Predicting one class of truncated matrix congruential generators with unknown parameters

Changcun Wang[1] and Zhaopeng Dai[1*]

[1]School of Mathematics and Statistics, Qingdao University, 308 Ningxia Road, Qingdao, 266071, Shandong, China.

*Corresponding author(s). E-mail(s): dzpeng@amss.ac.cn;

Abstract

Matrix congruential generators is an important class of pseudorandom number generators. In this paper we show how to predict a class of truncated matrix congruential generators with unknown parameters. Given a few truncated digits of high-order bits output by a matrix congruential generator, we give a method based on lattice reduction to recover the parameters and the initial state of the generator.

Keywords: Matrix congruential generators, Lattice reduction, Truncated digits

1 Introduction

Pseudo-random numbers play a crucial role in various fields such as computer science, information security, statistics, and simulations. In cryptography, pseudo-random numbers are extensively used to enhance the security of cryptographic algorithms. These algorithms require a significant amount of random data to generate encryption keys, initialization vectors, and other cryptographic components. Pseudo-random numbers help increase the complexity of cryptographic schemes, making it challenging for adversaries to predict or manipulate the encryption process.

In 1949, Lehmer [1] proposed an important class of PRNGs, linear congruence generator(LCG). An LCG output a sequence in the form of

$$a_{i+1} = ba_i + c \pmod{m}, \quad i \geq 0,$$

where b, c are constants and a_0 is initial state over $\mathbb{Z}_m$, the period of LCG is limited by modulus m, and its maximum period is m when $c \neq 0$ and $m-1$ when $c = 0$. Moreover, LCG on ellipitic curves have been proposed by Hallgren [2] in 1994 and extensively studied, see [3, 4]. In the case that the parameters b, c, m are unknown, Plumstead [5] showed that the fixed LCG is predictability if all bits of the sequence are available. Knuth [6] considered the problems arising when only truncated high-order bits of the sequence generated by an LCG are known. In the case that the modulus $m = 2^k$ is known and the parameters b, c are unknown, Knuth presented an attack to reconstruct the parameters and the seed of the LCG. Subsequently, the study is further extended, due to the work of Frieze [7, 8], Boyar [9, 10], Stern[11, 12].

Due to its simplicity, high efficiency, and clear theoretical properties, LCG has become popular. However, LCG are not recommended by today's standards because it has relatively short period bounded by the modulus m and inadequate uniformity in dimensions higher than 1. Therefore, to increase the period and , the multiple recursive generator(MRG) is proposed. An MRGs of order n defined as

$$a_{i+n} = c_{n-1}a_{i+n-1} + \cdots + c_1 a_{i+1} + c_0 a_i \pmod{m}, \quad i \geq 0$$

where $c_0, \ldots, c_{n-1}$ are not all zero and $a_0, \ldots, a_{n-1}$ is the initial state over $\mathbb{Z}_m$. If the modulus m is a prime number, say $m = p$, then the maximum period of the MRG is $p^n - 1$. For a general number m, let $m = p_1^{k_1} \cdots p_s^{k_s}$ be the canonical factorization of m, the maximum period of the MRG is $\mathrm{lcm}(p_1^{k_1-1}(p_1^n-1), p_2^{k_2-1}(p_2^n-1), \ldots, p_s^{k_s-1}(p_s^n-1))$. An MRG 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 $\mathbb{Z}_m$.

MRGs has a longer period and better randomness than LCGs. Recently, there are numerous research results on the predictability of MRG. With all parameters of a MRG are known, Yang et al. [13] gave a method based on Coppersmith's method to recover the initial by the truncated sequences. In the case that all MRG parameters are unknown, Sun et al. [14] extended Stern's algorithm to predict truncated MRG. They constructed proper lattices to recover the modulus m, the coefficients $c_0, \ldots, c_{n-1}$, and the initial state $a_0, \ldots, a_{n-1}$, respectively. Yu et al. [15] proposed a new method to recover the coefficients and initial state, their method is more efficient and requires less data compared to Sun et al.'s approach. At the same time, Yu et al. [16] successfully resolved the predicting problems of MRGs with constant c type. There are also many previous studies results on MRG that can be referred to [17–22].

The popularity of PRNGs is also accompanied by the popularity of pseudorandom vector generators(PRVGs). Random vectors are becoming more important because of the trend toward parallelization in scientific computing. Typical applications of random vectors arise in parallelized probabilistic algorithms, in parallel Monte Carlo and simulation methods, and in multivariate statistics. Matrix congruential generators (MCGs) as an important type of PRVGs. An MCG output a sequence $\underline{u} = (u_i)_{i \geq 0} =$

$(u_{i,0},\ldots,u_{i,k-1})_{i\geq 0}$ of the form

$$(u_{i+1,0},\cdots,u_{i+1,k-1}) = (u_{i,0},\cdots,u_{i,k-1})A \pmod m, i \geq 0.$$

where the vector $(u_{0,0},\cdots,u_{0,k-1}) \in \mathbb{Z}_m^k$ is the initial state and $A \in \mathbb{Z}_m^{k\times k}$, gcd(det$A$,$m$)=1, There are many results [23–25] about the period of MCGs, but there are currently few results on its predictability of MCGs.

