# ON BINARY SEQUENCES FROM RECURSIONS "modulo $2^e$" MADE NON-LINEAR BY THE BIT-BY-BIT "XOR" FUNCTION

W.G.CHAMBERS

Department of Electronic and Electrical Engineering, King's College London, Strand, London WC2R 2LS, UK

Z.D.DAI

Mathematics Department, Royal Holloway and Bedford New College (University of London), Egham, Surrey, UK. On leave from the Graduate School, Academia Sinica, Beijing, People's Republic of China; supported by SERC grant GR/F72727

**Abstract**: We consider binary sequences obtained by choosing the the most significant bit of each element in a sequence obtained from a feedback shift register of length $n$ operating over the ring $Z/2^e$, that is with arithmetic carried out modulo $2^e$. The feedback has been made non-linear by using the bit-by-bit exclusive-or function as well as the linear operation of addition. This should increase the cryptologic strength without greatly increasing the computing overheads. The periods and linear equivalences are discussed. Provided certain conditions are met it is easy to check that the period achieves its maximal value.

**1) Introduction**: For $e$ a positive integer let $Z/2^e$ denote the ring of integers $\{0, \ldots, 2^e - 1\}$ with addition, subtraction, and multiplication carried out "mod $2^e$". (In other words if the result of the arithmetic operation gives a value outside the ring, then it is brought back by adding or subtracting a suitable multiple of $2^e$.) We start by considering linear recursions of the form

$$a_{t+n} = \sum_{j=0}^{n-1} c_j a_{t+j} \bmod 2^e \quad \text{for} \quad t = 0, 1, 2, \ldots \tag{1}$$

with $a_0, a_1, \ldots, a_{n-1}$ specifying the initial conditions. At least one of these values is odd. Here $a_t$ and $c_j$ belong to $Z/2^e$. Here and throughout this presentation for $a$, $p$ integer with $p > 1$ we define $a \bmod p$ as that integer $\in \{0, 1, \ldots, p-1\}$ obtained by adding (subtracting) the appropriate integer multiple of $p$ to (from) $a$. (The operator "mod" will be taken as binding more loosely than arithmetic operators such as "+", but more tightly than "=", so that for instance $a = b + c \bmod p$ means $a = ((b + c) \bmod p)$.) We may then derive a binary output by picking the most significant bit of each $a_t$. This can provide a convenient way of generating pseudo-random binary sequences on general-purpose microprocessors, in which case $e$ would typically be the number of bits in a computer-word. It should be a particularly convenient technique on some digital signal processors which have high-speed facilities for multiply-accumulation. Now as regards the cryptologic security: The generator is a linear congruential generator and cryptanalytic techniques are available at least when the coefficients are known [1]. These techniques suppose that the output sequence is truncated down to the several most significant bits and it seems unlikely that they are practicable when only the most

D.W. Davies (Ed.): Advances in Cryptology - EUROCRYPT '91, LNCS 547, pp. 200-204, 1991.

significant bit of each output is available. Nonetheless it seems reasonable to increase the security. One way is to make the coefficients $F_j$ key-dependent and to use some kind of non-linear output filtering. We discuss instead what should be an even better possibility, the use of the bit-by-bit exclusive-or function as a source of non-linearity *inside* the recursion. Non-linearity should be cryptographically more effective here than when used in filtering the output. Moreover the exclusive-or function is a fast operation readily available on most microprocessors and digital signal processors and it is non-linear in $Z/2^e$ for $e > 1$.

