DIAMANT-Summer School on Elliptic and Hyperelliptic Curve Cryptography

Fast Arithmetic in Finite Fields

Christophe Doche

Macquarie University, Sydney

http://www.ics.mq.edu.au/~doche/

Overview

Introduction

Prime Fields

Extension Fields of Characteristic 2

Optimal Extension Fields

Introduction

Three types of finite fields are relevant in elliptic curve cryptography

- Prime fields $\mathbb{F}_p$
- Extension fields $\mathbb{F}_{2^d}$ with d prime
- Optimal Extension Fields $\mathbb{F}_{p^d}$ with p of size close to the length of a machine word (8, 16, 32 or 64 bits)

Introduction

In each case, we are concerned with the representation of elements and how to perform basic arithmetic

There is a common way of dealing with Finite Fields

In any case, $\mathbb{F}_q^*$ is a cyclic group generated by some element γ

So, an element α can be represented by the power k such that $\alpha = \gamma^k$

Introduction

Multiplication is obvious, but addition relies on a look-up table

To reduce the size of the table, use the notion of Zech's logarithm

Zech's logarithm of γ^k is $Z(k)$ such that

$$\gamma^{Z(k)} = 1 + \gamma^k$$

It works only for small fields

Prime Fields

For prime fields $\mathbb{F}_p$, we work modulo p

An element is represented as an integer in $[0, p-1]$ or in $[\lfloor -p/2 \rfloor, \lfloor p/2 \rfloor]$

Modular reduction is a crucial operation

A Euclidean division does the job but it is a costly operation

Prime Fields

There are several ways to speed-up reductions

One of them is Montgomery method

Let R be the smallest power of 2 bigger than p

Montgomery representation of the integer x in $[0, p-1]$ denoted by $[x]$ is simply $xR \pmod{p}$

Montgomery representation comes with a cheap reduction

Prime Fields

Montgomery reduction of u in $[0, Rp-1]$ is

$$uR^{-1} \pmod{p}$$

This operation denoted by REDC can be performed very efficiently

Precompute $p' = -p^{-1} \pmod{R}$ then

Prime Fields

Algorithm. Montgomery reduction

INPUT: A prime p, R and p', and an integer $u < Rp$.
OUTPUT: The integer $t = \text{REDC}(u) = (uR^{-1}) \bmod p$.

1. $m \leftarrow up' \pmod{R}$
2. $t \leftarrow (u + mp)/R$
3. **if** $t \geqslant p$ **then** $t = t - p$
4. **return** t

Prime Fields

Because the first multiplication is done modulo R, that is a power of 2, compute only the least significant bits

For the second multiplication, we divide by R so only the most significant bits are relevant

In each case, only half a multiplication is required

So the cost of REDC is approximately one field multiplication

Prime Fields

It works since $u + mp$ is divisible by R

$$u + mp \equiv u \pmod{p}$$

$$((u + mp)/R)R \equiv u \pmod{p}$$

So $tR \equiv u \pmod{p}$ and $t \equiv uR^{-1} \pmod{p}$

Finally $u < Rp$ and $m < R$

So $u + mp < Rp + Rp$ and $t < 2p$

Prime Fields

Conversions

To compute $[x]$ simply perform $\text{REDC}(xR_2)$ where $R_2 = R^2 \pmod{p}$ is precomputed