While previous work in [23–25] have predominantly explored characteristics like the period of MCGs, few results have been available about the predictability of MCGs. This paper focuses on predicting truncated k-order MCGs, where both the $k\times k$ matrix A and modulus m are unknown. The approach involves constructing an appropriate lattice to recover the modulus m, followed by the retrieval of the matrix A and the initial state. Similar to Stern's algorithm the success of the method in this paper relies on two lattice properties that cannot be proved mathematically. However, both the explanation of Stern based on a heuristic hypothesis and our repeated experiments confirm the correctness of the two properties, see Sect.3.3. for details. We have perform a lot of experiments to test our method.

The other parts of this article are organized as follows. Section 2 introduces the basic knowledge of lattice. Section 3 proposes our method to predict truncated MCGs. Section 4 provide an experimental example. Section 5 proposes issues for further study.

2 Preliminary

In this section,we briefly review the basic knowledge of lattice reduction.
All lattice take row vectors as a lattice basic in this paper.

Definition 1. Let $n \geq l \geq 1$ and let $v_1, v_2, \ldots, v_l$ be a set of l linearly independent vectors in $\mathbb{R}^n$. The l-dimensional lattice spanned by $v_1, v_2, \cdots, v_l$ in n-dimensional Euclidean space is defined to be

$$L = \mathbb{Z}v_1 + \mathbb{Z}v_2 + \cdots + \mathbb{Z}v_l = \{\sum_{i=1}^{l} k_i v_i \mid k_1, k_2, \ldots, k_l \in \mathbb{Z}\}.$$

For $i = 1, 2, \ldots, l$, let $v_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,n})$ and form the $l \times n$ matrix $V = (v_{i,j})$. Let V^T be the transpose of V. Then the determinant of lattice L is defined by

$$\det(\mathrm{L}) = \sqrt{\det(VV^T)}.$$

Definition 2. Let L be a lattice. The ith minimum of L denoted by $\lambda_i(L)$ is defined to be the radius of the smallest zero-centered ball containing at least i linearly independent lattice vectors. In particular, $\lambda_1(L)$ is the norm of the shortest non-zero vector in L.

Let L represent an n-dimensional lattice with determinant $\det(L)$, and consider C as a convex measurable subset of $\mathbb{R}^n$ with volume $\mathrm{vol}(C)$. The Gaussian heuristic posits that the approximate count of points in $L \cap C$ is proportional to $\mathrm{vol}(C)/\det(L)$. Nguyen [26] highlights that, while the Gaussian heuristic is generally considered an approximation, it can be rigorously proven for random lattices. Moreover, it is deemed unlikely to be significantly erroneous by an exponential factor, unless the lattice possesses exceptional characteristics. The value of $\lambda_1(L)$, presumably a lattice parameter, can be estimated using the following methodology.

lemma 1. [26] Let L be an n-dimensional lattice. If the Gaussian heuristic holds for L, then

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

Denote by $\|\mathbf{u}\|$ the Euclidean norm of vector $\mathbf{u}$. The following two types of difficult problems are commonly concerned in lattice.

(1) The shortest vector problem(SVP): Given a lattice $L \subset \mathbb{R}^n$, find a non-zero vector $\mathbf{u} \in L$, such that $\|\mathbf{u}\| \leq \|\mathbf{v}\|$ for any non-zero vector $\mathbf{v} \in L$.
(2)The closest vector problem(CVP): Given a lattice $L \subset \mathbb{R}^n$ and a vector $\mathbf{w} \in \mathbb{R}^n$, find a vector $\mathbf{u} \in L$, such that $\|\mathbf{u\text{-}w}\| \leq \|\mathbf{v\text{-}w}\|$ for any vector $\mathbf{v} \in L$.

The initial and most renowned algorithm for solving the Approximate Shortest Vector Problem (Approximate-SVP) is the LLL algorithm, proposed by Lenstra et al. [27]in 1982. The most practically efficient algorithm is the BKZ algorithm, introduced by Schnorr and Euchner [28] in 1994. In 2011, Chen and Nguyen [29] put forth a new version of the BKZ algorithm. For a comprehensive understanding of lattice reduction algorithms, refer to [30].

Consider an n-dimensional lattice denoted by L, and let $\frac{1}{4} < \delta < 1$ be the reduction parameter of the LLL algorithm. When provided with a basis of L as input, the LLL algorithm produces a δ-reduced basis of L in a number of steps bounded by a polynomial in the size of the input. The δ-reduced basis satisfies the following properties.

lemma 2. [27] Let L be an n-dimensional lattice. Assume δ is the reduction parameter of the LLL algorithm and $v_1, v_2, \ldots, v_n$ is a δ-reduced basis of L output by the LLL algorithm. Let $\lambda_i(L)$ be the ith minimum of L. Then

(1) $\|v_i\| \leq \rho^{(n-1)/2} \lambda_i(L), 1 \leq i \leq n,$
(2) $\|v_1\| \leq \rho^{(n-1)/4} \det(L)^{1/n},$

where $\rho = \frac{4}{4\delta - 1}$.

Remark 1. The value of the reduction parameter is usually set to $\delta = \frac{3}{4}$, at the time $\rho = 2$.