The exclusive-or function (denoted by $\oplus$) is defined as follows: For $a$, $b$ non-negative integers with $a = \sum_{k=0}^{\infty} a_k 2^k$ and $b = \sum_{k=0}^{\infty} b_k 2^k$ (with $a_k$, $b_k \in \{0,1\}$) we set

$$a \oplus b = \sum_{k=0}^{\infty} ((a_k + b_k) \bmod 2) 2^k.$$

To make the discussion definite we consider a recursion of the form

$$a_t = (\sum_{j=0}^{n-1} c_j a_{t+j} \bmod 2^e) \oplus (\sum_{j=0}^{n-1} d_j a_{t+j} \bmod 2^e) \quad \text{for} \quad t = 0, 1, 2, \ldots \tag{2}$$

Here at least one of the $d_j$ is non-zero, and at least two of the $c_j$ are odd to guarantee a proper non-linear carry. Moreover the *base polynomial*

$$h(x) = x^n + \sum_{j=0}^{n-1} (c_j + d_j) x^j \bmod 2 \tag{3}$$

is a primitive binary polynomial. (The operation "mod $p$" applied to a polynomial expression signifies that the coefficients are to be evaluated "mod $p$".)

The period of the generator (1) has been investigated long ago by Ward [2]; more recent work concerns the period and upper and lower bounds on the linear equivalence of the binary sequences produced by taking the bit of a given order of significance from each $a_t$ [3], [4], [5], [6]. Let $a_t$ have the binary decomposition $a_t = \sum_{i-0}^{e-1} a_{t,i} 2^i$ with $a_{t,i} \in \{0,1\}$. Denote the sequence $\{a_0, a_1, a_2, \ldots\}$ by $\alpha$ and the binary sequences $\{a_{0,i}, a_{1,i}, a_{2,i}, \ldots\}$ by $\alpha_i$. We quote the following results: If the base polynomial $h(x)$ is a primitive binary polynomial of degree $n$, the possible periods of $\alpha_i$ are $2^k(2^n - 1)$ with $k = 0, 1, \ldots, i$. Moreover for any $i$ satisfying $1 < i < e$ and with $h(x) = x^n + f(x) \bmod 2$ a specified primitive polynomial and with $\alpha_0$ not identically zero, all but a fraction $2/2^n$ of the possible connection polynomials $f(x) = \sum_{i=0}^{n-1} c_i x^i$ give $\alpha_i$ the maximal period $(2^n - 1)2^i$. From the practical point of view this means that provided we keep $n$ reasonably large, say $> 40$, there is not much risk of obtaining a short-period sequence. There are also "fast" tests for checking that the period is maximal [3], [4], [7].

Unless we have some understanding about the periods of sequences generated by (2) there is always the worry that we may obtain a dangerously short period for some initial conditions and/or choices of connection polynomial. This is·the standard objection to the use of non-linear recursions which otherwise would seem to be an attractive proposition in cryptography. The results stated in the last paragraph were derived using the linearity of (1). What can be said about the periods of sequences generated by (2) depends very much on the $d_j$. If any of the $d_j$ are odd we can say very little definitely apart from the fact that the period is a factor of $(2^n - 1)2^{e-1}$. Much more definite conclusions apply if all the $d_j$ are even, and for the rest of this presentation we concentrate on this case. There may be a price to pay for this increased understanding in that the generator may not be quite as strong cryptologically as in the general case.

We shall present (for the case when all the $d_j$ are even) a "fast" test to check whether the period is maximal, and the probability of the period's not being maximal if the coefficients are chosen at random. We shall also present expressions for characteristic polynomials from which upper bounds to the linear equivalence may be found; computer results demonstrate that these are tight in simple cases.

**2) Test for maximal period**: Tests for maximal period in the outputs from (1) have been discussed in [4] and [7]. We now describe a test for the generator (2), in the case when all the $d_j$ are even.

Let any quantity that takes only the values 0 and 1 be called *binary-valued*. Similarly designate a polynomial (sequence) as *binary-valued* if all its coefficients (elements) are binary-valued. Put $f(x) = \sum_{i=0}^{n-1} c_i x^i$, $g(x) = \sum_{i-0}^{n-1} d_i x^i$. Then set

$$f(x) = \sum_{k=0}^{e-1} f_k(x) 2^k, \quad g(x) = \sum_{k=0}^{e-1} g_k(x) 2^k, \quad \alpha = \sum_{k=0}^{e-1} \alpha_k 2^k,$$

with $f_k(x)$ and $g_k(x)$ binary-valued polynomials, and with the $\alpha_k$ binary-valued sequences.