To retrieve x from $[x]$ note that $\text{REDC}([x]) = x$

Prime Fields

Example. Take $p = 2011$, $R = 2048$ so that $R_2 = 1369$ and $p' = 941$

Let $x = 97$ and let us compute $[x]$

$[x] = \textsc{Redc}(97 \times 1369)$

$u = 97 \times 1369 = 132793$

$up' = 124958213$

But we don't perform this multiplication in full

Prime Fields

$up' = (1110111001010110 \;\; 11000000101)_2$

$m \equiv up' \pmod{2048}$

$m = 1541 = (11000000101)_2$

$u{+}mp = 3231744 = (11000101010 \;\; 00000000000)_2$

$(u + mp)/2048 = 1578 = (11000101010)_2$

So $[97] = 1578$

Prime Fields

Another option to have a cheap reduction is to consider special moduli

For instance Mersenne primes, i.e. primes p of the form $2^k - 1$

To reduce $0 \leqslant x < p^2$ write $x = x_1 2^k + x_0$ and

$$x \bmod p = x_1 + x_0$$

Prime Fields

Only problem: Mersenne primes are very rare

There are two possible generalizations

Consider primes $p = 2^k + c$ with c small

If $x = x_1 2^k + x_0$ then $2^k \equiv -c \pmod{p}$ and

$$x \bmod p = x_0 - cx_1$$

They are usually called pseudo-Mersenne numbers

Prime Fields

Or let $w = 16, 32$ or 64 and consider primes

$$p = 2^{n_k w} \pm 2^{n_{k-1} w} \pm \cdots \pm 2^{n_1 w} \pm 1$$

These primes are called Generalized Mersenne Numbers or NIST primes

The fundamental reduction relation is

$$2^{n_k w} \equiv \mp 2^{n_{k-1} w} \mp \cdots \mp 2^{n_1 w} \mp 1 \pmod{p}$$

It can be used recursively to reduce $x < p^2$

Prime Fields

Example. For $p = 2^{192} - 2^{64} - 1$, if $x < p^2$ then x can be written

$$x = x_5 2^{320} + x_4 2^{256} + x_3 2^{192} + x_2 2^{128} + x_1 2^{64} + x_0$$

and

$$\begin{aligned} 2^{320} &\equiv 2^{192} + 2^{128} \pmod{p} \\ 2^{256} &\equiv 2^{128} + 2^{64} \pmod{p} \\ 2^{192} &\equiv 2^{64} + 1 \pmod{p} \end{aligned}$$

Prime Fields

So that in the end

$$x \equiv (x_5 + x_4 + x_2)2^{128} + (x_5 + x_4 + x_3 + x_1)2^{64} + (x_5 + x_3 + x_0) \pmod{p}$$

Prime Fields

Concerning prime field arithmetic

Modular additions and multiplications rely on their integer counterparts

For sizes relevant to ECC, only the schoolbook and Karatsuba methods are really useful

Prime Fields

Karatsuba method

Let b be the radix used to represent integers

If p fits in n words, then write

$$\begin{aligned} uv &= (b^{n/2}U_1 + U_0)(b^{n/2}V_1 + V_0) \\ &= b^n U_1 V_1 + b^{n/2}(U_1 V_0 + U_0 V_1) + U_0 V_0 \end{aligned}$$

Instead of 1 multiplication of size n, it seems we need 4 multiplications of size $n/2$

Prime Fields

But we can do better

Namely, compute U_1V_1, U_0V_0, and $(U_1 + U_0) \times (V_1 + V_0)$

Indeed

$$(U_1V_0+U_0V_1) = (U_1+U_0)\times(V_1+V_0)-U_1V_1-U_0V_0$$

Instead of 4 multiplications of size $n/2$, we just need 3

Prime Fields

Example. Take $u = (1234)_{10}$ and $v = (5678)_{10}$

Set $U_1 = 12$, $U_0 = 34$, $V_1 = 56$, and $V_0 = 78$

Compute

- $U_1V_1 = 12 \times 56 = 672$
- $U_0V_0 = 34 \times 78 = 2652$
- $(U_1+U_0)(V_1+V_0) = (12+34)\times(56+78) = 6164$

Prime Fields

Finally we see that

$$6164 - 672 - 2652 = 2840$$

$$\begin{aligned} uv &= 672 \times 10^4 + 2840 \times 10^2 + 2652 \\ &= 6720000 + 284000 + 2652 \\ &= 7006652 \end{aligned}$$

Prime Fields

Note that the method can be applied recursively

For instance, we can compute 12×56 as

$$\begin{aligned} 12 \times 56 &= 5 \times 10^2 + (33 - 5 - 12) \times 10 + 12 \\ &= 500 + 160 + 12 \\ &= 672 \end{aligned}$$

Prime Fields

Applied recursively, the complexity to multiply two integers with Karatsuba method is

$$O(n^{\log_2 3}) \approx O(n^{1.585})$$

instead of

$$O(n^2)$$

with the schoolbook method

Prime Fields

Inversion is usually done via an extended GCD computation

Or using Fermat's theorem $1/\alpha = \alpha^{p-2}$

Note that there is a dedicated algorithm that is particularly efficient for special types of primes, e.g. Mersenne primes

Prime Fields

Algorithm. Prime field inversion

INPUT: A prime modulus p and an integer x prime to p.
OUTPUT: The integer $x^{-1} \bmod p$.

