JOACHIM VON ZUR GATHEN & IGOR E. SHPARLINSKI (2004). Predicting Subset Sum Pseudorandom Generators. In *Selected Areas in Cryptography*, HELENA HANDSCHUH & M. ANWAR HASAN, editors, volume 3357 of *Lecture Notes in Computer Science*, 241–251. Springer-Verlag, Berlin, Heidelberg. ISBN 3-540-24327-5. ISSN 0302-9743. URL https://dx.doi.org/10.1007/978-3-540-30564-4_17.

Predicting Subset Sum Pseudorandom Generators

Joachim von zur Gathen[1] and Igor E. Shparlinski[2]

[1] Fakultät für Elektrotechnik, Informatik und Mathematik
Universität Paderborn
33095 Paderborn, Germany
gathen@upb.de
http://www-math.upb.de/~aggathen

[2] Department of Computing, Macquarie University,
NSW 2109, Australia
igor@comp.mq.edu.au
http://www.comp.mq.edu.au/~igor

Abstract. We consider the subset sum pseudorandom generator, introduced by Rueppel and Massey in 1985 and given by a linear recurrence sequence $u_0, u_1, \ldots$ of order n over $\mathbb{Z}_2$, and weights $w = (w_0, \ldots, w_{n-1}) \in R^n$ for some ring R. The rings $R = \mathbb{Z}_m$ are of particular interest. The ith value produced by this generator is $\sum_{0 \le j < n} u_{i+j} w_j$. It is also recommended to discard about $\log n$ least significant bits of the result before using this sequence. We present several attacks on this generator (with and without the truncation), some of which are rigorously proven while others are heuristic. They work when one "half" of the secret is given, either the control sequence u_j or the weights w_j. Our attacks do not mean that the generator is insecure, but that one has to be careful in evaluating its security parameters.

1 Introduction

Let $u_0, u_1, \ldots$ be a linear recurrence sequence of order n over the field $\mathbb{Z}_2$ of two elements; see [11, Chapter 8]. We may also consider each u_j as an integer, namely 0 or 1, and multiply by it an element z of an arbitrary ring R, so that $u_j z \in R$.

We consider the following *subset sum* generator of pseudorandom elements. Given an n-dimensional vector $w = (w_0, \ldots, w_{n-1}) \in R^n$, its output is the sequence

$$v_i = \sum_{0 \le j < n} u_{i+j} w_j, \qquad \text{for } i = 0, 1, \ldots, \tag{1}$$

of elements of R. A popular choice is to take $R = \mathbb{Z}_m$, the residue ring modulo $m \ge 2$. Particularly recommended is the choice $m = 2^k$ with some integer k, in particular, it is natural to choose $k = n$; see [13, Section 6.3.2]. We also consider the case where $m = p$ is prime. We call (u_j) the *control sequence* and $w_0, \dots, w_{n-1}$ the *weights*.

This generator, which is also known as *knapsack generator*, was introduced in [18] and studied in [16], see also [13, Section 6.3.2] and [17, Section 3.7.9]. The generation algorithm is multiplication-free and involves only Boolean operations, integer additions and one modular reduction; in the case $R = \mathbb{Z}_{2^k}$, the reduction modulo $m = 2^k$ is essentially for free in the binary representation. Thus it presents a very attractive alternative to pseudorandom number generators based on Boolean functions. On the other hand, its close relation to the subset sum problem could make it cryptographically strong and suitable for using in stream ciphers.

For cryptographic applications, it is usually recommended to use a linear recurrence sequence of maximal period $2^n - 1$, however here we consider more general settings.

The linear complexity and distribution of this generator have been studied in [5, 16, 17] and have turned out to be rather attractive. We also remark that [13, page 220] notes that no weaknesses of this generator have been reported in the literature. The present paper presents some weaknesses. We do not, however, consider these as lethal.