Let

$$g_0(x) = 0$$

and let the base polynomial (3)

$$h(x) = \sum_0^n h_i x^i = x^n + f(x) \bmod 2$$

be a primitive binary polynomial, with binary valued coefficients $h_i$. Moreover let at least one of the initial settings $a_0, a_1, \ldots, a_{n-1}$ be odd. These are the conditions for the theorem. Next let $\theta$ denote a root of $h(x)$ (in GF$(2^n)$), and set

$$\Phi = \sum_{0 \le i < j < n} h_i h_j \theta^{i+j}, \quad \kappa = \sum_{i=1,3,5,\ldots}^{n} h_i \theta^i.$$

Also define $v_1$, $v_2$, $v_3$ by

$$v_1 = \Phi + (f_1(\theta))^2 + (g_1(\theta))^2,$$

$$v_2 = v_1(v_1^2 + \kappa^4) + (g_1(\theta))^2 \theta^{2n} \kappa^2,$$

$$v_3 = \kappa^4 v_2 + (g_1(\theta))^4 \theta^{4n} v_1.$$

All these quantities are in GF$(2^n)$. Then we have

**Theorem 1:** With the conditions just stated, the period of $\alpha_1$ has the maximal value $2(2^n - 1)$ if and only if $v_1 \neq 0$, the period of $\alpha_2$ has the maximal value $4(2^n - 1)$ if and only if $v_1v_2 \neq 0$, and for $k \geq 3$ the period of $\alpha_k$ has the maximal value $2^k(2^n-1)$ if and only if $v_1v_2v_3 \neq 0$.

We make the following remarks:

1) The period does not depend on $f_i(x)$ or $g_i(x)$ for $i \geq 2$.

2) If $g_1(x) = 0$ (as well as $g_0(x) = 0$), then the results depend only on $f_1(x)$ and we may equivalently use alternative methods [4], [7] for testing the period of the generator (1) obtained by setting $g(x) = 0$.

3) An alternative is to test the period of $\alpha_3$ for shortness, but this needs of the order of $2^n$ steps.

4) Computations in GF($2^n$) are equivalent to computations with binary polynomials "modulo" the base polynomial $h(x)$. The root $\theta$ is represented by the polynomial $x$. The coefficients $g_{ij}$ in the binary polynomial $\sum_{i=0}^{n-1} g_{ij}x^j$ obtained as the remainder when $x^i$ is divided by $h(x)$ are precomputed for $i = 0, 1, 2, \ldots, 2n-2$, and are used to find products. Each multiplication or squaring then requires of the order of $n^2$ operations.

We also have

**Theorem 2**: If (with $h(x)$ a fixed primitive polynomial) we choose $f_1(x)$ and $g_1(x)$ at random, then for $k \geq 3$ the probability that $\alpha_k$ has a short period is $4/2^n - 4/4^n$; if we set $g_1(x) = 0$ and choose $f_1(x)$ at random, then the probability is $2/2^n$.

**3) Upper bounds on the minimal polynomial**: We define the shift operator $x$ acting on a sequence by $x\alpha = \{a_1, a_2, a_3, \ldots\}$ where $\alpha$ is the sequence $\{a_0, a_1, a_2, \ldots\}$. More generally with $f(x) = \sum c_i x^i$ we define $f(x)\alpha$ as the sequence with its $t$-th element equal to $\sum c_i a_{t+i}$. We call the binary polynomial $f(x)$ a *characteristic polynomial* of the binary sequence $\alpha$ if $f(x)\alpha \bmod 2 = 0$. Evidently with $g(x)$ any polynomial we have that $f(x)g(x) \bmod 2$ is also a characteristic polynomial of $\alpha$, so that a characteristic polynomial of a given sequence is by no means unique. However there is only one characteristic polynomial of least degree which we shall call the minimal polynomial, and its degree is called the *linear equivalence*. Thus the degree of any characteristic polynomial provides an upper bound on the linear equivalence of a sequence. Moreover the minimal polynomial is a factor of every characteristic polynomial and so in terms of the partial ordering defined by divisibility characteristic polynomials provide *upper bounds* on the minimal polynomial. We present the following formulae for characteristic polynomials of the binary sequences described above. Computer studies have encouraged us to believe that these are of degree not much greater than the degree of the minimal polynomial and therefore that they as it were provide a close upper bound.

