GENERALIZED MERSENNE NUMBERS

Jerome A. Solinas

National Security Agency, Ft. Meade, MD, USA

jasolin@orion.ncsc.mil

CACR Visitor Jan.-June 1999

Abstract. There is a well known shortcut for modular multiplication modulo a Mersenne number, performing modular reduction without integer division. We generalize this technique to a larger class of primes, and discuss parameter choices which are particularly well suited for machine implementation.

Keywords: modular arithmetic, elliptic curves.

Introduction

It has long been known that certain integers are particularly well suited for modular reduction. The best known examples (*e.g.*, [1]) are the Mersenne numbers $m = 2^k - 1$. In this case, the integers (mod m) are represented as k-bit integers. When performing modular multiplication, one carries out an integer multiplication followed by a modular reduction. One thus has the problem of reducing modulo m a $2k$-bit number. Modular reduction is usually done by integer division, but this is unnecessary in the Mersenne case. Let $n < m^2$ be the integer to be reduced (mod m). Let T be the integer represented by the k most significant bits of n, and U the k least significant bits; thus

$$n = 2^k \cdot T + U,$$

with T and U each being k-bit integers. Then

$$n \equiv T + U \pmod{m}.$$

Thus the integer division by m can be replaced by an addition (mod m), which is much faster.

The main limitation of this scheme is the special multiplicative structure of Mersenne numbers. The above technique is useful only when one intends to perform modular arithmetic with a fixed long-term modulus. For most applications of this kind, the modulus needs to have a specific multiplicative structure, most commonly a prime. The above scheme proves most useful when k is a multiple of the word size of the machine. Since this word size is typically a power of 2, one must choose k which are highly composite. Unfortunately, the Mersenne numbers arising from such k are

never primes (see Proposition 13). It is therefore of interest to generalize the above technique to families of numbers containing primes.

One such family is due to Richard Crandall [2], namely, the integers $2^k - c$ for c positive and small enough to fit into one word. In this paper, we generalize in a different direction. Although there is some overlap, many of the generalized Mersenne numbers presented here are not Crandall numbers.

§1. An Example

As an example of the generalization we will pursue, let p be a prime of the form

$$p = 2^{3k} - 2^k + 1. \tag{1}$$

The integers (mod p) are represented as $3k$-bit integers. Represent a positive integer $n < p^2$ as the $6k$-bit expression

$$n = \sum_{j=0}^{5} A_j \cdot 2^{jk},$$

where $0 \le A_j < 2^k$ for each j. Our goal is to find B_j's, each a linear combination of the A_j's, such that

$$n \equiv \sum_{j=0}^{2} B_j \cdot 2^{jk} \pmod{p}.$$

This is done as follows.

We regard the expression (1) as

$$p = f(2^k),$$

where

$$f(t) = t^3 - t + 1.$$

We now base a linear feedback shift register over $\mathbb{Z}$ on $f(t)$ and step 3 times the fill $\boxed{0}\boxed{0}\boxed{1}$.

$$\begin{array}{ccccc}
(1) & & (t) & & (t^2) \\
-1 & & 1 & & 0 \\
\boxed{0} & \to & \boxed{0} & \to & \boxed{1} \\
\boxed{-1} & \to & \boxed{1} & \to & \boxed{0} \\
\boxed{0} & \to & \boxed{-1} & \to & \boxed{1} \\
\boxed{-1} & \to & \boxed{1} & \to & \boxed{-1}
\end{array}$$

The entries of the register provide the coefficients for the congruences

$$\begin{aligned}
t^3 &\equiv -1 + t \\
t^4 &\equiv -t + t^2 \\
t^5 &\equiv -1 + t - t^2
\end{aligned}$$

modulo $f(t)$. The matrix form of these congruences is

$$\begin{pmatrix} t^3 \\ t^4 \\ t^5 \end{pmatrix} \equiv \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \end{pmatrix} \pmod{f(t)}.$$

It follows that

$$\begin{aligned}
\sum_{j=0}^{5} A_j\, t^j &= \begin{pmatrix} A_0 & A_1 & A_2 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ t^2 \end{pmatrix} + \begin{pmatrix} A_3 & A_4 & A_5 \end{pmatrix} \begin{pmatrix} t^3 \\ t^4 \\ t^5 \end{pmatrix} \\
&\equiv \left(\begin{pmatrix} A_0 & A_1 & A_2 \end{pmatrix} + \begin{pmatrix} A_3 & A_4 & A_5 \end{pmatrix} \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 1 & -1 \end{pmatrix} \right) \begin{pmatrix} 1 \\ t \\ t^2 \end{pmatrix}
\end{aligned}$$

modulo $f(t)$. Therefore

$$\sum_{j=0}^{5} A_j\, t^j \equiv \sum_{j=0}^{2} B_j\, t^j \pmod{f(t)}, \tag{2}$$

where the B_j's are given by

$$(\, B_0 \quad B_1 \quad B_2\,) = (\, A_0 \quad A_1 \quad A_2\,) + (\, A_3 \quad A_4 \quad A_5\,) \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 1 & -1 \end{pmatrix}.$$