3 Main results

In this paper, the k-order ($k \geq 2$) of MCG is denoted as

$$(u_{i+1,0}, u_{i+1,1}, \ldots, u_{i+1,k-1}) \equiv (u_{i,0}, u_{i,1}, \ldots, u_{i,k-1})A \pmod{m}, \tag{1}$$

where m be an integer greater than 1 and

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{pmatrix} \in \mathbb{Z}_m^{k \times k}.$$

To guarantee that the sequence $\underline{u} = (u_i)_{i \geq 0} = (u_{i,0}, u_{i,1}, \ldots, u_{i,k-1})_{i \geq 0}$ is purely periodic, we assume that $\gcd(\det A, m) = 1$.

Define $l = \lceil \log m \rceil$ as the number of bits in $m - 1$. Assuming that $y_{i,j}$ encompasses a fraction α of the high-order bits of $u_{i,j}$ and $z_{i,j}$ comprises the remaining low-order bits, we can express this as follows:

$$u_{i,j} = 2^{\beta l} y_{i,j} + z_{i,j}, \tag{2}$$

where $\beta = 1 - \alpha$ and $2^{\beta k} \in \mathbb{Z}$.

We study the following predictability problem: Given the first N truncated digits

$$\{(y_{0,0}, y_{0,1}, \ldots, y_{0,k-1}), \ldots, (y_{N-1,0}, y_{N-1,1}, \ldots, y_{N-1,k-1})\}.$$

We want to predict the rest of the sequence accurately. Obviously, if we can recover the modulus m, the matrix A, and the initial state $(u_{0,0}, u_{0,1}, \ldots, u_{0,k-1})$, then we can predict the rest of the sequence. As a result, to solve the predictability problem we just need to solve the following problem.

Problem Let $y_{i,j}$ be a proportion α of the high-order bits of $u_{i,j}$. Given the first N truncated digits $((y_{0,0}, y_{0,1}, \ldots, y_{0,k-1}), \ldots, (y_{N-1,0}, y_{N-1,1}, \ldots, y_{N-1,k-1}))$, the problem is to recover the modulus m, the matrix A, and the initial state $(u_{0,0}, u_{0,1}, \ldots, u_{0,k-1})$.

By (1), we can get

$$\begin{cases} u_{i+1,0} \equiv a_{11}u_{i,0} + a_{21}u_{i,1} + \cdots + a_{k1}u_{i,k-1} \pmod{m} \\ u_{i+1,1} \equiv a_{12}u_{i,0} + a_{22}u_{i,1} + \cdots + a_{k2}u_{i,k-1} \pmod{m} \\ \qquad\qquad \vdots \\ u_{i+1,k-1} \equiv a_{1k}u_{i,0} + a_{2k}u_{i,1} + \cdots + a_{kk}u_{i,k-1} \pmod{m} \end{cases}. \tag{3}$$

Then by the recurrence relation of the sequence $\underline{u}$, we have

$$(u_{i+j,0}, u_{i+j,1}, \ldots, u_{i+j,k-1}) \equiv (u_{i,0}, u_{i,1}, \ldots, u_{i,k-1})A^j \pmod{m}.$$

Let the elements of A^j is

$$\begin{pmatrix} q_{j,0} & q_{j,k} & \cdots & q_{j,k(k-1)} \\ q_{j,1} & q_{j,k+1} & \cdots & q_{j,k(k-1)+1} \\ \vdots & \vdots & \ddots & \vdots \\ q_{j,k-1} & q_{j,2k-1} & \cdots & q_{j,k(k-1)+k-1} \end{pmatrix}.$$

For $s \in \{0, \ldots, k-1\}$, then we have

$$u_{i+j,s} \equiv \sum_{v=0}^{k-1} q_{j,sk+v} u_{i,v} \pmod{m}. \tag{4}$$

In particular, we have $q_{1,0} = a_{11}, \ldots, q_{1,k(k-1)+k-1} = a_{kk}$.

3.1 Searching linear dependence relations

lemma 3. [11] Assume $Y_0, Y_1, \cdots, Y_{r-1}$ is a family of vectors with integer coordinates in the t-dimensional space, with $t < r$. Let M denote an upper bound for the absolute values of all coordinates of the various Y_is. There exist integers $\zeta_0, \zeta_1, \cdots, \zeta_{r-1}$ such that

$$\sum_{i=0}^{r-1} \zeta_i Y_i = 0,$$

where $\max| \zeta_i | \leq B$, and B is given by

$$logB = t\frac{logM + logr + 1}{r - t}.$$

Let $r = ks$, $s \in \mathbb{Z}$. Choose positive r, t satisfying $r > t > n$ and construct vectors $Y_i \in \mathbb{Z}^t$

$$\begin{gathered} Y_{ki} = (y_{i,0}, y_{i+1,0}, \ldots, y_{i+t-1,0}), \\ Y_{ki+1} = (y_{i,1}, y_{i+1,1}, \ldots, y_{i+t-1,1}), \\ Y_{ki+2} = (y_{i,2}, y_{i+1,2}, \ldots, y_{i+t-1,2}), \\ \vdots \\ Y_{ki+k-1} = (y_{i,k-1}, y_{i+1,k-1}, \ldots, y_{i+t-1,k-1}), \ (i = 0, 1, \ldots, s-1). \end{gathered} \tag{5}$$

According to the Lemma 3, there exist integer coefficients $\eta_0, \ldots, \eta_{r-1}$, such that

$$\sum_{i=0}^{r-1} \eta_i Y_i = 0, \tag{6}$$

where $\mid \eta_i \mid \leq B$, and B satisfying

$$logB = t\frac{log(2^{\alpha l}) + logr + 1}{r-t} = t\frac{\alpha l + logr + 1}{r-t}.$$

