Optimal Irreducible Polynomials for GF(2^m) arithmetic

Michael Scott
School of Computing
Dublin City University

GF(2^m) polynomial representation

- A polynomial with coefficients either 0 or 1 (*m* is a small prime)
- Stored as an array of bits, of length *m*, packed into computer words
- Addition (and subtraction) – easy – XOR. No reduction required as bit length does not increase.

GF(2^m) arithmetic 1

- Squaring, easy, simply insert 0 between coefficients.
  - Example 110101 → 10100010001
- Multiplication – <u>artificially hard</u> as instruction sets do not support “binary polynomial” multiplication, or “multiplication without carries” – which is actually simpler in hardware than integer multiplication! Really annoying!

GF(2^m) arithmetic 2

- So we use Comb or Karatsuba methods…
- Squaring or multiplication results in a polynomial with 2*m*-1 coefficients.
- This must be *reduced* with respect to an irreducible polynomial, to yield a field element of *m* bits.
- For example for m=17, $x^{17}+x^5+1$

Dublin City University
Ollscoil Chathair Bhaile Átha Cliath

GF(2^m) arithmetic 3

- This trinomial has no factors (irreducible)
- Reduction can be performed using shifts and XORs
- $x^{17}+x^5+1$ = 100000000000100001
- Example – reduce
- 10100101010101101010101

GF(2^m) arithmetic 4

$$
\begin{array}{l}
10100101010101101010101 \\
100000000000100001 \quad \oplus \\
00100101010110000110101 \quad \leftarrow \\
100101010110000110101 \\
100000000000100001 \quad \oplus \\
000101010110100111101 \leftarrow \\
101010110100111101 \\
100000000000100001 \quad \oplus \\
001010110100011100 \leftarrow \\
1010110100011100 \rightarrow \text{result!}
\end{array}
$$

Reduction in software - 1

- Consider the standard pentanomial $x^{163}+x^7+x^6+x^3+1$
- Assume value to be reduced is represented as 11 32-bit words *g[.]*
- To be reduced to 6 word result
- In software the unrolled reduction algorithm looks like this

Dublin City University
Ollscoil Chathair Bhaile Átha Cliath

Reduction in software - 2

- $g10 \leftarrow g[10]$, $g9 \leftarrow g[9]$, $g8 \leftarrow g[8]$, $g7 \leftarrow g[7]$, $g6 \leftarrow g[6]$
- $g[10] \leftarrow g[9] \leftarrow g[8] \leftarrow g[7] \leftarrow g[6] \leftarrow 0$
- $g[5] \leftarrow g[5] \oplus (g10 \ll 4) \oplus (g10 \ll 3) \oplus g10 \oplus (g9 \gg 28) \oplus (g9 \gg 29)$
- $g[4] \leftarrow g[4] \oplus (g10 \ll 29) \oplus (g9 \gg 3) \oplus (g9 \ll 4) \oplus (g9 \ll 3) \oplus g9 \oplus (g8 \gg 28) \oplus (g8 \gg 29)$
- $g[3] \leftarrow g[3] \oplus (g9 \ll 29) \oplus (g8 \gg 3) \oplus (g8 \ll 4) \oplus (g8 \ll 3) \oplus g8 \oplus (g7 \gg 28) \oplus (g7 \gg 29)$
- $g[2] \leftarrow g[2] \oplus (g8 \ll 29) \oplus (g7 \gg 3) \oplus (g7 \ll 4) \oplus (g7 \ll 3) \oplus g7 \oplus (g6 \gg 28) \oplus (g6 \gg 29)$
- $g[1] \leftarrow g[1] \oplus (g7 \ll 29) \oplus (g6 \gg 3) \oplus (g6 \ll 4) \oplus (g6 \ll 3) \oplus g6$
- $g[0] \leftarrow g[0] \oplus (g6 \ll 29)$
- $t \leftarrow g[5] \gg 3$, $g[0] \leftarrow g[0] \oplus t$, $t \leftarrow (t \ll 3)$
- $g[1] \leftarrow g[1] \oplus (t \gg 28) \oplus (t \gg 29)$
- $g[0] \leftarrow g[0] \oplus t \oplus (t \ll 4) \oplus (t \ll 3)$
- $g[5] \leftarrow g[5] \oplus t$