We study predictability properties of the subset sum generator and show that its security is smaller than has been assumed previously, but presumably still large enough, with appropriate parameters. In the simplest cases our attacks are based on linear algebra. In more practical settings we use lattice algorithms, namely algorithms for the *shortest vector problem* which essentially go back to the seminal paper of Lenstra, Lenstra and Lovász [10]. Thus our results add one more example to the substantial list of cryptographic constructions which have been successfully attacked by such algorithms, see [12, 14, 15].

We note that our results resemble those about predictability of various recursive pseudorandom number generators; see [1–4, 7–9] and references therein.

In general, if $R = \mathbb{Z}_m$, the whole generator is defined by about $n(2 + \log m)$ bits. Indeed, one needs n bits to describe the characteristic polynomial of the control linear recurrence sequence (u_j), n bits for its initial values, and about $n \log m$ bits to describe the weight vector w. Thus a brute force search through the space of all possible parameters takes about $(4m)^n$ steps.

In our attacks we use polynomial time and assume that some partial information about the generator is known. However, one might as well "guess" this information; in this formulation our attacks lead to a substantial reduction of the cost of brute force search. In the same vein, some of our results deal with the generator before truncation, but one may simply "guess" the truncated parts and then apply our attacks. For example, as we have mentioned, it is suggested to discard about $\log n$ bits of each output v_i, see [13, Section 6.3.2]. Usually our attacks need only $O(n)$ consecutive outputs, thus the total number of guesses of the discarded bits is $2^{O(n \log n)}$ which, for typically recommended values of m near 2^n, is substantially smaller than $(4m)^n \asymp 2^{n^2}$. On the other hand, in some cases our attacks, empowered by lattice basis reduction algorithms, apply to truncated outputs directly.

The upshot is that when n is large enough and both controls and weights are kept secret, we still consider the generator to be secure. But we can mount an exhaustive search attack with cost 2^{2n}, so that it is not clear in how far larger values of m make the generator much more secure than $m = 2$ or $m = 3$. (Our short vector attack becomes more expensive with growing m, but only by a polynomial factor.) A variant of the truncation technique can be applied to $m = 2$, by discarding output values, say according to some bit derived from the controls.

2 Attacks with known control sequence

We first consider the case when the linear recurrence sequence (u_j) is known. It is equivalent to know the characteristic polynomial and n initial values, or just $2n$ initial values; the characteristic polynomial can then be computed by the Berlekamp-Massey algorithm (see, for example, [6, Section 12.3]).

It is useful to express (1) in terms of the power series

$$h_u = \sum_{i \geq 0} u_i x^i, \quad h_v = \sum_{i \geq 0} v_i x^i, \quad h_w = \sum_{0 \leq i < n} w_{n-i-1} x^i$$

in $R[[x]]$. We show that the power series $h_u \cdot h_w$ and $x^{n-1} h_v$ agree at all but the small-order coefficients.

Lemma 1 *Let $r = h_u \cdot h_w \operatorname{rem} x^{n-1}$ be the remainder of $h_u \cdot h_w$ on division by x^{n-1}. Then*

$$h_u \cdot h_w - r = x^{n-1} h_v. \tag{2}$$

Proof. We have

$$\begin{aligned} x^{n-1} h_v &= \sum_{i \geq 0} v_i x^{i+n-1} = \sum_{i \geq 0} \sum_{0 \leq j < n} u_{i+j} w_j x^{i+n-1} \\ &= \sum_{\substack{i \geq 0 \\ 0 \leq j < n}} u_{i+j} x^{i+j} \cdot w_j x^{n-j-1} = \sum_{\substack{k+l \geq n-1 \\ 0 \leq l < n}} u_k x^k \cdot w_{n-l-1} x^l. \end{aligned}$$

The bijective correspondence

$$(i, j) = (k + l - n + 1, n - l - 1) \leftrightarrow (k, l) = (i + j, n - j - 1)$$

is responsible for the last equation. The condition $i \geq 0$ means that $k+l \geq n-1$. Thus the coefficient of the terms of degree at least $n-1$ in the products $x^{n-1} h_v$ and $h_u \cdot h_w$ coincide. ∎