Evaluating (2) at $t = 2^k$, we obtain

$$\sum_{j=0}^{5} A_j \cdot 2^{jk} \equiv \sum_{j=0}^{2} B_j \cdot 2^{jk} \pmod{p}.$$

We have thus found our rule for fast modular reduction. Explicitly, the rule is this:

$$\sum_{j=0}^{5} A_j \cdot 2^{jk} \equiv \sum_{j=0}^{2} B_j \cdot 2^{jk} \pmod{2^{3k} - 2^k + 1}, \tag{3}$$

where

$$\begin{aligned} B_0 &= A_0 - A_3 - A_5, \\ B_1 &= A_1 + A_3 - A_4 + A_5, \\ B_2 &= A_2 + A_4 - A_5. \end{aligned} \tag{4}$$

It remains to determine how many modular additions and subtractions are necessary to implement this rule. We rearrange the terms of (4) to group together the additions and the subtractions:

$$\begin{aligned} B_0 &= A_0 && -(A_3 + A_5), \\ B_1 &= A_1 + A_3 + A_5 && -A_4, \\ B_2 &= A_2 + A_4 && -A_5. \end{aligned}$$

Thus

$$\sum_{j=0}^{2} B_j \cdot 2^{jk} = T + S_1 + S_2 - D_1 - D_2, \tag{5}$$

where

$$\begin{aligned} T &= A_0 \quad + A_1 \cdot 2^k + A_2 \cdot 2^{2k}, \\ S_1 &= \qquad A_3 \cdot 2^k + A_4 \cdot 2^{2k}, \\ S_2 &= \qquad A_5 \cdot 2^k, \\ D_1 &= A_3 \quad + A_4 \cdot 2^k + A_5 \cdot 2^{2k}, \\ D_2 &= A_5. \end{aligned}$$

If $A_0, \ldots, A_5$ are k-bit integers, then (3) and (5) give a method of reducing a $6k$-bit integer (mod $2^{3k} - 2^k + 1$) using 2 modular additions and 2 modular subtractions, or a total of 4 modular operations.

It should be further noted that modular additions and subtractions are simpler modulo a generalized Mersenne number than for a general modulus. This is due to the simple binary expansion of the prime as well as the prime being close to a power of 2.

We can also obtain the number of modular operations directly from the square matrix, avoiding the explicit calculations. The number of modular additions is found by summing the positive terms in each column and taking the maximum. The number of modular subtractions is found by summing the absolute values of the negative terms in each column and taking the maximum.

Thus the middle column has positive entries summing to 2, so that 2 modular additions are required. (Note that both S_i's contain a 2^k term.) Similarly, the left column has negative entries summing to -2, so that 2 modular subtractions are required. (Note that both D_i's contain a constant term.)

§2. The Modular Reduction Weight

We now address the general case. As in the above example, we need only step a shift register and examine the resulting entries.

Let $f(t)$ be a monic integral polynomial of degree d. Let

$$X = \begin{pmatrix} X_{0,0} & \cdots & X_{0,d-1} \\ \vdots & \ddots & \vdots \\ X_{d-1,0} & \cdots & X_{d-1,d-1} \end{pmatrix}$$

be the d-by-d matrix whose i^{th} row is the result of stepping $i+1$ times the shift register based on $f(t)$ with initial fill $\boxed{0}\cdots\boxed{0}\boxed{1}$. In other words, the entries $X_{i,j}$ are generated as follows. Let

$$f(t) = t^d - c_1\, t^{d-1} - \cdots - c_d, \tag{6}$$

a notation we will retain throughout this paper. Then

$$X_{0,j} = c_{d-j} \qquad \text{for } 0 \le j < d \tag{7}$$

and, for $1 \le i < d$,

$$X_{i,j} = \begin{cases} X_{i-1,j-1} + X_{i-1,d-1}\, c_{d-j} & \text{for } j > 0, \\ X_{i-1,d-1}\, c_d & \text{for } j = 0. \end{cases} \tag{8}$$