- 38 XORs, 33 shifts

Dublin City University
Ollscoil Chathair Bhaile Átha Cliath

Reduction in software - 3

- The shift values are
- S=163 mod 32 and 32-S = (3,29)
- S=163-7 mod 32 and 32-S = (28,4)
- S=163-6 mod 32 and 32-S = (29,3)
- S=163-3 mod 32 and 32-S = (0,32) (!!)
- A shift by 32 results in a zero, and a shift by 0 is free. What if the irreducible polynomial was chosen to make this happen more often? Saves 1 XOR and 2 shifts per line…

A better polynomial

- Now try $x^{163}+x^{99}+x^{97}+x^3+1$
- Note that this time the shifts are by (3,29), (0,32), (2,30) and (0,32)
- Should result in shorter, faster reduction code.
- It does…

Dublin City University
Ollscoil Chathair Bhaile Átha Cliath

A better polynomial

- $g10 \leftarrow g[10]$, $g9 \leftarrow g[9]$, $g8 \leftarrow g[8]$, $g7 \leftarrow g[7]$, $g6 \leftarrow g[6]$
- $g[10] \leftarrow g[9] \leftarrow g[8] \leftarrow g[7] \leftarrow g[6] \leftarrow 0$
- $g8 \leftarrow g8 \oplus g10 \oplus (g10 \gg 2)$, $g7 \leftarrow g7 \oplus (g10 \ll 30) \oplus g9 \oplus (g9 \gg 2)$
- $g6 \leftarrow g6 \oplus (g9 \ll 30) \oplus g8 \oplus (g8 \gg 2)$
- $g[5] \leftarrow g[5] \oplus g10 \oplus (g8 \ll 30) \oplus g7 \oplus (g7 \gg 2)$
- $g[4] \leftarrow g[4] \oplus (g10 \ll 29) \oplus (g9 \gg 3) \oplus g9 \oplus (g7 \ll 30) \oplus g6 \oplus (g6 \gg 2)$
- $g[3] \leftarrow g[3] \oplus (g9 \ll 29) \oplus (g8 \gg 3) \oplus g8 \oplus (g6 \ll 30)$
- $g[2] \leftarrow g[2] \oplus (g8 \ll 29) \oplus (g7 \gg 3) \oplus g7$
- $g[1] \leftarrow g[1] \oplus (g7 \ll 29) \oplus (g6 \gg 3) \oplus g6$
- $g[0] \leftarrow g[0] \oplus (g6 \ll 29)$
- $t \leftarrow g[5] \gg 3$, $g[0] \leftarrow g[0] \oplus t$, $t \leftarrow (t \ll 3)$
- $g[0] \leftarrow g[0] \oplus t$
- $g[2] \leftarrow g[2] \oplus (t \ll 30)$
- $g[3] \leftarrow g[3] \oplus t \oplus (t \gg 2)$
- $g[5] \leftarrow g[5] \oplus t$

- 35 XORs and 23 shifts

A better polynomial

- If the irreducible trinomial is of the form x^m+x^a+1, and (m-a) is a multiple of the word-length, then it's a *lucky* trinomial (LT).
- If the irreducible pentanomial is of the form $x^m+x^a+x^b+x^c+1$, and (m-a), (m-b), (m-c) are all multiples of the word-length, then it's a *lucky* pentanomial (LP).

A better polynomial

- If only two out of three of (m-a), (m-b) and (m-c) are multiples of the wordlength, then it’s a *fortunate* pentanomial (FP).
- LTs may not exist or very rare.
- LPs can be quite plentiful
- FPs are even more plentiful
- Clearly helps if a (and b and c for a pentanomial) are all odd.

Dublin City University
Ollscoil Chathair Bhaile Átha Cliath

Square roots

- As it happens having, *a* (and *b* and *c*) odd is already a good idea, as it facilitates a much faster square rooting algorithm (Fong et al.)
- Square roots are important for point-halving algorithms, and for the η_T pairing

Square roots

- Assume trinomial
- Let $\zeta = x^{(m+1)/2} + x^{(a+1)/2}$
- Then $\sqrt{a} = a_{even} + \zeta . a_{odd}$
- Where a_{even} are the even indexed elements of a collapsed into a half sized bit array, and a_{odd} are the odd indexed elements of a also collapsed into a half sized bit array.

Dublin City University
Ollscoil Chathair Bhaile Átha Cliath

Example

- In GF(2^{17}) find square root of
- 01010110100011100
- a_{even} = 000110110
- a_{odd} = 011100010
- $\zeta = x^9 + x^3$
- So the square root is..

Example

0000000000000110110

0111000100000000000 ⊕

0000000111000100000 ⊕

=

0111000011001001100

No reduction required!