When we take the weights as unknowns, the equations (1), or, equivalently, (2) yield a Hankel system of linear equations with the matrix

$$H = (u_{i+j})_{0 \leq i,j < n}. \tag{3}$$

In a finite prime field, the Hankel matrix (3) is not guaranteed to be nonsingular. Our attack works by building up matrices of maximal rank from lines of the Hankel matrix (3). Accordingly, we may have to use n arbitrary outputs, not necessarily the first ones. More precisely, we consider algorithms that for $i = 0, 1, \ldots$ either output v_i or query v_i. The following result shows that we can do with few queries.

Theorem 2. *Over a finite field $R = \mathbb{F}_q$ of q elements, given a control sequence (u_j) of order n, there is a deterministic algorithm to compute the sequence v_i for $i = 0, 1, \ldots$, in polynomial time per element, making no more than n queries in total.*

Proof. The algorithm builds up $l \times n$ matrices U_l consisting of rows

$$r_i = (u_i, u_{i+1}, \ldots, u_{i+n-1}) \in \mathbb{F}_q^n$$

for growing values of l, up to n. The matrix U_l has rank l over $\mathbb{F}_q$. We also store the values v_i for the rows r_i that appear in U_l.

We start with $U_0 = I_0 = \emptyset$, and consider $i = 0, 1, \ldots$. If r_i is not linearly dependent over $\mathbb{F}_q$ on the rows of the current U_l (this is the case in the first step, where $i = 0$, unless $r_0 = 0$), then we set $I_{l+1} = I_l \cup \{i\}$ and add the row r_i to U_l to obtain U_{l+1}, of rank $l + 1$. We also query and store v_i.

Otherwise we can write

$$r_i = \sum_{k \in I_l} c_k r_k$$

as a linear combination of the rows r_k of U_l, with $k \in I_l$ and coefficients $c_k \in \mathbb{F}_q$. Then we output

$$\begin{aligned}\sum_{k \in I_l} c_k v_k &= \sum_{k \in I_l} c_k \sum_{0 \leq j < n} u_{k+j} w_j \\ &= \sum_{0 \leq j < n} w_j \sum_{k \in I_l} c_k u_{k+j} \sum_{0 \leq j < n} w_j \, u_{i+j} = v_i.\end{aligned}$$

We have to make at most n queries for values v_i, since once we have n linearly independent (over $\mathbb{F}_q$) rows r_i, then we can actually compute the weight vector w, and predict correctly ever after. □

In characteristic 2, the Hankel matrix (3) is guaranteed to be nonsingular, and the algorithm simplifies as follows.

Corollary 3 *Given an integer $k \geq 1$, a control sequence (u_j) of order n over $\mathbb{Z}_2$, and n consecutive outputs v_i for $0 \leq i < n$ over $R = \mathbb{Z}_{2^k}$, one can find the unknown weight vector $w \in R^n$ in deterministic polynomial time.*

Proof. Because (u_j) is of order n in $\mathbb{F}_2$, the integer Hankel matrix (3) is nonsingular modulo 2, see [11, Section 8.6], and hence also modulo 2^k. $\square$

The algorithm also works over rings $R = \mathbb{Z}_m$ with squarefree $m \geq 2$, by using a "lazy" variant of Gaussian elimination. Here, whenever an element is to be inverted, one calculates its greatest common divisor with the current moduli (which initially is just m). If the greatest common divisor is nontrivial, one obtains a factorization of the modulus, and continues with the factors separately as new moduli.

3 Attacks with known weights

Here we consider the dual question, where the linear recurrence sequence (u_n) is unknown but the vector of weights $w = (w_1, \ldots, w_r) \in \mathbb{Z}_m^n$ is given. When we are given only a single output of the generator, then this is a subset sum problem and NP-complete. However having several consecutive outputs allows us to mount an efficient short vector attack.