Let $\theta$ denote a specifically chosen root of the base polynomial $h(x)$, a primitive element in GF($2^n$), since $h(x)$ is a primitive binary polynomial of degree $n$. For any non-negative integer $i$ let $w(i)$ denote the number of 1's in the base-2 expression of $i$. Then for $s > 0$ define the (binary) polynomials

$$g_s(x) = \prod_{0<i\leq T:\ w(i)\leq s} (x-\theta^i),$$

where $T = 2^n - 1$. (Note that $g_1(x) = h(x)$, and that $g_s(x) = (1 - x^T)$ for $s \geq n$.) Then it can be shown that

$$T_k(x) = g_1(x) \cdot \prod_{0\leq j<2^{k-1}} g_{k+1+2j-w(j)}(x)$$

is a characteristic polynomial of $\alpha_k$ in all cases [3]. For $k \geq n-1$ we have the simple formula

$$T_k(x) = h(x) \cdot (1 - x^T)^{2^{k-1}}.$$

When $g_0(x) = 0$ but $g_1(x) \neq 0$, so that $g(x)$ is a multiple of 2 but not of 4, then

$$T_k'(x) = g_1(x) \cdot g_5(x) \cdot g_6(x) \cdot \prod_{2 \leq j < 2^{k-1}} g_{\Delta(k,j)}(x)$$

with

$$\Delta(k,j) = k + 1 + 2j - w(j) - \max(0, k - 3 - \lfloor \log_2 j \rfloor)$$

is a characteristic polynomial of $\alpha_k$; it is a factor of $T_k(x)$ for $k \geq 4$. Finally when $g(x)$ is a multiple of 4, then

$$T_k''(x) = g_1(x) \cdot g_4(x) \cdot \prod_{1 \leq j < 2^{k-1}} g_{\Delta(k,j)}(x)$$

is a characteristic polynomial of $\alpha_k$; it is a factor of $T_k'(x)$ for $k \geq 2$ and of $T_k(x)$ for $k \geq 3$. This result is just the generalization of the result for (1) given in [3] and [5]. Computer studies demonstrate that for values of $n$ and $k$ up to 12 these characteristic polynomials are minimal in many cases; thus they should provide reasonably tight bounds on the linear equivalence, even in the general case. In any practical case with $n$ and $k$ fairly large the ratio of the upper bound provided by $T_k$ with that provided by $T_k'$ or $T_k''$ is very close to unity.

**REFERENCES**

[1] A M Frieze, J Hastad, R Kannan, J C Lagarias, A Shamir, "Reconstructing truncated integer variables satisfying linear congruences", *SIAM J. Comput.*, **17**, 262-280 (1988)

[2] M Ward, "The arithmetical theory of linear recurring series", Transactions of the American Mathematical Society, **35**, 600-628 (July 1933)

[3] Z D Dai, "Binary Sequences Derived from Maximal Length Linear Sequences over Integral Residue Rings", *Proceedings of the Workshop on Stream Ciphers*, eds. T Beth, D Gollmann, F Piper, P Wild, Report 89/1, Europäisches Institut für Systemsicherheit, Universität Karlsruhe, D-7500 Karlsruhe 1.

[4] Z D Dai, M Q Huang, "A Criterion for Primitiveness of Polynomials over $Z/(2^d)$", Kexue Tongbao, to be published

[5] Z D Dai, "Binary Sequences Derived from Sequences over the Integral Residue Rings: (I) Periods and Minimal Polynomials", to be submitted

[6] Z D Dai, T Beth, D Gollmann, "Lower Bounds for the Linear Complexity of Binary Sequences derived from Sequences over Residue Rings", Proceedings of Eurocrypt-90

[7] W G Chambers, Z D Dai, "A simple but effective modification to a multiplicative congruential random-number generator", to be published in IEE Proc E