Square roots

- In software it helps if $(m+1)/2$ and $(a+1)/2$ are multiples of the word length – again less shifts will be needed.
- No reason not to insist on trinomials/pentanomials with odd *a* (*b* and *c*)

Some bad news..

- If a trinomial does not exist for the given field, a lucky pentanomial does not exist either (Buhler)
- If $m=\pm 1 \bmod 8$ a trinomial might be found, and so might a lucky pentanomial
- If $m= \pm 3 \bmod 8$, no trinomial, no lucky pentanomial, but maybe a fortunate pentanomial.

Pecking order..

- A lucky trinomial beats a lucky pentanomial, beats an ordinary trinomial, beats a fortunate pentanomial.
- But for $m = \pm 3 \bmod 8$ only hope is a fortunate pentanomial ☹
- So pentanomial can be better than a trinomial – Yes!

Pentanomial beats a trinomial?

- Consider $GF(2^{233})$
- Trinomial $x^{233}+x^{159}+1$
- Lucky pentanomial for 32-bit processor $x^{233}+x^{201}+x^{105}+x^{9}+1$
- Trinomial requires 4 XORs and 4 shifts per iteration of the (rolled) reduction algorithm
- Pentanomial requires 5 XORs and only 2 shifts

Dublin City University
Ollscoil Chathair Bhaile Átha Cliath

Code for GF(2^{233}) trinomial

```
for (i=xl-1;i>=8;i--)
{
   w=gx[i]; gx[i]=0;
   gx[i-2]^=(w>>10);
   gx[i-3]^=(w<<22);
   gx[i-7]^=(w>>9);
   gx[i-8]^=(w<<23);
}  /* XORs= 4 shifts= 4 */
top=gx[7]>>9; gx[0]^=top; top<<=9;
gx[4]^=(top<<22);
gx[5]^=(top>>10);
gx[7]^=top;
```

Code for GF(2^{233}) pentanomial

```
for (i=xl-1;i>=8;i--)
{
   w=gx[i]; gx[i]=0;
   gx[i-1]^=w;
   gx[i-4]^=w;
   gx[i-7]^=(w>>9)^w;
   gx[i-8]^=(w<<23);
}  /* XORs= 5 shifts= 2 */
top=gx[7]>>9; gx[0]^=top; top<<=9;
gx[0]^=top;
gx[3]^=top;
gx[6]^=top;
gx[7]^=top;
```

Pentanomial beats a trinomial?

- But wait.. On the ARM processor shifts are free!
- EOR R1,R2,R3,LSL #7
- $R1 = R2 \oplus (R3 \ll 7)$
- Very nice feature for $GF(2^m)$ arithmetic!
- But not supported on most other architectures!

Pentanomial beats a trinomial?

- In fact on smaller processors shifts may be <u>much</u> more expensive than XOR.
- May only support 1-bit shifts and rotates, so multi-bit shifts require multiple clock cycles.
- On MIPS/Pentium a shift (of any length) costs same as XOR
- So in many (most?) cases a lucky pentanomial beats a trinomial.

DCU
Dublin City University
Ollscoil Chathair Bhaile Átha Cliath

Real-world architectures

- Texas Instruments msp430
- 16-bit processor
- One bit shifts/rotates only
- Used in Wireless Sensor Networks
- Atmel Atmega-128
- 8-bit processor
- One bit shifts/rotates only

Dublin City University
Ollscoil Chathair Bhaile Átha Cliath

What about m=±3 mod 8?

- Maybe try to find a lucky redundant pentanomial (Brent & Zimmerman)?
- For $m=163$, $x^{165}+x^{69}+x^{37}+x^5+1$ has 163 degree factor.
- Are redundant polynomials a good idea??

Results

- Using a simple cost model which costs XORs and shifts appropriately for a selection of architectures, we search for the optimal polynomials, for standard values of *m*.
- For different word length, often different results (not surprising), but also often there is agreement

Results

- In many cases a lucky pentanomial is better than a trinomial
- For msp430 processor, pentanomial is always superior.
- For Atmel 8-bit processor lucky trinomials are possible.

Dublin City University
Ollscoil Chathair Bhaile Átha Cliath

Last words

- The current standard polynomials should be scrapped! They are awful!
- Replacements should be square root friendly, and close to optimal for a wide range of architectures.

Optimal Irreducible Polynomials

Questions??

Dublin City University
Ollscoil Chathair Bhaile Átha Cliath