We start with characteristic 2 and present our results in the case when the characteristic polynomial of the control linear recurrence sequence (u_j) is irreducible, which includes the most interesting cases of such sequences. In the general case one can obtain similar results, which however hold only for almost all weights rather than for all $w \in \mathbb{Z}_{2^k}^n$.

Theorem 4. *Given an integer $k \geq 1$, the weights $w = (w_0, \ldots, w_{n-1}) \in R^n$ over $R = \mathbb{Z}_{2^k}$, and $2n$ consecutive outputs v_i for $0 \leq i < 2n$, not all even, one can find the controls $u = (u_0, \ldots, u_{2n-1}) \in \mathbb{Z}_2^{2n}$ in deterministic polynomial time, provided that the (unknown) characteristic polynomial of degree n over $\mathbb{Z}_2$ of the control linear recurrence sequence (u_j) is irreducible.*

Proof. The reduction of the sequence (v_i) modulo 2 satisfies the same linear recurrent relation as the control sequence (u_j). By assumption, this reduction is not identical to zero modulo 2. We use the Berlekamp–Massey algorithm, see [6, Chapter 7] or [11, Section 8.6], to recover the characteristic polynomial

$$f = \sum_{0 \leq i \leq n} f_i x^i \in \mathbb{Z}_2[x]$$

of this sequence, so that

$$\sum_{0 \leq i \leq n} f_i u_{k+i} = 0$$

for all $k \geq 0$. The first n equations in (2) plus the $n-1$ equations for the control values $u_n, \ldots, u_{2n-2}$ lead to the following system of $2n-1$ linear equations in the $2n-1$ unknowns $u_0, \ldots, u_{2n-2}$:

$$\begin{pmatrix} w_0 & w_1 & \cdots & w_{n-1} & 0 & \cdots & 0 \\ 0 & w_0 & \cdots & w_{n-2} & w_{n-1} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \cdots & \vdots \\ 0 & \cdots & \cdots & w_0 & w_1 & \cdots & w_{n-1} \\ f_0 & f_1 & \cdots & f_{n-1} & f_n & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & f_1 & f_2 & \cdots & f_n \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \\ u_n \\ u_{n+1} \\ \vdots \\ u_{2n-2} \end{pmatrix} = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

We denote the matrix of the above system of equations by $A \in R^{(2n-1)\times(2n-1)}$, and observe that A is the (transpose of the) Sylvester matrix of the two polynomials f and

$$w = \sum_{0 \le i < n} w_i x^i \in \mathbb{Z}_2[x].$$

The outputs are not all even, and hence also the weights, and thus w is nonzero of degree less than n. Since f is irreducible of degree n, we have $\gcd(w, f) = 1$ and hence A is nonsingular. Thus we can solve the system for $u_0, \ldots, u_{2n-2}$. $\square$

When the characteristic polynomial f is not irreducible, the characteristic polynomial g of $v_0, v_1, \ldots$ is a divisor of f. If the weights are chosen at random, we expect $g = f$ to hold with high probability; see [6, Section 12.4]. Furthermore, for random w the condition $\gcd(w, f) = 1$ (so that A is nonsingular) holds with probability

$$\Phi(f)/2^n = \prod_{j=1}^{s} (1 - 2^{-d_j}),$$

where $d_1, \ldots, d_s$ are the degrees of the distinct irreducible factors of f. Thus, $\Phi(f)$ is the polynomial analogue of Euler's ϕ function. Using the fact that the number of irreducible polynomials of degree d over $\mathbb{Z}_2[x]$ is $2^d/d + O(2^{d/2})$, one can show that this probability is also reasonably large.