The next step of the process is to discuss in which case the equation

$$U \triangleq \sum_{i=0}^{r-1} \eta_i A_i = 0 \tag{7}$$

holds, where

$$\begin{gathered} A_{ki} = (u_{i,0}, u_{i+1,0}, \ldots, u_{i+t-1,0}), \\ A_{ki+1} = (u_{i,1}, u_{i+1,1}, \ldots, u_{i+t-1,1}), \\ A_{ki+2} = (u_{i,2}, u_{i+1,2}, \ldots, u_{i+t-1,2}), \\ \vdots \\ A_{ki+k-1} = (u_{i,k-1}, u_{i+1,k-1}, \ldots, u_{i+t-1,k-1}),\ (i = 0, 1, \ldots, s-1). \end{gathered}$$

As a preparation for recovering the unknown parameters, we still need to find a set of linear relations for

$$\underline{u}^N = ((u_{0,0}, u_{0,1}, \ldots, u_{0,k-1}), \ldots, (u_{N-1,0}, u_{N-1,1}, \ldots, u_{N-1,k-1})).$$

However, we do not need to find the vectors $\eta = (\eta_0, \eta_1, \ldots, \eta_{r-1})$ that validate both (6) and (7). Actually, validating (7) is our ultimate goal while (6) can be neglected. Observe that if (7) holds, then

$$0 = \sum_{i=0}^{r-1} \eta_i A_i = 2^{\beta l} \sum_{i=0}^{r-1} \eta_i Y_i + \sum_{i=0}^{r-1} \eta_i Z_i,$$

where

$$\begin{gathered} Z_{ki} = (z_{i,0}, z_{i+1,0}, \ldots, z_{i+t-1,0}), \\ Z_{ki+1} = (z_{i,1}, z_{i+1,1}, \ldots, z_{i+t-1,1}), \\ Z_{ki+2} = (z_{i,2}, z_{i+1,2}, \ldots, z_{i+t-1,2}), \\ \vdots \\ Z_{ki+k-1} = (z_{i,k-1}, z_{i+1,k-1}, \ldots, z_{i+t-1,k-1}),\ (i = 0, 1, \ldots, s-1), \end{gathered}$$

and so

$$\sum_{i=0}^{r-1}\eta_i Y_i = -2^{-\beta l}\sum_{i=0}^{r-1}\eta_i Z_i.$$

Since $|z_{i,j}| < 2^{\beta l}$, it follows that

$$\|\sum_{i=0}^{r-1}\eta_i Y_i\| \le \sum_{i=0}^{r-1}\| - 2^{-\beta l}\eta_i Z_i\| \le \sqrt{t}\sum_{i=0}^{r-1}|\eta_i|,$$

This means that $\sum_{i=0}^{r-1}\eta_i Y_i$ is a short vector, which inspires us to search a short vector of the lattice rather than a vector that validates (6).

Construct the following lattice

$$L' = \begin{pmatrix} Y_0 & 1 & 0 & \cdots & 0 \\ Y_1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Y_{r-1} & 0 & 0 & \cdots & 1 \end{pmatrix},$$

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

$$\mathbb{W} = (w_0, \ldots, w_{t-1}, w_t, \ldots, w_{r+t-1}) = (\sum_{i=0}^{r-1}\eta_i Y_i, \eta_0, \eta_1, \ldots, \eta_{r-1}) \in L'.$$

If η is a vector that validates (7), then $\sum_{i=0}^{r-1}\eta_i Y_i$ is a short vector, and so is $\mathbb{W}$. Therefore, the desire vectors $\eta = (\eta_0, \eta_1, \ldots, \eta_{r-1})$ that validate (7) can be obtained by applying LLL to the lattice L'.

Remark 2. If the value of r and t are chosen correctly, a single call to lattice reduction algorithm can output multiple sets of $\eta_0, \eta_1, \ldots, \eta_{r-1}$ that satisfy(7).

3.2 Constructing congruence equations

We can appropriately determine the values of r and t through the aforementioned process. we can get

$$\begin{cases} \eta_0 u_{0,0} + \cdots + \eta_{k-1}u_{0,k-1} + \eta_k u_{1,0} + \cdots + \eta_{2k-1}u_{1,k-1} + \cdots + \eta_{r-1}u_{s-1,k-1} = 0 \\ \eta_0 u_{1,0} + \cdots + \eta_{k-1}u_{1,k-1} + \eta_k u_{2,0} + \cdots + \eta_{2k-1}u_{2,k-1} + \cdots + \eta_{r-1}u_{s,k-1} = 0 \\ \vdots \\ \eta_0 u_{t-1,0} + \cdots + \eta_{k-1}u_{t-1,k-1} + \eta_k u_{t,0} + \cdots + \eta_{2k-1}u_{t,k-1} + \cdots + \eta_{r-1}u_{t+s-2,k-1} = 0 \end{cases} \tag{8}$$

by means of (4). Therefore, any $u_{i,j}, (i \geq 1, j = 0, 1, \ldots, k-1)$ can be written in $u_{0,0}, u_{0,1}, \ldots, u_{0,k-1}$ linear combination.