1. $z \leftarrow x \bmod p$ and $u \leftarrow 1$
2. **while** $z \neq 1$ **do**
3. $\quad q \leftarrow -\lfloor p/z \rfloor$
4. $\quad z \leftarrow p + qz$
5. $\quad u \leftarrow (qu) \bmod p$
6. **return** u

Prime Fields

Exercise. Prove this algorithm returns the inverse of x mod p

Hint: Find a loop invariant

Prime Fields

Arithmetic with Montgomery representation is slightly more complicated

We have $[x] + [y] = [x + y]$

But we don't have $[x][y] = [xy]$

Instead $\textsc{Redc}([x][y]) = [xy]$

Indeed

$$\big(xR \pmod p\big)\big(yR \pmod p\big)R^{-1} \bmod p = xyR \pmod p$$

Prime Fields

Example. Take $p = 2011$, $x = 97$, and $y = 45$

Let us compute $x^2 y \pmod{p}$ using Montgomery representation

$$
\begin{aligned}
[x] &= \textsc{Redc}(97 \times 1369) = 1578 \\
[y] &= \textsc{Redc}(45 \times 1369) = 1668 \\
[x][y] &= 1578 \times 1668 = 2627370 \\
[xy] &= \textsc{Redc}(2627370) = 625 \\
[x^2 y] &= \textsc{Redc}([x][xy]) = \textsc{Redc}(97 \times 625) = 295 \\
x^2 y &= \textsc{Redc}(295) = 1095
\end{aligned}
$$

Prime Fields

There is also a notion of Montgomery inverse namely

$$\mathrm{Inv}(u) = u^{-1}R^2 \bmod p$$

If $u = [x] = xR$ then

$$\mathrm{Inv}([x]) = x^{-1}R \bmod p = [x^{-1}]$$

$$\textsc{Redc}([x]\mathrm{Inv}[x]) = R \bmod p = [1]$$

There is an efficient algorithm to compute the Montgomery inverse

Prime Fields

Square root computation

Legendre symbol allows to decide if an integer a is a square modulo p

In this case, there are explicit formulas to compute a square root x of a for $p \not\equiv 1 \pmod 8$

For instance, if $p \equiv 3 \pmod 4$ then

$$x \equiv \pm a^{(p+1)/4} \pmod p$$

Prime Fields

If a is a square mod p, we have $a^{(p-1)/4} = \pm 1$

If $p \equiv 5 \pmod 8$ and $a^{(p-1)/4} = 1$

Then $x \equiv \pm a^{(p+3)/8} \pmod p$

Now if $a^{(p-1)/4} = -1$

Then $x \equiv \pm 2a(4a)^{(p-5)/8} \pmod p$

Prime Fields

When $p \equiv 1 \pmod 8$ use the following algorithm, which works for any p

Prime Fields

Algorithm. Square root computation

INPUT: A prime p and an square a modulo p.
OUTPUT: An integer x such that $x^2 \equiv a \pmod{p}$.

1. Take b such that $b^2 - 4a$ is not a square modulo p
2. $f(X) \leftarrow X^2 - bX + a$
3. $x \leftarrow X^{(p+1)/2} \pmod{f(X)}$
4. **return** x

Prime Fields

Exercise.