We now consider the case of an arbitrary modulus m. Given k consecutive values v_i for $0 \le i < k$, we may define the lattice L as the set of all integer solutions $\mathbf{x} = (x_{-1}, x_0, x_1, \ldots x_{k+n-1}) \in \mathbb{Z}^{k+n+1}$ of the system of congruences

$$\sum_{0 \le j < n} x_{i+j} w_j + v_i x_{-1} \equiv 0 \bmod m \qquad \text{for } 0 \le i < k.$$

By (1), it contains a very short vector $\mathbf{u} = (-1, u_0, \ldots, u_{k+n-1})$ with Euclidean norm at most $\|\mathbf{u}\| \le (k+n+1)^{1/2}$. Standard heuristic arguments, as in [15, Section 3.4], imply that the discriminant of L is likely to be $D_k = m^k$.

On the other hand, also standard heuristic arguments suggest that if $\|\mathbf{u}\|$ is substantially smaller than $D_k^{1/(k+n)}$, then any other nonzero vector $\mathbf{x} \in L$ of length substantially smaller than $D_k^{1/(k+n)}$ is likely to be proportional to $\mathbf{u}$. Thus applying any of the algorithms for the shortest vector problem, we can hope to recover $\mathbf{x}$; see [12, 14, 15] for outlines of recent progress in this area since the celebrated result of Lenstra, Lenstra and Lovász [10].

If $k \geq n+1$, then the vector $\mathbf{x}$ gives us the values u_j for $0 \leq j < 2n$. By the Berlekamp-Massey algorithm, one can find the characteristic polynomial of the linear recurrence sequence (u_j) over $\mathbb{Z}_2$ and thus continue to generate the sequence (v_i).

Furthermore, with $k = n+1$ we expect

$$D_k^{1/(k+n)} \sim m^{k/(k+n)} \geq m^{1/2}$$

which is much larger than $(k+n)^{1/2} = (2n+1)^{1/2}$ for all practically interesting situations.

We now consider the case when some bits of the output are discarded before exhibiting the remaining bits. Although our approach works in more general settings, here consider only the case which is outlined in [13, Section 6.5.6]. In this case $t = 2^n - 1$, $m = 2^n$ and $\ell = \lceil \log n \rceil$ bits of each value v_i get discarded before the rest is output, that is, only the "truncated" values $\lfloor v_i/2^\ell \rfloor$ are known.

Given k consecutive values $\lfloor v_i/2^\ell \rfloor$ for $0 \leq i < k$, we define a lattice $\mathcal{L}_k$ as the set of all integer solutions $\mathbf{x} = (x_{-1}, x_0, x_1, \ldots x_{k+n-1}, y_0, \ldots, y_{k-1}) \in \mathbb{Z}^{2k+n-1}$ of the system of congruences

$$\sum_{0 \leq j < n} x_{i+j} w_j + 2^\ell \left\lfloor v_i/2^\ell \right\rfloor x_{-1} + y_i \equiv 0 \bmod m \qquad \text{for } 0 \leq i < k.$$

Again we observe that the discriminant of $\mathcal{L}_k$ is likely to be $D_k = m^k$. We also see that it contains a very short vector

$$z = (-1, u_0, \ldots, u_{k+n}, z_0, \ldots, z_{k-1}),$$

where $z_i = 2^\ell \lfloor v_i/2^\ell \rfloor - v_i$ for $0 \leq i < k$, whose Euclidean norm satisfies

$$\|z\| \leq \left(k + n + k\left(2^\ell - 1\right)\right)^{1/2} = \left(k 2^\ell + n\right)^{1/2} \leq (2kn + n)^{1/2}.$$

We see that if $k = \lceil \log n \rceil$, then $\|z\| \leq (2n \log n + O(n))^{1/2}$, while

$$D_k^{1/(2k+n)} = 2^{kn/(2k+n)} \geq n^{n/(2k+n)} = n^{1+o(1)}, \qquad \text{for } n \to \infty,$$

is much larger. Certainly increasing the value of k increases the chances that z is much shorter than any other non-parallel vectors in $\mathcal{L}_k$ and thus can be found by an appropriate algorithm for the shortest vector problem.

We have conducted several tests for values of n up to $n = 100$ with m a 100-bit prime. In each case, the short vector computed provided correctly the control sequence. In all cases, there have not been other "smallish" short vectors in the lattice. These experiments confirm that the algorithm always finds the control sequence, at least for sufficiently large problems of cryptographically interesting sizes.

