Aalto University
School of Science

Another Look at Inversions over Binary Fields

Vassil Dimitrov[1] *Kimmo Järvinen*[2]

[1]Dept. of Electrical and Computer Engineering, University of Calgary, Canada
[2]Dept. of Information and Computer Science, Aalto University, Finland

The 21st IEEE International Symposium on Computer Arithmetic, ARITH21
Austin, TX, USA, April 7–10, 2013

Introduction

Inversions in binary fields

- Applications, especially, in public-key cryptography (e.g., elliptic curve cryptography)
- Can be computed essentially in two different ways: Extended Euclidean Algorithm or Fermat's Little Theorem

We will introduce new algorithms for computing inversions that

- are more economical than the popular Itoh-Tsujii algorithm,
- achieve the lowest possible number of multiplications for four out of five NIST fields, and
- have nice implementation properties, especially, on HW

Inversion with Fermat's Little Theorem

Multiplicative inverse

Given $A \neq 0 \in GF(2^m)$, find A^{-1} such that $A^{-1} \cdot A = 1$

- $A^{2^m-1} = 1$ for all $A \neq 0 \in GF(2^m)$

$\Rightarrow$ $A^{-1} = A^{2^m-2}$

- $A^{2(2^{m-1}-1)} = A^{2(1+2+2^2+\ldots+2^{m-2})}$

Inversion with Fermat's Little Theorem

Multiplicative inverse

Given $A \neq 0 \in GF(2^m)$, find A^{-1} such that $A^{-1} \cdot A = 1$

- $A^{2^m-1} = 1$ for all $A \neq 0 \in GF(2^m)$

$\Rightarrow$ $A^{-1} = A^{2^m-2}$

- $A^{2(2^{m-1}-1)} = A^{2(1+2+2^2+\ldots+2^{m-2})}$

Standard exponentiation

$A^{2(1+2+2^2+\ldots+2^{m-2})} = B \cdot B^2 \cdot B^{2^2} \cdot \ldots \cdot B^{2^{m-2}}$ where $B = A^2$

- $m-2$ multiplications
- $m-1$ squarings

Itoh-Tsujii

Introduced by Itoh and Tsujii in 1988

$$1+2+\ldots+2^{m-2} = \begin{cases} (1+2)(1+2^2+\ldots+2^{m-3}), & \text{if } m-1 \text{ even} \\ 1+2(1+2)(1+2^2+\ldots+2^{m-4}), & \text{if } m-1 \text{ odd} \end{cases}$$

Example

$GF(2^{31})$: $1+2+\ldots+2^{29} =$
$(1+2)(1+2^2(1+2^2)(1+2^4(1+2^4)(1+2^8(1+2^8))))$
$\Rightarrow$ 7 multiplications, 30 squarings

In general

- $\lfloor \log(m-1) \rfloor + H(m-1) - 1$ multiplications
- $m-1$ squarings

The New Algorithm

Idea

Use the same approach as IT but try to minimize the number of additions by using multiple bases

Algorithm

Double-base with bases $\{2, 3\}$:

$$1 + 2 + \ldots + 2^{m-2} =$$

$$\begin{cases} (1+2+2^2)\cdot(1+2^3+2^6+\ldots+2^{m-4}) & \text{if } m-1 = 0,3 \pmod 6 \\ (1+2)\cdot(1+2^2+2^4+\ldots+2^{m-3}) & \text{if } m-1 = 2,4 \pmod 6 \\ 1+2\cdot(1+2)\cdot(1+2^2+2^4+\ldots+2^{m-4}) & \text{if } m-1 = 1,5 \pmod 6 \end{cases}$$

For triple-base version with bases $\{2, 3, 5\}$, we extend this with:

$((1+2)(1+2^2)+2^4)(1+2^5+\ldots+2^{m-6})$ if $m-1 = 0 \pmod 5$

Example: $GF(2^{31})$

We find a decomposition for

$$1 + 2 + 2^2 + \ldots + 2^{28} + 2^{29}$$

Example: $GF(2^{31})$

We find a decomposition for

$$1 + 2 + 2^2 + \ldots + 2^{28} + 2^{29}$$

$$30 \bmod 6 = 0 \quad \Rightarrow (1 + 2 + 2^2) \cdot (1 + 2^3 + 2^{3 \cdot 2} + \ldots + 2^{3 \cdot 9})$$