By (8), we can get

$$\begin{pmatrix} u_{0,0} & u_{0,1} & \cdots & u_{0,k-1} \\ u_{1,0} & u_{1,1} & \cdots & u_{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{t-1,0} & u_{t-1,1} & \cdots & u_{t-1,k-1} \end{pmatrix} \begin{pmatrix} t_0 \\ t_1 \\ \vdots \\ t_{k-1} \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \bmod m, \tag{9}$$

where

$$t_i = \eta_i + \sum_{j=k}^{2k-1} \eta_j q_{1,k(j-k)+i} + \sum_{j=2k}^{3k-1} \eta_j q_{2,k(j-2k)+i} + \cdots + \sum_{j=(s-1)k}^{sk-1} \eta_j q_{s-1,k(j-(s-1)k)+i},$$

for $0 \leq i \leq k-1$.

Denote by T the coefficient matrix of (9). If (9) has only the trivial solution over $\mathbb{Z}_m^k$, then we can get

$$\begin{cases} t_0 \equiv 0 \mod m \\ t_1 \equiv 0 \mod m \\ \quad \vdots \\ t_{k-1} \equiv 0 \mod m \end{cases}. \tag{10}$$

So, the modulus m and the matrix A can be recovered by (10) which is to be seen in Sects. 3.3 and 3.4. Therefore, it is necessary to discuss the condition that (9) has only trivial solution.

Denote by T the coefficient matrix in (9). Obviously, if select a k-order submatrix from T randomly, and the probability that the determinant of the submatrix is coprime to m is $6/\pi^2$. So the probability that (9) has only the trivial solution is about $C_t^k \times 6/\pi^2$, then the probability is almost equal to 1.

3.3 Recovering the modulus m

In this section we will prove that for any $i \in \{0, 1, \ldots, k-1\}$, the modulus m can be recovered by $t_i \equiv 0 \bmod m$ in (10).

From (10), we know that $t_i \equiv 0 \bmod m$, i.e.,

$$t_i = \eta_i + \sum_{j=k}^{2k-1} \eta_j q_{1,k(j-k)+i} + \sum_{j=2k}^{3k-1} \eta_j q_{2,k(j-2k)+i} + \cdots + \sum_{j=(s-1)k}^{sk-1} \eta_j q_{s-1,k(j-(s-1)k)+i} \equiv 0 \bmod m,$$

then there exists an integer u_i making the following equation holds.

$$\eta_i = u_i m - (\sum_{j=k}^{2k-1} \eta_j q_{1,k(j-k)+i} + \sum_{j=2k}^{3k-1} \eta_j q_{2,k(j-2k)+i} + \cdots + \sum_{j=(s-1)k}^{sk-1} \eta_j q_{s-1,k(j-(s-1)k)+i}).$$

Constructing the following lattice $L(t_i)$ whose dimension is $r-k+1$

$$L(t_i) = \begin{pmatrix} m & 0 & 0 & \cdots & 0 \\ -q_{1,i} & 1 & 0 & \cdots & 0 \\ -q_{1,k+i} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -q_{s-1,k(k-1)+i} & 0 & 0 & \cdots & 1 \end{pmatrix}.$$

The determinant of $L(t_i)$ is m. Denoted by ω_j the j row vector of $L(t_i)$ for $0 \le j \le r-k$. Then we have

$$\eta(i) = u_i\omega_0 + \sum_{j=k}^{r-1} \eta_j \omega_{j-k+1} \in L(t_i),$$

where $\eta(i) = (\eta_i, \eta_k, \eta_{k+1}, \ldots, \eta_{r-1})$.

From Sections 3.1 and 3.2, it is evident that with a finite truncated sequence of $s+t-1$ vectors represented as

$$y_l(s+t-1) = ((y_{l,0}, \ldots, y_{l,k-1}), \ldots, (y_{l+s+t-2,0}, \ldots, y_{l+s+t-2,k-1}))$$

for some $l \ge 0$, it becomes possible to identify at least one $\eta(i)$ such that $t_i \equiv 0 \bmod m$. If we are acquainted with d distinct finite truncated sequences (where some y_l may be shared by distinct finite truncated sequences), the conclusion can be drawn that we can obtain at least d such vectors

$$\eta(i)^{(j)} = (\eta_i^{(j)}, \eta_k^{(j)}, \eta_{k+1}^{(j)}, \ldots, \eta_{r-1}^{(j)}) \quad 0 \le j \le d-1$$

satisfying $t_i \equiv 0 \bmod m$.

Let

$$M(t_i) = \begin{pmatrix} {\eta_i}^{(0)} & {\eta_k}^{(0)} & {\eta_{k+1}}^{(0)} & \cdots & {\eta_{r-1}}^{(0)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ {\eta_i}^{(d-1)} & {\eta_k}^{(d-1)} & {\eta_{k+1}}^{(d-1)} & \cdots & {\eta_{r-1}}^{(d-1)} \end{pmatrix}.$$

The row vectors of $M(t_i)$ are not necessary linearly independent. So we reduce $M(t_i)$ through LLL algorithm and let $L(t_i)^*$ be the lattice generated by the output reduced basis. Since

$$\eta(i)^{(j)} = (\eta_i^{(j)}, \eta_k^{(j)}, \eta_{k+1}^{(j)}, \ldots, \eta_{r-1}^{(j)}) \in L(t_i)$$

for any $j \in \{0, \ldots, d-1\}$. Obviously, $L(t_i)^*$ is a sublattice of $L(t_i)$.