Let $A_0, \ldots, A_{2d-1}$ be integers, and define $B_0, \ldots, B_{d-1}$ by the matrix equation

$$(\, B_0 \quad \cdots \quad B_{d-1}\,) = (\, A_0 \quad \cdots \quad A_{d-1}\,) + (\, A_d \quad \cdots \quad A_{2d-1}\,)\, X.$$

Then

$$\sum_{j=0}^{2d-1} A_j\, t^j \equiv \sum_{j=0}^{d-1} B_j\, t^j \pmod{f(t)}.$$

If $p = f(2^k)$, then

$$\sum_{j=0}^{2d-1} A_j \cdot 2^{jk} \equiv \sum_{j=0}^{d-1} B_j \cdot 2^{jk} \pmod{p}. \tag{9}$$

For each j, let

$$Y_j = \sum_{\substack{i \\ X_{i,j}>0}} X_{i,j}$$

and

$$Z_j = \sum_{\substack{i \\ X_{i,j}<0}} -X_{i,j}.$$

Computing the right-hand side of (9) requires

$$Y_{\max} = \max_j Y_j$$

modular additions and

$$Z_{\max} = \max_j Z_j$$

modular subtractions. Thus the total number of modular operations is

$$\mathrm{wt}(f) = Y_{\max} + Z_{\max}.$$

We call $\mathrm{wt}(f)$ the *modular reduction weight* of f.

§3. Machine Implementation

Given a monic degree d integral polynomial $f(t)$ and an integer k, we can reduce $2kd$-bit integers modulo the kd-bit integer $p = f(2^k)$ using $\mathrm{wt}(f)$ modular operations by the above method. This method is especially efficient when implemented on machines whose word size divides k. This is

because the numbers S_i and D_i (as in the above example) can be formed by combining words in the appropriate way. If the word size does not divide k, it is necessary to shift bits within words.

For instance, let $k = 8$ in the above example. Then p is the prime 16776961. In binary this is

$$11111111 \;\; 11111111 \;\; 00000001.$$

Now let $n = 5863761194200$. This is less than p^2. In binary, it is

$$00000101 \;\; 01010101 \;\; 01000011 \;\; 01100111 \;\; 00101100 \;\; 11011000.$$

To reduce n modulo p, we add and subtract (mod p) the binary numbers

$$\begin{array}{cccc} & 01100111 & 00101100 & 11011000 \\ + & 01010101 & 01000011 & \\ + & & 00000101 & \\ - & 00000101 & 01010101 & 01000011 \\ - & & & 00000101 \end{array}$$

The result is 12001168, which is n (mod p).

Note that we have formed the numbers to be added and subtracted without disturbing the bits within the 8-bit bytes. Thus, this calculation is especially efficient on a machine whose word size divides 8 bits.

§4. Reduced Polynomials

Our next goal is to find families of polynomials $f(t)$ with small modular reduction weight. We will do this by searching all integral polynomials of sufficiently small degree and coefficients. We first establish three results which will shorten our search.

DEFINITION. We say that the integral polynomial $f(t)$ is *reduced* if, for each ℓ with $1 < \ell \leq d$ we have $c_j \neq 0$ for some $j \not\equiv 0 \pmod{\ell}$.

We now show that we can restrict our attention to reduced polynomials.

PROPOSITION 10. *Suppose that $p = f(2^k)$, where $f(t) \in \mathbb{Z}[t]$. Then there exists a reduced polynomial $g(t)$, having the same modular reduction weight as $f(t)$, and an integer n such that $p = g(2^n)$. If k is a multiple of a given word size, then so is n.*

Proof: We need only consider the case in which $f(t)$ is not reduced. If $c_j = 0$ for all $j \not\equiv 0 \pmod{\ell}$, then $f(t) = g(t^\ell)$ for some polynomial g. Thus, if $p = f(2^k)$, then $p = g(2^{\ell k})$, and if k is a multiple of our word size, then so is ℓk.

It remains to show that $f(t)$ and $g(t)$ have the same modular reduction weight. If X is the d-by-d matrix for $f(t)$ and X' the (d/ℓ)-by-(d/ℓ) matrix for $g(t)$, then

$$X_{i,j} = \begin{cases} X'_{i',j'} & \text{if } i \equiv j \pmod{\ell}, \\ 0 & \text{if } i \not\equiv j \pmod{\ell}. \end{cases}$$

where

$$i' = \left[\frac{i}{\ell}\right] \qquad \text{and} \qquad j' = \left[\frac{j}{\ell}\right].$$

It follows that

$$Y_j = Y'_{j'} \qquad \text{and} \qquad Z_j = Z'_{j'}$$

for all j, so that

$$Y_{\max} = Y'_{\max} \qquad \text{and} \qquad Z_{\max} = Z'_{\max}.$$

Thus

$$\mathrm{wt}(f) = \mathrm{wt}(g).$$

If g is not reduced, we can repeat this construction. $\square$

The above proposition tells us that we lose no generality in assuming that $f(t)$ is reduced. We will therefore make that assumption throughout this paper. Needless to say, we will search only reduced polynomials.