Example: $GF(2^{31})$

We find a decomposition for

$$1 + 2 + 2^2 + \ldots + 2^{28} + 2^{29} =$$
$$(1 + 2 + 2^2)$$

$$30 \bmod 6 = 0 \quad \Rightarrow (1 + 2 + 2^2) \cdot (1 + 2^3 + 2^{3\cdot 2} + \ldots + 2^{3\cdot 9})$$

Example: $GF(2^{31})$

We find a decomposition for

$$1 + 2 + 2^2 + \ldots + 2^{28} + 2^{29} =$$

$$(1 + 2 + 2^2)$$

$$30 \bmod 6 = 0 \quad \Rightarrow (1 + 2 + 2^2) \cdot (1 + 2^3 + 2^{3\cdot 2} + \ldots + 2^{3\cdot 9})$$
$$10 \bmod 6 = 4 \quad \Rightarrow (1 + 2^3) \cdot (1 + 2^6 + 2^{6\cdot 2} + 2^{6\cdot 3} + 2^{6\cdot 4})$$

Example: $GF(2^{31})$

We find a decomposition for

$$1 + 2 + 2^2 + \ldots + 2^{28} + 2^{29} =$$
$$(1 + 2 + 2^2) \cdot (1 + 2^3)$$

$$30 \bmod 6 = 0 \quad \Rightarrow (1 + 2 + 2^2) \cdot (1 + 2^3 + 2^{3 \cdot 2} + \ldots + 2^{3 \cdot 9})$$
$$10 \bmod 6 = 4 \quad \Rightarrow (1 + 2^3) \cdot (1 + 2^6 + 2^{6 \cdot 2} + 2^{6 \cdot 3} + 2^{6 \cdot 4})$$

Example: $GF(2^{31})$

We find a decomposition for

$$1 + 2 + 2^2 + \ldots + 2^{28} + 2^{29} =$$
$$(1 + 2 + 2^2) \cdot (1 + 2^3)$$

$$30 \bmod 6 = 0 \quad \Rightarrow (1 + 2 + 2^2) \cdot (1 + 2^3 + 2^{3\cdot 2} + \ldots + 2^{3\cdot 9})$$
$$10 \bmod 6 = 4 \quad \Rightarrow (1 + 2^3) \cdot (1 + 2^6 + 2^{6\cdot 2} + 2^{6\cdot 3} + 2^{6\cdot 4})$$
$$5 \bmod 6 = 5 \quad \Rightarrow 1 + 2^6 \cdot (1 + 2^6) \cdot (1 + 2^{6\cdot 2})$$

Example: $GF(2^{31})$

We find a decomposition for

$$1 + 2 + 2^2 + \ldots + 2^{28} + 2^{29} =$$
$$(1 + 2 + 2^2) \cdot (1 + 2^3) \cdot (1 + 2^6 \cdot (1 + 2^6) \cdot (1 + 2^{12}))$$

$$30 \bmod 6 = 0 \quad \Rightarrow (1 + 2 + 2^2) \cdot (1 + 2^3 + 2^{3 \cdot 2} + \ldots + 2^{3 \cdot 9})$$
$$10 \bmod 6 = 4 \quad \Rightarrow (1 + 2^3) \cdot (1 + 2^6 + 2^{6 \cdot 2} + 2^{6 \cdot 3} + 2^{6 \cdot 4})$$
$$5 \bmod 6 = 5 \quad \Rightarrow 1 + 2^6 \cdot (1 + 2^6) \cdot (1 + 2^{6 \cdot 2})$$

Example: $GF(2^{31})$

We find a decomposition for

$$1 + 2 + 2^2 + \ldots + 2^{28} + 2^{29} =$$
$$(1 + 2 + 2^2) \cdot (1 + 2^3) \cdot (1 + 2^6 \cdot (1 + 2^6) \cdot (1 + 2^{12}))$$

$$\begin{aligned}
30 \bmod 6 = 0 \quad &\Rightarrow (1 + 2 + 2^2) \cdot (1 + 2^3 + 2^{3\cdot 2} + \ldots + 2^{3\cdot 9}) \\
10 \bmod 6 = 4 \quad &\Rightarrow (1 + 2^3) \cdot (1 + 2^6 + 2^{6\cdot 2} + 2^{6\cdot 3} + 2^{6\cdot 4}) \\
5 \bmod 6 = 5 \quad &\Rightarrow 1 + 2^6 \cdot (1 + 2^6) \cdot (1 + 2^{6\cdot 2})
\end{aligned}$$