4 Final remarks

As noted before, our results do not rule out the possibility of successfully using the subset sum generator for cryptographic purposes. They merely imply that the security is less than its naive estimate based on counting unknown bits in the parameters defining the generator. Thus with a careful choice of these parameters this generator can turn out to be very useful and reliable. Unfortunately, at the present time it is hard to give any practical recommendations on the specific choice of the parameters. This requires more extensive numerical experiments using more computational power than has been used in our tests. Certainly this issue deserves more attention.

It would be very interesting to obtain rigorous proofs for the heuristic attacks described in this paper. Besides being of theoretic value, this may also give further insight on the structure and thus security of the subset sum generator.

Finally, the construction itself may be applied to some other rings R, not necessary residue rings. This may produce new and more robust sequences.

References

1. S. R. Blackburn, D. Gomez-Perez, J. Gutierrez and I. E. Shparlinski, 'Predicting the inversive generator', *Lect. Notes in Comp. Sci.*, Springer-Verlag, Berlin, **2898** (2003), 264–275.
2. S. R. Blackburn, D. Gomez-Perez, J. Gutierrez and I. E. Shparlinski, 'Predicting nonlinear pseudorandom number generators', *Math. Comp.*, (to appear).
3. S. R. Blackburn, D. Gomez-Perez, J. Gutierrez and I. E. Shparlinski, 'Reconstructing noisy polynomial evaluation in residue rings', *J. Algorithms*, (to appear).
4. E. F. Brickell and A. M. Odlyzko, 'Cryptoanalysis: A survey of recent results', *Contemp. Cryptology*, IEEE Press, NY, 1992, 501–540.
5. A. Conflitti and I. E. Shparlinski, 'On the multidimensional distribution of the subset sum generator of pseudorandom numbers', *Math. Comp.*, **73** (2004), 1005–1011.
6. J. von zur Gathen and J. Gerhard, *Modern computer algebra*, Cambridge University Press, Cambridge, 2003.
7. A. Joux and J. Stern, 'Lattice reduction: A toolbox for the cryptanalyst', *J. Cryptology*, **11** (1998), 161–185.
8. H. Krawczyk, 'How to predict congruential generators', *J. Algorithms*, **13** (1992), 527–545.
9. J. C. Lagarias, 'Pseudorandom number generators in cryptography and number theory', *Proc. Symp. in Appl. Math.*, Amer. Math. Soc., Providence, RI, **42** (1990), 115–143.
10. A. K. Lenstra, H. W. Lenstra and L. Lovász, 'Factoring polynomials with rational coefficients', *Mathematische Annalen*, **261** (1982), 515–534.
11. R. Lidl and H. Niederreiter, *Finite fields*, Cambridge University Press, Cambridge, 1997.
12. D. Micciancio and S. Goldwasser, *Complexity of lattice problems*, Kluwer Acad. Publ., 2002.
13. A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, *Handbook of applied cryptography*, CRC Press, Boca Raton, FL, 1996.

14. P. Q. Nguyen and J. Stern, 'Lattice reduction in cryptology: An update', *Lect. Notes in Comp. Sci.*, Springer-Verlag, Berlin, **1838** (2000), 85–112.
15. P. Q. Nguyen and J. Stern, 'The two faces of lattices in cryptology', *Lect. Notes in Comp. Sci.*, Springer-Verlag, Berlin, **2146** (2001), 146–180.
16. R. A. Rueppel, *Analysis and design of stream ciphers*, Springer-Verlag, Berlin, 1986.
17. R. A. Rueppel, 'Stream ciphers', *Contemporary cryptology: The science of information integrity*, IEEE Press, NY, 1992, 65–134.
18. R. A. Rueppel and J. L. Massey, 'Knapsack as a nonlinear function', *IEEE Intern. Symp. of Inform. Theory*, IEEE Press, NY, 1985, 46.