§5. Polynomials with Constant Terms

Proposition 11. *If* $f(t) = t\,g(t)$, *then* $\mathrm{wt}(f) \geq \mathrm{wt}(g)$.

Proof: If X is the d-by-d matrix for $f(t)$ and X' the $(d-1)$-by-$(d-1)$ matrix for $g(t)$, then for $i < d-1$,

$$X_{i,j} = \begin{cases} X'_{i,j-1} & \text{for } j > 0, \\ 0 & \text{for } j = 0. \end{cases}$$

Thus, for $j > 0$ we have

$$Y_j = Y'_j + X_{d,j} \qquad \text{and} \qquad Z_j = Z'_j$$

if $X_{d-1,j} \geq 0$, and

$$Y_j = Y'_j \qquad \text{and} \qquad Z_j = Z'_j + |X_{d,j}|$$

if $X_{d-1,j} < 0$. Since $X_{d-1,0} = 0$, then $Y_0 = Z_0 = 0$. Therefore

$$Y_{\max} \geq Y'_{\max} \qquad \text{and} \qquad Z_{\max} \geq Z'_{\max},$$

from which the result follows. □

As a result, we can restrict at first our search for low-weight polynomials to those with $c_d \neq 0$. Suppose that we want a list of polynomials of weight at most B. We first search the polynomials with $c_d \neq 0$ and list those with weight at most B. For each $f(t)$ in our list, we then check $t\,f(t)$, $t^2\,f(t)$, $t^3\,f(t), \ldots$ until we find an integer n for which $\mathrm{wt}(t^n\,f(t)) \leq B$ and $\mathrm{wt}(t^{n+1}\,f(t)) > B$. It follows from (11) that $\mathrm{wt}(t^k\,f(t)) \leq B$ for $k \leq n$ and $\mathrm{wt}(t^k\,f(t)) > B$ for $k > n$. Thus we append $t\,f(t), t^2\,f(t), \ldots, t^n\,f(t)$ to our list. This shortcut saves our looking at most polynomials divisible by t.

§6. Positive Polynomials

An important special case occurs if $c_j \geq 0$ for all j. (We say in this case that $f(t)$ is *positive*.) In this case, it follows from (7) and (8) that $X_{i,j} \geq 0$ for all i and j. Thus $Z_{\max} = 0$, so that our modular reductions involve only modular additions; no modular subtractions are required. This makes the positive case preferable, all other things being equal.

For a final shortcut, we define the *modular reduction complement* of the polynomial $f(t)$ defined in (6):

$$f^*(t) = t^d - \sum_{j=1}^{d} (-1)^j \, c_j \, t^{d-j}.$$

Since we are assuming $f(t)$ reduced, then $c_j \neq 0$ for some odd j; thus $f^*(t)$ is reduced and not equal to $f(t)$. By (7) and (8), the matrices of the two polynomials are related by

$$X^*_{i,j} = (-1)^{i+j+1} \, X_{i,j}.$$

Proposition 12. *If $f(t)$ is positive, then* $\mathrm{wt}(f) \leq \mathrm{wt}(f^*)$.

Proof: Since $f(t)$ is positive, then $X_{i,j} \geq 0$ for all i and j. Since $Z_j = 0$,

$$\begin{aligned} Y^*_j + Z^*_j &= \sum_i |X^*_{i,j}| \\ &= \sum_i |X_{i,j}| \\ &= Y_j. \end{aligned}$$

Thus

$$\max_j Y_j \leq \max_j Y^*_j + \max_j Z^*_j,$$

from which the result follows. $\square$

As before, this shortens the list of polynomials we need to check for low weight. We can skip at first all the complements of the positive polynomials for which the highest-degree nonzero term has j even, and check later the complements of the those positive polynomials which have low weight.

§7. Further Conditions

The search for appropriate prime moduli can be further shortened if we restrict our attention to *proper* polynomials: integral polynomials whose second-largest-degree term has negative coefficient.

The proper polynomials are the appropriate ones for our modular reduction methods. For if $f(t)$ is proper, then for large k we will have $p < 2^{kd}$; thus the integers (mod p) are always kd-bit numbers (rather than $kd + 1$ bits). This is particularly important if k is a multiple of the word size.