- 6 multiplications and 30 squarings
- IT required 7 multiplications and 30 squarings

The New Algorithm vs. Itoh-Tsujii

Average number of multiplications:

- $1.5 \log(m-1)$ for IT
- $1.42 \log(m-1)$ for $\{2,3\}$
- $1.39 \log(m-1)$ for $\{2,3,5\}$

For fields $GF(2^m)$, $1 \leq m \leq 1023$:

- 18 (1.8 %): $\{2,3\}$ is the best
- 109 (10.7 %): $\{2,3,5\}$ is the best
- 387 (37.8 %): $\{2,3\}$ and $\{2,3,5\}$ are the best
- 79 (7.7 %): IT ($\{2\}$) is the best
- 430 (42.0 %): All are equally good

$\Rightarrow$ We are better for 50.2 % and worse for 7.7 % of the cases

The NIST Fields

Itoh-Tsujii:

$GF(2^{163})$	$GF(2^{233})$	$GF(2^{283})$	$GF(2^{409})$	$GF(2^{571})$
9	10	11	11	13

The best from both $\{2, 3\}$ and $\{2, 3, 5\}$:

$GF(2^{163})$	$GF(2^{233})$	$GF(2^{283})$	$GF(2^{409})$	$GF(2^{571})$
9	10	12	10	12

Addition Chains

- Inversion algorithms can be derived from addition chains
- Using an optimal addition chain (OAC) leads to the smallest number of multiplications
- Different chains can have different costs even if the length (number of multiplications) is the same
- Which is the best?

Example

162 : 99 OACs (length 10)
232 : 894 OACs (length 11)
282 : 5600 OACs (length 12)
408 : 40 OACs (length 11)
570 : 4387 OACs (length 13)

Practical Implications

Fewer (even by one) multiplications make a large difference and, therefore, practically all work so far has concentrated on minimizing multiplications.

Although multiplications usually dominate the costs of inversions, other aspects should not be overlooked

- Temporary variables
- Squarings

Temporary Variables

How Are Inversions Computed?

From a Decomposition to an Algorithm

$$GF(2^{31}) : A^{-1} = A^{2^{31}-2} = A^{2(2^{30}-1)} = A^{2(1+2+\ldots+2^{29})}$$

$$1 + 2 + \ldots + 2^{29} = (1 + 2 + 2^2)(1 + 2^3)(1 + 2^6\,(1 + 2^6)(1 + 2^{12}))$$

How Are Inversions Computed?

From a Decomposition to an Algorithm

$$GF(2^{31}) : A^{-1} = A^{2^{31}-2} = A^{2(2^{30}-1)} = A^{2(1+2+\ldots+2^{29})}$$

$$1 + 2 + \ldots + 2^{29} = (1 + 2 + 2^2)(1 + 2^3)(1 + 2^6(1 + 2^6)(1 + 2^{12}))$$

1. $T_1 \leftarrow A^2$

How Are Inversions Computed?

From a Decomposition to an Algorithm

$$GF(2^{31}) : A^{-1} = A^{2^{31}-2} = A^{2(2^{30}-1)} = A^{2(1+2+\ldots+2^{29})}$$

$$1 + 2 + \ldots + 2^{29} = (1 + 2 + 2^2)(1 + 2^3)(1 + 2^6\,(1 + 2^6)(1 + 2^{12}))$$

1. $T_1 \leftarrow A^2$
2. $T_2 \leftarrow T_1^2$
3. $T_1 \leftarrow T_1 \times T_2$
4. $T_2 \leftarrow T_2^2$
5. $T_1 \leftarrow T_1 \times T_2$

How Are Inversions Computed?

From a Decomposition to an Algorithm

$$GF(2^{31}) : A^{-1} = A^{2^{31}-2} = A^{2(2^{30}-1)} = A^{2(1+2+\ldots+2^{29})}$$

$$1 + 2 + \ldots + 2^{29} = (1 + 2 + 2^2)(1 + 2^3)(1 + 2^6(1 + 2^6)(1 + 2^{12}))$$

1. $T_1 \leftarrow A^2$
2. $T_2 \leftarrow T_1^2$
3. $T_1 \leftarrow T_1 \times T_2$
4. $T_2 \leftarrow T_2^2$
5. $T_1 \leftarrow T_1 \times T_2$
6. $T_2 \leftarrow T_1^{2^3}$
7. $T_1 \leftarrow T_1 \times T_2$

How Are Inversions Computed?