Similarly to case of LCGs, the lattice $L(t_i)^*$ also has the following two properties which were found by Stern [11, 12]:

(1) As d grows, the dimension of $L(t_i)^*$ very quickly increases to full rank $r-k+1$;

(2) Once the dimension of $L(t_i)^*$ reaches full rank, the determinant of $L(t_i)^*$ is a multiple of m, and very quickly decreases to m.

We cannot prove the above two properties mathematically. Stern encountered the same problem when studying the predictability of truncated LCGs. He gave a heuristic analysis of the correctness of the above two properties under a heuristic hypothesis that the output polynomials are random elements of the lattice L. Here we make similar heuristic hypothesis that each $\eta(i)^{(j)} = (\eta_i^{(j)}, \eta_k^{(j)}, \eta_{k+1}^{(j)}, \ldots, \eta_{r-1}^{(j)})$ is a random vector of $L(t_i)$, then we can explain the correctness of the above two properties in the same way as Stern. Please refer to [11] for detail analysis. Our repeated experiments also confirm the correctness of the above two properties.

If we use the method in Sect. 3.1 several times, we can get a couple of such vectors $\eta(i) = (\eta_i, \eta_k, \eta_{k+1}, \ldots, \eta_{r-1})$ satisfying $t_i \equiv 0 \bmod m$. Then we construct the matrix $M(t_i)$ and reduce $M(t_i)$ through lattice reduction algorithm. By the two properties above, we can see that when enough vectors are added, the determinant of the lattice $L(t_i)^*$ reaches m. Thus, we can recover the modulus m by calculating the determinant of $L(t_i)^*$. It should be noted that to recover m, we can choose any $i \in \{0, \ldots, k-1\}$ and calculate the determinant of $L(t_i)^*$. However to recover the mattrix A, the determinant of $L(t_i)^*$ needs to reach m. Thus, to recover the matrix A, the determinant of $L(t_0)^*, \ldots, L(t_{k-1})^*$ all need to reach m. The detailed process to recover the matrix A is described in the next subsection.

3.4 Recovering the matrix A

In this section we will show that the value of $a_{ij}, i, j \in \{1, \ldots, k\}$ can be recovered by (10).

Building on the discussion in Section 3.3, it is apparent that $L(t_i)^*$ forms a sublattice of $L(t_i)$. Upon achieving full rank, the lattice $L(t_i)^*$ ensures that its determinant becomes a multiple of m. As new vectors are added, the determinant of $L(t_i)^*$ rapidly decreases, eventually reaching m. When the determinant of $L(t_i)^*$ reaches m, we have

$$L(t_i)^* = L(t_i).$$

Let

$$L(t_i)^* = \begin{pmatrix} \tau_{0,0} & \tau_{0,1} & \tau_{0,2} & \cdots & \tau_{0,r-k} \\ \tau_{1,0} & \tau_{1,1} & \tau_{1,2} & \cdots & \tau_{1,r-k} \\ \tau_{2,0} & \tau_{2,1} & \tau_{2,2} & \cdots & \tau_{2,r-k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tau_{r-k,0} & \tau_{r-k,1} & \tau_{r-k,2} & \cdots & \tau_{r-k,r-k} \end{pmatrix}.$$

Choose a large constant $K > m2^{(r-k)/2}$. Multiply the jth column vector of $L(t_i)^*$ by K for $2 \le j \le r-k$ and denoted by G the new lattice, that is

$$G = \begin{pmatrix} \tau_{0,0} & \tau_{0,1} & K\tau_{0,2} & \cdots & K\tau_{0,r-k} \\ \tau_{1,0} & \tau_{1,1} & K\tau_{1,2} & \cdots & K\tau_{1,r-k} \\ \tau_{2,0} & \tau_{2,1} & K\tau_{2,2} & \cdots & K\tau_{2,r-k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tau_{r-k,0} & \tau_{r-k,1} & K\tau_{r-k,2} & \cdots & K\tau_{r-k,r-k} \end{pmatrix}.$$

Reduce G through LLL algorithm and denoted by $H = (h_{u,v})_{0 \le u,v \le r-k}$ the reduced matrix. Let H_j be the jth row vector of H for $0 \le j \le r-k$. Then we have the following conclusion.

Theorem 1. [14] If $L(t_i)^* = L(t_i)$, then the lattice generated by H_0 and H_1 is equal to the lattice generated by $(m, 0, 0, \ldots, 0)$ and $(-q_{1,i}, 1, 0, \ldots, 0)$.

In particularly, by Theorem 1, we can get the value of $q_{1,j}$, for $j \in \{k, \ldots, (k+1)(k-1)\}$.

3.5 Recovering the initial state $(u_{0,0}, u_{0,1}, \ldots, u_{0,k-1})$

We have known the modulus m and $a_{ij}, i, j \in \{1, \ldots, k\}$. Our objective now is to recover the initial state $(u_{0,0}, u_{0,1}, \ldots, u_{0,k-1})$. Given that $(y_{0,0}, y_{0,1}, \ldots, y_{0,k-1})$ is known, by (2), recovering $(u_{0,0}, u_{0,1}, \ldots, u_{0,k-1})$ is essentially equivalent to recovering $(z_{0,0}, z_{0,1}, \ldots, z_{0,k-1})$. Yang [13] has successfully addressed this challenge by transforming it into the task of finding small integer solutions to systems of linear congruences. He then applied the method proposed by Frieze [7] to solve this problem. Let's briefly outline this method.