Since all positive polynomials are proper, the searching shortcut mentioned at the end of the previous section is unnecessary when searching only for proper polynomials.

Since proper polynomials are the appropriate ones for our reduction technique, we shall restrict our attention to them from now on.

We shall also attach two further conditions:

- *The integral polynomial $f(t)$ should be irreducible.* If this is not true, then the modulus $m = f(2^k)$ will certainly be composite.
- *The constant term of $f(t)$ must be odd.* If not, then $m = f(2^k)$ is even and therefore composite.

We shall describe an integral polynomial as being *suitable* for the Mersenne reduction method if it is reduced, proper, irreducible, of low weight, and has odd constant term.

§8. Families of Low-Weight Polynomials

We now prepare to list specific suitable polynomials. We first list some families of polynomials whose weights are easy to compute.

- The polynomial $t - b$ has weight $|b|$.
- If c and d are relatively prime, then the polynomial $t^d - t^c - 1$ is reduced. This polynomial is positive and has weight

$$w = 1 + \left\lceil \frac{d}{d-c} \right\rceil .$$

  Given w, the polynomials $t^d - t^c - 1$ with weight w are those with

$$(w-1)(w-3)\,d < (w-1)(w-2)\,c \le (w-2)^2\,d.$$

- If $d > 1$, then $t^d - t^{d-1} + 1$ is reduced and has weight $2d - 1$.
- If c and d are relatively prime and $c < d - 1$, then the polynomial $t^d - t^c + 1$ is reduced. Its weight is

$$w = 2 \left\lceil \frac{d}{d-c} \right\rceil .$$

  Given b, the polynomials $t^d - t^c + 1$ with weight $2b$ are those with

$$b(b-2)\,d < b(b-1)\,c < (b-1)^2\,d.$$

§9. List of Low-Weight Polynomials

The following list of suitable polynomials was obtained by direct search. The list includes members of the above families.

Weight 1*:*

- The polynomial $t - 1$.

Weight 3*:*

- The polynomial $t - 3$.
- The polynomials $t^d - t^c - 1$, where $0 < 2\,c \leq d$ and c and d relatively prime.
- For $d > 1$, the polynomials

$$t^d - t^{d-1} + t^{d-2} - \cdots \pm 1 = \frac{t^{d+1} + (-1)^d}{t+1}.$$

Weight 4*:*

- The polynomials $t^d - t^c - 1$, where $3\,d < 6\,c \leq 4\,d$ and c and d relatively prime.
- The polynomials $t^d - t^c + 1$, where $0 < 2\,c < d$ and c and d relatively prime.
- The polynomial $t^4 - t^3 + t^2 + 1$.

§10. Small-Weight Prime Moduli

We now list the following families of kd-bit prime moduli which are suitable for Mersenne modular reduction. All of these arise from the polynomials listed in the previous section.

Weight 1*:*

- $p = 2^k - 1$ (Mersenne primes) for k prime (see Proposition 13).

Weight 3*:*

- $p = 2^{dk} - 3$ (Crandall numbers)

- $p = 2^{dk} - 2^{ck} - 1$ where $0 < 2\,c \le d$ and $GCD(c,d) = 1$,
- $p = 2^{dk} - 2^{(d-1)k} + 2^{(d-2)k} - \cdots - 2^k + 1$

  for d even and $k \not\equiv 2 \pmod 4$ (see Proposition 14 and 15).

Weight 4*:*

- $p = 2^{dk} - 2^{ck} - 1$ where $3\,d < 6\,c < 4\,d$ and $GCD(c,d) = 1$,
- $p = 2^{dk} - 2^{ck} + 1$ where $0 < 2\,c < d$ and $GCD(c,d) = 1$,
- $p = 2^{4k} - 2^{3k} + 2^{2k} + 1$.

§11. Composite Moduli

The results of this paper can sometimes be applied usefully to composite moduli. More specifically, the modulus can also be *almost prime, i.e.,* the product of a large prime and a small integer. For example, the generalized Mersenne number $m = 2^{384} - 2^{320} - 1$ factors into $m = 19\,p$, where p is a prime. For applications in which many computations are to be done modulo p, one can work modulo m instead, using the techniques of this paper. The final result will be in terms of integers modulo $19\,p$; these can then be reduced to integers (mod p) by repeated subtraction.

Note that ordinary Mersenne numbers $m = 2^k - 1$ cannot be used in this way if k is a multiple of a typical word size. For, since k is even, the factorization

$$2^k - 1 = (2^{k/2} - 1)(2^{k/2} + 1)$$

into two nearly equal factors shows that m is not almost prime.