From a Decomposition to an Algorithm

$$GF(2^{31}) : A^{-1} = A^{2^{31}-2} = A^{2(2^{30}-1)} = A^{2(1+2+\ldots+2^{29})}$$

$$1 + 2 + \ldots + 2^{29} = (1 + 2 + 2^2)(1 + 2^3)(1 + 2^6\,(1 + 2^6)(1 + 2^{12}))$$

1. $T_1 \leftarrow A^2$
2. $T_2 \leftarrow T_1^2$
3. $T_1 \leftarrow T_1 \times T_2$
4. $T_2 \leftarrow T_2^2$
5. $T_1 \leftarrow T_1 \times T_2$
6. $T_2 \leftarrow T_1^{2^3}$
7. $T_1 \leftarrow T_1 \times T_2$
8. $T_3 \leftarrow T_1$
9. $T_1 \leftarrow T_1^{2^6}$
10. $T_2 \leftarrow T_1^{2^6}$
11. $T_1 \leftarrow T_1 \times T_2$
12. $T_2 \leftarrow T_1^{2^{12}}$
13. $T_1 \leftarrow T_1 \times T_2$
14. $T_1 \leftarrow T_3 \times T_1$

How Are Inversions Computed?

From a Decomposition to an Algorithm

$$GF(2^{31}) : A^{-1} = A^{2^{31}-2} = A^{2(2^{30}-1)} = A^{2(1+2+\ldots+2^{29})}$$

$$1 + 2 + \ldots + 2^{29} = (1 + 2 + 2^2)(1 + 2^3)(1 + 2^6\,(1 + 2^6)(1 + 2^{12}))$$

1. $T_1 \leftarrow A^2$
2. $T_2 \leftarrow T_1^2$
3. $T_1 \leftarrow T_1 \times T_2$
4. $T_2 \leftarrow T_2^2$
5. $T_1 \leftarrow T_1 \times T_2$
6. $T_2 \leftarrow T_1^{2^3}$
7. $T_1 \leftarrow T_1 \times T_2$
8. $T_3 \leftarrow T_1$
9. $T_1 \leftarrow T_1^{2^6}$
10. $T_2 \leftarrow T_1^{2^6}$
11. $T_1 \leftarrow T_1 \times T_2$
12. $T_2 \leftarrow T_1^{2^{12}}$
13. $T_1 \leftarrow T_1 \times T_2$
14. $T_1 \leftarrow T_3 \times T_1$
15. **Return** $T_1 = A^{-1}$

Number of Variables

$(1 + 2^k)$	One short-time variable (T_2)
$(1 + 2^k + 2^{2k})$	One short-time variable (T_2)
$((1 + 2^k)(1 + 2^{2k}) + 2^{4k})$	Two short-time variables (T_2, T_3)
$1 + 2^k(1 + 2^k)$	One short-time variable (T_2) and one long-time variable (T_3 or T_4)

- A short-time variable can be reused by the next term
- A long-time variable must hold its value to the end
- Multiple long-time variables can be accumulated into a single variable $\Rightarrow$ at most one long-time variable is needed

Results

- IT requires 3 variables unless $m-1=2^n$; then it requires 2
- DB requires only 2 variables iff $m-1=2^{n_1}3^{n_2}$
- TB requires either 3 or 4 unless it reduces to DB
- Notably, $162=2\cdot 3^4$ and DB needs only 2 variables

$\Rightarrow$ The DB algorithm achieves the lowest possible memory footprint for inversion in $GF(2^{163})$ used, for example, in operations on popular NIST B/K-163 elliptic curves

Squarings

Motivation

Example

An inversion over $GF(2^{163})$ requires:

- 9 multiplications and
- 162 squarings.