Suppose the first $d(d > n)$ consecutive truncated digits are given, we need to recover $z_{0,0}, \ldots, z_{0,k-1}$. By (2) and (4), for $i \in \{0, \ldots, k-1\}$, we can get

$$(\sum_{v=0}^{k-1} q_{j,ik+v} z_{0,v} - z_{j,i}) \equiv 2^{\beta l}(y_{j,i} - \sum_{v=0}^{k-1} q_{j,ik+v} y_{0,v}) \pmod m, \ 1 \le j \le d-1. \quad (11)$$

It can be seen that to recover the unknown $(z_{0,0},\ldots,z_{0,k-1})$, it suffices to solve (11). Let

$$L(m,d+k-1)=\begin{pmatrix} m & 0 & \cdots & 0 & 0 & 0 & \cdots & 0\\ 0 & m & \cdots & 0 & 0 & 0 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & m & 0 & 0 & \cdots & 0\\ q_{1,ik} & q_{1,ik+1} & \cdots & q_{1,ik+k-1} & -1 & 0 & \cdots & 0\\ q_{2,ik} & q_{2,ik+1} & \cdots & q_{2,ik+k-1} & 0 & -1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots\\ q_{d-1,ik} & q_{d-1,ik+1} & \cdots & q_{d-1,ik+k-1} & 0 & 0 & \cdots & -1 \end{pmatrix}.$$

Let $z=(z_{0,0},\ldots,z_{0,k-1},z_{1,i},\ldots,z_{d-1,i})$. Denoted by λ_{d+k-1} the $(d+k-1)$th minimum of the lattice $L(m,d+k-1)$. The solution of (11) has been solved in [7].

lemma 4. [7] The system of modular Eqs.(11) has at most one solution $z\in\mathbb{Z}^{d+k-1}$ satisfying

$$\|z\|\le m\lambda_{d+k-1}^{-1}2^{-((d+k-1)-1)/2-1}.$$

Furthermore, there is a polynomial time algorithm that either finds z or proves that no such z exists.

Since $|z_{i,j}|<2^{\beta l}$, let $H=d+k-1$, then we have $\|z\|<\sqrt{H}2^{\beta l}$, by Lemma 4, if

$$\sqrt{H}2^{\beta l}\le m\lambda_H^{-1}2^{-(H-1)/2-1}$$

then

$$\alpha k\ge\frac{logH}{2}+log\lambda_H+\frac{H+1}{2}$$

then we can recover the initial state $(u_{0,0},\ldots,u_{0,k-1})$ by solving (11).

4 Example

We have performed a lot of experiments to verify the correctness and effectiveness of our method. All the experiments are performed on our personal computer(Windows 11, AMD R7-5800H, 3.20GHz). The lattice reduction algorithm used is the LLL algorithm in the Maple and Magma.

We choose the parameters $u_{0,0},u_{0,1},u_{0,2},a_{ij},i,j\in\{1,2,3\}$ randomly over $\mathbb{Z}_m$ and generate the sequence $\underline{u}=(u_i)_{i\ge0}=(u_{i,0},u_{i,1},u_{i,2})_{i\ge0}$.

$$u_0=(333,511,76),A=\begin{pmatrix}1&2&3\\3&4&5\\7&8&1\end{pmatrix},m=555.$$

In this experiment, suppose that the modulus m, the matrix A, and the initial state $u_{0,0},u_{0,1},u_{0,2}$ are all unknown, and let $y_{i,j}$ be the 9 high-order bits of $u_{i,j}$, we

want to recover the modulus m, the parameters $a_{ij}, i, j \in \{1, 2, 3\}$ and the initial state $u_{0,0}, u_{0,1}, u_{0,2}$. The experimental process includes the following three steps.

Step 1 We choose $r = 30$, $t = 9$ and then repeatedly use the method in Sect. 3.1 to get a couple of such vectors $(\eta_0, \eta_1, \ldots, \eta_{29})$ satisfying $t_i \equiv 0 \bmod m$ for $i = 0, 1, 2$.

Step 2 For $i \in \{0, 1, 2\}$, we recover the matrix A. The modulus m can be recovered during the process of recovering the parameters a_{ij}. When recovering a_{ij}, by the method in Sects. 3.3 and 3.4, we construct the matrix $M(t_i)$ using the vectors $\eta_i = (\eta_i, \eta_3, \ldots, \eta_{29})$ obtained in step 1 and reduce $M(t_i)$ through the LLL algorithm in Magma. As the number of added row vectors grows, the dimension of the lattice $L(t_i)^*$ very quickly reaches full rank 28, and the determinant of $L(t_i)^*$ is a multiple of m, and very quickly decreases to m. When the determinant of $L(t_i)^*$ is m, we have $L(t_i)^* = L(t_i)$, and then we can recover the matrix A using the method in Sect. 3.4.

Step 3 When the modulus m and the matrix A have been recovered, we use the method in Sect. 3.5 to recover the initial state $u_{0,0}, u_{0,1}, u_{0,2}$. In the experiments, we can use the first $d = 8$ consecutive truncated digits $\{(y_{0,0}, y_{0,1}, y_{0,2}), \ldots, (y_{5,0}, y_{5,1}, y_{5,2}))\}$ to recover the initial state $u_{0,0}, u_{0,1}, u_{0,2}$.