References

1. Knuth, Donald E., *Seminumerical Algorithms*, Addison-Wesley, 1981.
2. Crandall, Richard E., *Method and apparatus for public key exchange in a cryptographic system* (Oct. 27, 1992), U.S. Patent # 5,159,632.

Appendix 1: Four Examples

The following four generalized Mersenne primes appear in the document "Recommended Elliptic Curves for Federal Government Use," issued July 1999 by NIST and available on their website

`http://csrc.nist.gov/encryption` .

In each case, A represents an integer less than p^2, and B the result of reducing A modulo p.

Example 1.

Let $p = 2^{192} - 2^{64} - 1$. Every integer A less than p^2 can be written

$$A = A_5 \cdot 2^{320} + A_4 \cdot 2^{256} + A_3 \cdot 2^{192} + A_2 \cdot 2^{128} + A_1 \cdot 2^{64} + A_0,$$

where each A_i is a 64-bit integer. As a concatenation of 64-bit words, this can be denoted by

$$A = (A_5 \ \| \ A_4 \ \| \ \cdots \ \| \ A_0)\,.$$

The expression for B is

$$B := T + S_1 + S_2 + S_3 \bmod p,$$

where the 192-bit terms are given by

$$\begin{aligned} T &= (\ A_2 \ \|\ A_1 \ \|\ A_0\ ) \\ S_1 &= (\ 0 \ \|\ A_3 \ \|\ A_3\ ) \\ S_2 &= (\ A_4 \ \|\ A_4 \ \|\ 0\ ) \\ S_3 &= (\ A_5 \ \|\ A_5 \ \|\ A_5\ ) \end{aligned}$$

Example 2.

Let $p = 2^{224} - 2^{96} + 1$. Every integer A less than p^2 can be written

$$\begin{aligned} A = & A_{13} \cdot 2^{416} + A_{12} \cdot 2^{384} + A_{11} \cdot 2^{352} + A_{10} \cdot 2^{320} + \\ & A_9 \cdot 2^{288} + A_8 \cdot 2^{256} + A_7 \cdot 2^{224} + A_6 \cdot 2^{192} + A_5 \cdot 2^{160} + \\ & A_4 \cdot 2^{128} + A_3 \cdot 2^{96} + A_2 \cdot 2^{64} + A_1 \cdot 2^{32} + A_0, \end{aligned}$$

where each A_i is a 32-bit integer. As a concatenation of 32-bit words, this can be denoted by

$$A = (A_{13} \ \| \ A_{12} \ \| \ \cdots \ \| \ A_0)\,.$$

The expression for B is

$$B := T + S_1 + S_2 - D_1 - D_2 \bmod p,$$

where the 224-bit terms are given by

$$\begin{aligned} T &= (\ A_6 \ \|\ A_5 \ \|\ A_4 \ \|\ A_3 \ \|\ A_2 \ \|\ A_1 \ \|\ A_0\ ) \\ S_1 &= (\ A_{10} \ \|\ A_9 \ \|\ A_8 \ \|\ A_7 \ \|\ 0 \ \|\ 0 \ \|\ 0\ ) \\ S_2 &= (\ 0 \ \|\ A_{13} \ \|\ A_{12} \ \|\ A_{11} \ \|\ 0 \ \|\ 0 \ \|\ 0\ ) \\ D_1 &= (\ A_{13} \ \|\ A_{12} \ \|\ A_{11} \ \|\ A_{10} \ \|\ A_9 \ \|\ A_8 \ \|\ A_7\ ) \\ D_2 &= (\ 0 \ \|\ 0 \ \|\ 0 \ \|\ 0 \ \|\ A_{13} \ \|\ A_{12} \ \|\ A_{11}\ ) \end{aligned}$$

Example 3.

Let $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$. Every integer A less than p^2 can be written

$$\begin{aligned} A = & A_{15} \cdot 2^{480} + A_{14} \cdot 2^{448} + A_{13} \cdot 2^{416} + A_{12} \cdot 2^{384} + A_{11} \cdot 2^{352} + \\ & A_{10} \cdot 2^{320} + A_9 \cdot 2^{288} + A_8 \cdot 2^{256} + A_7 \cdot 2^{224} + A_6 \cdot 2^{192} + \\ & A_5 \cdot 2^{160} + A_4 \cdot 2^{128} + A_3 \cdot 2^{96} + A_2 \cdot 2^{64} + A_1 \cdot 2^{32} + A_0, \end{aligned}$$

where each A_i is a 32-bit integer. As a concatenation of 32-bit words, this can be denoted by