Modern HW implementations of ECC use fast multipliers and squarings start to dominate:

- $M = 163 \Rightarrow$ Squarings take 10% of the time (162 vs. 1467)
- $M = 15 \Rightarrow$ Squarings take 55% of the time (162 vs. 135)
- $M = 4 \Rightarrow$ Squarings take 82% of the time (162 vs. 36)
- $M = 1 \Rightarrow$ Squarings take 95% of the time (162 vs. 9)

OK but the number of squarings is $m - 1 = 162$ for both IT and the new algorithm.

Squarings

Normal Basis

An element $A \in GF(2^m)$ is given by $A = \sum_{i=0}^{m-1} a_i \beta^{2^i}$. Then, $A^{2^s} = A \lll s$ (cyclic shift).

Polynomial Basis

An element $A \in GF(2^m)$ is given by $A = \sum_{i=0}^{m-1} a_i x^i$. Then, $A^2 = \sum_{i=0}^{m-1} a_i x^{2i} \bmod p(x)$ and

$$A^{2^s} = \begin{bmatrix} 1 & q_{0,1}^{(s)} & \cdots & q_{0,m-1}^{(s)} \\ 0 & q_{1,1}^{(s)} & \cdots & q_{1,m-1}^{(s)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & q_{m-1,1}^{(s)} & \cdots & q_{m-1,m-1}^{(s)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{m-1} \end{bmatrix}$$

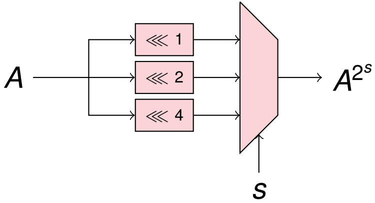
Repeated Squarer (Normal Basis / HW)

A repeated squarer is a component that can compute A^{2^s} for all $s \in \mathcal{S}$ with the same latency (one clock cycle)

- Repeated squarers are simply m-bit C-to-1 multiplexers where C is the cardinality of $\mathcal{S}$

Example

A repeated squarer with $\mathcal{S} = \{1, 2, 4\}$ is a 3-to-1 multiplexer:



Example: The NIST Field $GF(2^{163})$

Itoh-Tsujii

$$1 + 2 + \ldots + 2^{161} = (1+2)(1+2^2(1+2^2)(1+2^4)(1+2^8)(1+2^{16})(1+2^{32}(1+2^{32})(1+2^{64})))$$

$$\Rightarrow \mathcal{E} = (1, 1, 2, 4, 8, 16, 32, 64, 32, 2)$$

DB/TB algorithms

$$1 + 2 + \ldots + 2^{161} = (1+2+2^2)(1+2^3+2^6)(1+2^9+2^{18})(1+2^{27}+2^{54})(1+2^{81})$$

$$\Rightarrow \mathcal{E} = (1, 1, 1, 3, 3, 9, 9, 27, 27, 81)$$

Example: The NIST Field $GF(2^{163})$ (cont.)

With different C, $\mathcal{S}_{\text{opt}}$ and L are as follows:

$\mathcal{E}$	IT $(1, 1, 2, 4, 8, 16, 32, 64, 32, 2)$	DB/TB $(1, 1, 1, 3, 3, 9, 9, 27, 27, 81)$
$C = 1$	$\{1\}, 162$	$\{1\}, 162$
$C = 2$	$\{1, 16\}, 27$	$\{1, 9\}, 26$
$C = 3$	$\{1, 4, 32\}, 17$	$\{1, 3, 27\}, 16$
$C = 4$	$\{1, 2, 8, 32\}, 13$	$\{1, 3, 9, 27\}, 12$
$C = 5$	$\{1, 2, 4, 8, 32\}, 12$	$\{1, 3, 9, 27, 81\}, 10$
$C = 6$	$\{1, 2, 4, 8, 16, 32\}, 11$	—
$C = 7$	$\{1, 2, 4, 8, 16, 32, 64\}, 10$	—

- We have a smaller latency when $C > 1$
- We can use smaller repeated squarers (multiplexers) to get the same latency

Conclusions

A new algorithm for inversion in $GF(2^m)$ that has provably lower number of multiplications compared to the popular IT and outperforms it in about half of the cases for $1 \leq m \leq 1023$

The algorithm has some nice by-products that may be important in many implementations in practice

Conclusions

A new algorithm for inversion in $GF(2^m)$ that has provably lower number of multiplications compared to the popular IT and outperforms it in about half of the cases for $1 \leq m \leq 1023$

The algorithm has some nice by-products that may be important in many implementations in practice

Thank you! Questions?