Show the algorithm computes a square root of a

Hint: Show that x *satisfies* $x^2 = a$ *then prove that* $x \in \mathbb{F}_p$

Extension Fields of Characteristic 2

To represent $\mathbb{F}_{2^d}$, use its vector space strucutre over $\mathbb{F}_2$

For any extension, there exist two types of basis

polynomial basis: work modulo an irreducible polynomial P of degree d over $\mathbb{F}_2$

normal basis: work in $(\alpha, \alpha^2, \ldots, \alpha^{2^{d-1}})$ where α is a normal element

Extension Fields of Characteristic 2

Polynomial basis are straightforward to use

Up to isomorphisms, there is only one finite field with cardinality 2^d

So any irreducible polynomial of degree d can be used to construct $\mathbb{F}_{2^d}$

In practice, use sparse polynomials to speed-up the reduction process

Extension Fields of Characteristic 2

Example. To represent $\mathbb{F}_{2^{17}}$ consider

$$\mathbb{F}_2[X]/(X^{17} + X^3 + 1)$$

If we want to compute the square of X^{15}, we need to reduce X^{30} modulo $X^{17}+X^3+1$, i.e. $X^{16}+X^{13}$

Extension Fields of Characteristic 2

If instead we represent $\mathbb{F}_{2^{17}}$ as

$$\mathbb{F}_2[X]/(X^{17}+X^{10}+X^9+X^7+X^6+X^5+1)$$

The reduction of X^{30} modulo this irredcucible polynomials is

$$X^{16}+X^{14}+X^{13}+X^{10}+X^8+X^5+X^3+X^2+X$$

Much more work!

Extension Fields of Characteristic 2

Binomials would be good but except $X + 1$, no binomial is irreducible over $\mathbb{F}_2$

Next best thing are irreducible trinomials and pentanomials when there is no trinomial

Up to degree $10,000$, there is always an irreducible trinomial or a pentanomial to represent $\mathbb{F}_{2^d}$

Extension Fields of Characteristic 2

It is a bit more tricky to find a normal basis

There are several techniques but not all normal bases provide efficient arithmetic

Extension Fields of Characteristic 2

Arithmetic

In polynomial representation

Addition is trivial

Squaring is almost free (simply perform a reduction)

Multiplication relies on polynomial multiplication (school-book or Karatsuba) followed by a reduction

Extension Fields of Characteristic 2

Exercise.

Adapt Karatsuba method to multiply two polynomials in $\mathbb{F}_2[X]$

Test your approach to check that

$$(X^{14} + X^8 + X^2)(X^{20} + X^{10} + X^4 + X) =$$

$$X^{34} + X^{28} + X^{24} + X^{22} + X^{15} + X^9 + X^6 + X^3$$

Extension Fields of Characteristic 2

Inversion: Extended Euclid GCD for polynomials

Or based on Fermat's theorem $\alpha^{2^d-2} = 1/\alpha$

The idea is to write $2^d - 2 = (2^{d-1} - 1)2$ and take advantage of a chain to compute $d-1$ in order to compute $\alpha^{2^{d-1}-1}$

Extension Fields of Characteristic 2

Example. Compute the inverse of $\alpha \in \mathbb{F}_{2^{19}}$, i.e. $\alpha^{2^{19}-2}$

$2^{19} - 2 = 2(2^{18} - 1)$ and

$\alpha^2 \times \alpha = \alpha^3 \quad \alpha^{3\times 2} \times \alpha = \alpha^7 \quad \alpha^{7\times 8} \times \alpha^7 = \alpha^{63}$

$\alpha^{63\times 64} \times \alpha^{63} = \alpha^{4095}$

$\alpha^{4095\times 64} \times \alpha^{63} = \alpha^{2^{18}-1}$

Extension Fields of Characteristic 2

With a normal basis

Addition is trivial

Squaring is totally free (cyclic shift of coordinates)

Multiplication is tricky (uses a precomputed matrix)

Extension Fields of Characteristic 2

Multiplication complexity depends on the number of nonzero coefficients in the matrix (density)

The density is always $\geqslant 2d-1$ and by definition, this bound is attained for an Optimal Normal Basis