$$A = (A_{15} \ \| \ A_{14} \ \| \ \cdots \ \| \ A_0)\,.$$

The expression for B is

$$B := T + 2\,S_1 + 2\,S_2 + S_3 + S_4 - D_1 - D_2 - D_3 - D_4 \bmod p,$$

where the 256-bit terms are given by

$$\begin{aligned} T &= (\ A_7 \ \|\ A_6 \ \|\ A_5 \ \|\ A_4 \ \|\ A_3 \ \|\ A_2 \ \|\ A_1 \ \|\ A_0\ ) \\ S_1 &= (\ A_{15} \ \|\ A_{14} \ \|\ A_{13} \ \|\ A_{12} \ \|\ A_{11} \ \|\ 0 \ \|\ 0 \ \|\ 0\ ) \\ S_2 &= (\ 0 \ \|\ A_{15} \ \|\ A_{14} \ \|\ A_{13} \ \|\ A_{12} \ \|\ 0 \ \|\ 0 \ \|\ 0\ ) \\ S_3 &= (\ A_{15} \ \|\ A_{14} \ \|\ 0 \ \|\ 0 \ \|\ 0 \ \|\ A_{10} \ \|\ A_9 \ \|\ A_8\ ) \\ S_4 &= (\ A_8 \ \|\ A_{13} \ \|\ A_{15} \ \|\ A_{14} \ \|\ A_{13} \ \|\ A_{11} \ \|\ A_{10} \ \|\ A_9\ ) \\ D_1 &= (\ A_{10} \ \|\ A_8 \ \|\ 0 \ \|\ 0 \ \|\ 0 \ \|\ A_{13} \ \|\ A_{12} \ \|\ A_{11}\ ) \\ D_2 &= (\ A_{11} \ \|\ A_9 \ \|\ 0 \ \|\ 0 \ \|\ A_{15} \ \|\ A_{14} \ \|\ A_{13} \ \|\ A_{12}\ ) \\ D_3 &= (\ A_{12} \ \|\ 0 \ \|\ A_{10} \ \|\ A_9 \ \|\ A_8 \ \|\ A_{15} \ \|\ A_{14} \ \|\ A_{13}\ ) \\ D_4 &= (\ A_{13} \ \|\ 0 \ \|\ A_{11} \ \|\ A_{10} \ \|\ A_9 \ \|\ 0 \ \|\ A_{15} \ \|\ A_{14}\ ) \end{aligned}$$

Example 4.

Let $p = 2^{384} - 2^{128} - 2^{96} + 2^{32} - 1$. Every integer A less than p^2 can be written

$$\begin{aligned} A = & A_{23} \cdot 2^{736} + A_{22} \cdot 2^{704} + A_{21} \cdot 2^{672} + A_{20} \cdot 2^{640} + A_{19} \cdot 2^{608} + \\ & A_{18} \cdot 2^{576} + A_{17} \cdot 2^{544} + A_{16} \cdot 2^{512} + A_{15} \cdot 2^{480} + A_{14} \cdot 2^{448} + \\ & A_{13} \cdot 2^{416} + A_{12} \cdot 2^{384} + A_{11} \cdot 2^{352} + A_{10} \cdot 2^{320} + A_9 \cdot 2^{288} + \\ & A_8 \cdot 2^{256} + A_7 \cdot 2^{224} + A_6 \cdot 2^{192} + A_5 \cdot 2^{160} + A_4 \cdot 2^{128} + \\ & A_3 \cdot 2^{96} + A_2 \cdot 2^{64} + A_1 \cdot 2^{32} + A_0, \end{aligned}$$

where each A_i is a 32-bit integer. As a concatenation of 32-bit words, this can be denoted by

