg (2003)

[10] Zierler, N.: Linear recurring sequences. J. Soc. Ind. Appl. Math. **7**(1), 31–48 (1959)

[11] L'Ecuyer, P.: Good parameters and implementations for combined multiple recursive random number generators. Oper. Res. **47**(1), 159–164 (1999)

[12] Chen, H.J., Qi, W.F.: On the distinctness of maximal length sequences over $\mathbb{Z}/(pq)$ modulo 2. Finite Fields Appl. **15**(2), 23–39 (2009)

[13] Massey, J.L.: Shift register synthesis and bch decoding. IEEE Trans. Inf. Theory **15**(1), 122–127 (1969)

[14] Berlekamp, E.R.: Algebraic Coding Theory. McGraw-Hill, New York (1968)

[15] Sugiyama, Y., Kasahara, M., Hirasawa, S., Namekawa, T.: A method for solving key equation for decoding goppa codes. Inf. Control **27**(1), 87–99 (1975)

[16] Mills, W.H.: Continued fractions and linear recurrences. Math. Comput. **29**(129), 173–180 (1975)

[17] Reeds, A., Sloane, N.J.A.: Shift-register synthesis (mod m). SIAM J. Comput. **14**(3), 505–513 (1985)

[18] Ward, M.: Arithmetical properties of sequences in rings. Ann. of Math. **39**(1), 210–219 (1938)

[19] Fan, S.Q., Han, W.B.: Random properties of the highest level sequences of primitive sequences over $\mathbb{Z}/(2^e)$. IEEE Trans. Inf. Theory **49**(6), 1553–1557 (2003)

[20] Sole, P., Zinoviev, D.: The most significant bit of maximum length sequences over $\mathbb{Z}/(2^l)$: autocorrelation and imbalance. IEEE Trans. Inf. Theory **50**(8), 1844–1846 (2004)

[21] Zheng, Q.-X.: On the distinctness of sequences derived from primitive sequences over integer residue rings. Doctoral dissertation, Information Engineering University, Zhengzhou, China (2013)

[22] Hachenberger, D., Jungnickel, D.: Topics in Galois Fields. Springer, Cham, Switzerland (2020)

[23] Huang, M.Q.: Analysis and cryptologic evaluation of primitive sequences over an integer residue ring. Doctoral dissertation, USTC, Hefei, China (1988)

[24] Kuzmin, A.S., Nechaev, A.A.: Linear recurring sequences over galois ring. Algebra Log. **34**, 87–100 (1995)

[25] Zhu, X.Y.: Some results on injective mappings of primitive sequences modulo prime powers. Doctoral dissertation, Information Engineering University, Zhengzhou, China (2004)

[26] Kuzmin, A.S., Marchalko, G.B., Nechaev, A.A.: Reconstruction of a linear recurrence over a primary residue ring. Memoires Discrete Math. **12**, 155–194 (2009)

[27] Kuzmin, A.S., Nechaev, A.A.: Reconstruction of a linear recurrence of maximal period over a galois ring from its highest coordinate sequence. Discrete Math. Appl. **21**(2), 145–178 (2011)

[28] Yang, J.B.: Reconstructing truncated sequences derived from primitive sequences over inter residue rings. Mas.thesis, Information Engineering University, Zhengzhou, China (2017)

[29] Sun, H.-Y., Zhu, X.-Y., Zheng, Q.-X.: Predicting truncated multiple recursive generators with unknown parameters. Des. Codes Cryptogr. **88**(6), 1083–1102 (2020)

[30] Yu, H.-B., Zheng, Q.-X.: Predicting truncated galois linear feedback shift registers. IEEE Trans. Inf. Theory **70**(12), 9179–9194 (2024)

[31] Coppersmith, D.: Finding a small root of a univariate modular equation. In: Advances in Cryptology–EUROCRYPT 1996, pp. 155–165. Springer, Berlin, Heidelberg (1996)

[32] Ward, M.: The arithmetical theory of linear recurring series. Trans. Amer. Math. Soc. **35**(3), 600–628 (1933)

[33] Nguyen, P.Q.: Hermite's constant and lattice algorithms. In: Nguyen, P.Q., Vallée, B. (eds.) The LLL Algorithm, pp. 19–69. Springer, Berlin, Heidelberg (2010)

[34] Ajtai, M.: Generating random lattices according to the invariant distribution (2006). Draft of March

[35] Gama, N., Nguyen, P.Q.: Predicting lattice reduction. In: Smart, N. (ed.) Advances in Cryptology–EUROCRYPT 2008, pp. 31–51. Springer, Berlin, Heidelberg (2008)

[36] Nguyen, P.Q., Stehlé, D.: LLL on the average. In: Hess, F., Pauli, S., Pohst, M. (eds.) Algorithmic Number Theory, pp. 238–256. Springer, Berlin, Heidelberg (2006)

[37] Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mathematische Annalen **261**(4), 515–534 (1982)

[38] Schnorr, C.P., Euchner, M.: Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Math. Program. **66**(2), 181–199 (1994)

[39] Kannan, R.: Improved algorithms for integer programming and related lattice problems. In: Proceedings of the 15th Annual ACM Symposium on Theory of Computing, pp. 193–206. Association for Computing Machinery, New York, NY, USA (1983)

[40] Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Res. **12**(3), 415–440 (1987)

[41] Ajtai, M., Kumar, R., Sivakumar, D.: A sieve algorithm for the shortest lattice vector problem. In: Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, pp. 601–610. Association for Computing Machinery, New York, NY, USA (2001)

[42] William, C.B.: Matrices over Commutative Rings. Marcel Dekker, New York (1993)

[43] Becker, A., Gama, N., Joux, A.: Speeding-up lattice sieving without increasing the memory, using sub-quadratic nearest neighbor search. Preprint at http://eprint.iacr.org/2015/522 (2015)

[44] Shoup, V.: Number Theory C++ Library (NTL), version 11.4.3. http://www.shoup.net/ntl/

[45] Albrecht, M.R., Ducas, L., Herold, G., Kirshanova, E., Postlethwaite, E.W., Stevens, M.: The general sieve kernel and new records in lattice reduction. In: Ishai, Y., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2019, pp. 717–746. Springer, Cham (2019)