5 Conclusion and Future Work

In this paper, the predictability of truncated MCGs with unknown parameters is studied. Concretely, given sufficiently many of the most significant bits of several consecutive outputs of one class of MCGs, we present a method based on lattice reduction algorithm to recover all of the parameters and the initial state of the MCG.

The limitations of our approach primarily revolve around one aspect: We lack a valid heuristic analysis for the choices of r and t. Open questions whether there is a valid heuristic analysis for the choices of r and t which are worthy of further study.

References

[1] Lehmer, D.H.: Mathematical models in large-scale computing units. Ann. Comput. Lab.(Harvard University) **26**, 141–146 (1951)

[2] Hallgren, S.: Linear congruential generators over elliptic curves. Carnegie Mellon University (1994)

[3] Hess, F., Shparlinski, I.E.: On the linear complexity and multidimensional distribution of congruential generators over elliptic curves. Designs Codes and Cryptography **35**(1), 111–117 (2005)

[4] Mérai, L.: Predicting the elliptic curve congruential generator. Applicable Algebra in Engineering, Communication and Computing **28**, 193–203 (2017)

[5] Plumstead, J.B.: Inferring a sequence generated by a linear congruence. In: 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982), pp. 153–159 (1982). IEEE

[6] Knuth, D.E.: The Art of Computer Programming: Seminumerical Algorithms. Addison-Wesley, Reading, WA (1969)

[7] Frieze, A.M., Hastad, J., Kannan, R., Lagarias, J.C., Shamir, A.: Reconstructing truncated integer variables satisfying linear congruences. SIAM Journal on Computing **17**(2), 262–280 (1988)

[8] Frieze, A.M., Kannan, R., Lagarias, J.C.: Linear congruential generators do not produce random sequences. In: 25th Annual Symposium onFoundations of Computer Science, 1984., pp. 480–484 (1984). IEEE

[9] Boyar, J.: Inferring sequences produced by pseudo-random number generators. Journal of the ACM (JACM) **36**(1), 129–141 (1989)

[10] Boyar, J.: Inferring sequences produced by a linear congruential generator missing low-order bits. Journal of Cryptology **1**(3), 177–184 (1989)

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

[12] Stern, J.: Secret linear congruential generators are not cryptographically secure. In: 28th Annual Symposium on Foundations of Computer Science (sfcs 1987), pp. 421–426 (1987). IEEE

[13] Yang, J.: Reconstructing truncated sequences derived from primitive sequences over inter residue rings. PLA Information Engineering University, Zhengzhou (2017)

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

[15] 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., *et al.*: An improved method for predicting truncated multiple recursive generators with unknown parameters. Designs, Codes and Cryptography **91**(5), 1713–1736 (2023)

[16] YU, H., ZHENG, Q.: A lattice-based method for recovering the unknown parameters of truncated multiple recursive generators with constant. Chinese Journal of Electronics **33**, 1–10 (2023)

[17] Contini, S., Shparlinski, I.E.: On stern's attack against secret truncated linear congruential generators. In: Australasian Conference on Information Security and

Privacy, pp. 52–60 (2005). Springer

[18] Huang, M.-Q.: Analysis and cryptologic evaluation of primitive sequences over an integer residue ring. Ph. D. dissertation, Graduate School, Univ. Sci. Technol. China, Beijing, China (1988)

[19] Kuzmin, A.S., Nechaev, A.A.: Reconstruction of a linear recurrence of maximal period over a galois ring from its highest coordinate sequence (2011)

[20] Kuzmin, A., Marshalko, G., Nechaev, A.: Reconstruction of a linear recurrence over a primary residue ring. Memoires in Discrete Mathematics **12**, 155–194 (2009)

[21] Kuzmin, A., Nechaev, A.: Linear recurring sequences over galois rings. Algebra and Logic **34**(2), 87–100 (1995)

[22] Zhu, X.: Some results on injective mappings of primitive sequences modulo prime powers. PLA Information Engineering University, Zhengzhou (2004)

[23] Eichenauer-Herrmann, J., Grothe, H., Lehn, J.: On the period length of pseudorandom vector sequences generated by matrix generators. Mathematics of Computation **52**(185), 145–148 (1989)

[24] Grothe, H.: Matrix generators for pseudo-random vector generation. Statistische Hefte **28**(1), 233–238 (1987)

[25] Niederreiter, H.: A pseudorandom vector generator based on finite field arithmetic. Math. Japonica **31**(5), 759–774 (1986)

[26] Nguyen, P.Q., Stehlé, D.: LLL on the average. In: Algorithmic Number Theory: 7th International Symposium, ANTS-VII, Berlin, Germany, July 23-28, 2006. Proceedings 7, pp. 238–256 (2006). Springer

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

[28] Schnorr, C.-P., Euchner, M.: Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Mathematical programming **66**, 181–199 (1994)

[29] Chen, Y., Nguyen, P.Q.: Bkz 2.0: Better lattice security estimates. In: International Conference on the Theory and Application of Cryptology and Information Security, pp. 1–20 (2011). Springer

[30] Nguyen, P.Q., Vallée, B.: The LLL Algorithm: Survey and Applications. Springer, Berlin (2009)