$$A = (A_{23} \ \| \ A_{22} \ \| \ \cdots \ \| \ A_0) .$$

The expression for B is

$$B := T + 2\,S_1 + S_2 + S_3 + S_4 + S_5 + S_6 - D_1 - D_2 - D_3 \bmod p,$$

where the 384-bit terms are given by

$$\begin{aligned} T &= (A_{11} \| A_{10} \| A_9 \| A_8 \| A_7 \| A_6 \| A_5 \| A_4 \| A_3 \| A_2 \| A_1 \| A_0) \\ S_1 &= (0 \| 0 \| 0 \| 0 \| 0 \| A_{23} \| A_{22} \| A_{21} \| 0 \| 0 \| 0 \| 0) \\ S_2 &= (A_{23} \| A_{22} \| A_{21} \| A_{20} \| A_{19} \| A_{18} \| A_{17} \| A_{16} \| A_{15} \| A_{14} \| A_{13} \| A_{12}) \\ S_3 &= (A_{20} \| A_{19} \| A_{18} \| A_{17} \| A_{16} \| A_{15} \| A_{14} \| A_{13} \| A_{12} \| A_{23} \| A_{22} \| A_{21}) \\ S_4 &= (A_{19} \| A_{18} \| A_{17} \| A_{16} \| A_{15} \| A_{14} \| A_{13} \| A_{12} \| A_{20} \| 0 \| A_{23} \| 0) \\ S_5 &= (0 \| 0 \| 0 \| 0 \| A_{23} \| A_{22} \| A_{21} \| A_{20} \| 0 \| 0 \| 0 \| 0) \\ S_6 &= (0 \| 0 \| 0 \| 0 \| 0 \| 0 \| A_{23} \| A_{22} \| A_{21} \| 0 \| 0 \| A_{20}) \\ D_1 &= (A_{22} \| A_{21} \| A_{20} \| A_{19} \| A_{18} \| A_{17} \| A_{16} \| A_{15} \| A_{14} \| A_{13} \| A_{12} \| A_{23}) \\ D_2 &= (0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| A_{23} \| A_{22} \| A_{21} \| A_{20} \| 0) \\ D_3 &= (0 \| 0 \| 0 \| 0 \| 0 \| 0 \| 0 \| A_{23} \| A_{23} \| 0 \| 0 \| 0) \end{aligned}$$

Appendix 2: Prime-Producing Proper Polynomials

Many of the families of low-weight polynomials can assume prime values only when the parameters satisfy certain conditions. We present some elementary conditions for this, for two of the families of polynomials from §9.

Proposition 13. *If $p = 2^k - 1$ is prime, then k is prime.*

Proof: If $a > 1$ and $b > 1$, then each factor of

$$2^{ab} - 1 = (2^a - 1)\,(2^{a(b-1)} + \cdots + 2^a + 1)$$

is greater than 1; hence $2^{ab} - 1$ is composite. Thus $p = 2^k - 1$ is composite if k is composite. □

Proposition 14. *If $d > 1$ is odd and $k > 0$, then*

$$m = 2^{dk} - 2^{(d-1)k} + 2^{(d-2)k} - \cdots + 2^k - 1$$

is composite (except for the case $d = 3$, $k = 1$).

Proof: We factor

$$m = (2^k - 1)\,(2^{(d-1)k} + 2^{(d-3)k} + \cdots + 2^{2k} + 1).$$

If $k > 1$, then both factors are greater than 1, so that m is composite. If $k = 1$, then

$$\begin{aligned} m &= 1 + 2^2 + 2^4 + \cdots + 2^{d-1} \\ &= (2^{d+1} - 1)\,/\,3 \\ &= (2^{(d+1)/2} - 1)\,(2^{(d+1)/2} + 1)\,/\,3. \end{aligned}$$

If $d \geq 5$, then $2^{(d+1)/2} + 1 > 2^{(d+1)/2} - 1 > 3$, so that m has at least two factors greater than 1. $\square$

PROPOSITION 15. *If $d > 1$ is even and $k \equiv 2 \pmod 4$ is positive, then*

$$m = 2^{dk} - 2^{(d-1)k} + 2^{(d-2)k} - \cdots - 2^k + 1$$

is composite.

Proof: Since

$$m = \frac{2^{(d+1)k} + 1}{2^k + 1},$$

it suffices to factor the numerator into two factors, each larger than $2^k + 1$. Let $w = (k-2)/4$; then $w \geq 0$, and

$$\begin{aligned} 2^{(d+1)k} + 1 &= 2^{4w+2} + 1 \\ &= \left(2^{2dw+2w+d+1} - 2^{dw+w+1+d/2} + 1\right) \\ &\quad \times \left(2^{2dw+2w+d+1} + 2^{dw+w+1+d/2} + 1\right). \end{aligned}$$

Thus we need to prove that

$$\begin{aligned} 2^{4w+2} &< 2^{2dw+2w+d} - 2^{dw+w+1+d/2} \\ &= 2^{dw+w+1+d/2} \left(2^{dw+w-1+d/2} - 1\right). \end{aligned} \tag{16}$$

If $d \geq 4$, then

$$4w + 2 < d\,w + w + 1 + d/2,$$

so that (16) holds. In the case $d = 2$, (16) becomes

$$4\,W^4 < 4\,W^3 \left(4\,W^3 - 1\right),$$

where $W = 2^w$. This holds for all $w \geq 0$ because

$$W < 4\,W^3 - 1$$

for all $W \geq 1$. $\square$