There are 2 types of normal bases

But for cryptographic applications, only type II ONB are relevant

Extension Fields of Characteristic 2

Drawbacks

For a given prime degree extension, a type II ONB does not necessarily exist

Even ONBs do not compete with polynomial bases regarding multiplication

Extension Fields of Characteristic 2

Square roots

Every element $\alpha \in \mathbb{F}_{2^d}$ is a square and computing a square root is trivial, e.g. $\sqrt{\alpha} = \alpha^{2^d-1}$

If α is represented by $f(X) = \sum_{i=0}^{d-1} f_i X^i$ on a polynomial basis modulo $m(X)$

Extension Fields of Characteristic 2

Write

$$\sqrt{f(X)} = \sum_{i \text{ even}} f_i X^{i/2} + \sqrt{X} \sum_{i \text{ odd}} f_i X^{\frac{i-1}{2}}$$

where $\sqrt{X}$ has been precomputed modulo $m(X)$

Extension Fields of Characteristic 2

Quadratic equations

Solving $X^2 + aX + b = 0$ in $\mathbb{F}_{2^d}$ with $a \neq 0$ is slightly more complicated

Change of variable $X \leftarrow X/a$ to get the simpler equation

$$X^2 + X = c \quad \text{with } c = b/a^2$$

Extension Fields of Characteristic 2

Exercise. Show the following

This equation has a solution in $\mathbb{F}_{2^d}$ if and only if $\mathrm{Tr}(c) = 0$

If x is a solution then $x + 1$ is the other one

If d is odd, one as

$$x = \sum_{i=0}^{(d-3)/2} c^{2^{2i+1}}$$

Optimal Extension Fields

Take an extension field $\mathbb{F}_{p^d}$ such that

- the characteristic p fits in a machine word and allows a fast reduction in $\mathbb{F}_p$
- the irreducible polynomial defining $\mathbb{F}_{p^d}$ allows a fast polynomial reduction

Optimal Extension Fields

More precisely, an OEF is an extension field $\mathbb{F}_{p^d}$ where

- p is a pseudo-Mersenne prime, that is $p = 2^n + c$ with c small

- there is an irreducible binomial $m(X) = X^d - \omega$ over $\mathbb{F}_p$

Optimal Extension Fields

Arithmetic

Let $\alpha, \beta \in \mathbb{F}_{p^d}$ represented by $\alpha = \sum_{i=0}^{d-1} a_i X^i$ and $\beta = \sum_{i=0}^{d-1} b_i X^i$ where a_i and b_i are in $\mathbb{F}_p$

Then using the relation $X^d \equiv \omega \pmod{m(X)}$, one has

$$\alpha\beta = c_{d-1} + \sum_{k=0}^{d-2} (c_k + \omega c_{d+k}) X^k$$

with $c_k = \sum_{i=0}^{k} a_i b_{k-i}$

Optimal Extension Fields

Exponentiation

Computed faster by the use of the Frobenius

The coefficients of $\alpha = \sum_{j=0}^{d-1} a_j X^j$ are in $\mathbb{F}_p$, so there is an explicit formula for α^{p^i}

$$\alpha^{p^i} = \sum_{j=0}^{d-1} a_j \omega^{\lfloor jp^i/d \rfloor} X^{jp^i \bmod d}$$

Optimal Extension Fields

Inversion relies on the following observation

$$\alpha^{-1} = \alpha^{r-1}\alpha^{-r} \quad \text{with} \quad r = (p^d - 1)/(p - 1)$$

Now, it is easy to check that $\alpha^r \in \mathbb{F}_p$ and α^{r-1} is obtained via an exponentiation

Optimal Extension Fields

There are explicit formulas for degrees 3 and 5 for multiplications and inversions

For instance, inversion in $\mathbb{F}_{p^5}$ is obtained via the explicit formula

$$\alpha^{-1} = \frac{(\alpha^p \alpha^{p^2} (\alpha^p \alpha^{p^2})^{p^2}}{\alpha\big(\alpha^p \alpha^{p^2} (\alpha^p \alpha^{p^2})^{p^2}\big)}$$

It works since $\alpha\big(\alpha^p \alpha^{p^2} (\alpha^p \alpha^{p^2})^{p^2}\big) \in \mathbb{F}_p$