Finite fields

Christophe Ritzenthaler

1 Construction and operations

In this section we describe a method for constructing finite fields.

1.1 Definitions

Définition 1.1. *Let p be a prime and n a positive integer n. We call the* finite field *over $\mathbb{Z}/p\mathbb{Z}$ of degree n the decomposition field of the polynomial $X^{p^n} - X$. We denote this field $\mathbb{F}_{p^n}$.*

It can be shown that $\mathbb{F}_{p^n}$ is the only field with p^n elements up to isomorphism. In particular, if f is an irreducible polynomial of degree n over $\mathbb{F}_p$, it defines an extension of $\mathbb{F}_p$ of degree n, $\mathbb{F}_p[X]/f$, which can be identified to $\mathbb{F}_{p^n}$.

Exemple 1. *The polynomial $X^2 + X + 1$ is irreducible over $\mathbb{F}_2$ (otherwise it would have a root in $\mathbb{F}_2$). It defines then the extension $\mathbb{F}_4$.*

Proposition 1.1. *The extension $\mathbb{F}_{p^n}/\mathbb{F}_p$ is Galois with Galois group isomorphic to $\mathbb{Z}/n\mathbb{Z}$. A generator of this group is the automorphism $x \mapsto x^p$.*

1.2 Irreducible polynomials

Rabin's test relies on the following theorem.

Proposition 1.2. *If f is of degree n and irreducible then $f | X^{p^n} - X$.*

Démonstration. Let F be the extension of $\mathbb{F}_p$ defined by f. As F is Galois, f has all its roots in F. As f is irreducible, these roots have multiplicity one and we know that they are zeros of $X^{p^n} - X$. So $f | X^{p^n} - X$. □

So we can apply the following algorithm :

1. Test if $f | X^{p^n} - X$. If not f is not irreducible.
2. for each prime q dividing n compute $\gcd(f, X^{p^{n/q}} - X)$. f is irreducible if and only if all these numbers are equal to 1.

Exemple 2. *Let $p = 2$ and $f(X) = X^8 + X^4 + X^3 + X + 1$. We can check that $f | X^{2^8} - X$ and $\gcd(f, X^{2^4} - X) = 1$.*

1.3 Basic operations

The elements of the field are residue classes modulo f. The construction of those residue classes corresponds to the construction of residue classes in $\mathbb{Z}$. By euclidian division, a representative of the class is a polynomial of degree less than $\deg(f)$. Addition and multiplication can be done imitating the process in $\mathbb{Z}/n\mathbb{Z}$.

Exemple 3. *Representatves of the residue classes in* $\mathbb{F}_2[X]/X^2+X+1$ *are* $0, 1, X, 1+X$. *We obtain the following addition and multiplication tables.*

$+$	0	1	X	$X+1$
0	0	1	X	$X+1$
1	1	0	$X+1$	X
X	X	$X+1$	0	1
$X+1$	$X+1$	X	1	0

$\times$	1	X	$X+1$
1	1	X	$X+1$
X	X	$X+1$	1
$X+1$	$X+1$	1	X

We can also compute the inverse of an element using Bézout theorem. Indeed, since f is irreducible, for any $g \in \mathbb{F}_p[X]$ of degree less than $\deg(f)$, f and g are coprime. There exist then (u, v) such that $uf + vg = 1$. Thus v is the inverse of g modulo f.

Exemple 4. *Let* $p = 2$ *and* $f(X) = X^8 + X^4 + X^3 + X + 1$. *We determine the inverse of* $X + 1$ *modulo* f. *We have*

$$f(X) = (X+1)(X^7 + X^6 + X^5 + X^4 + X^2 + X) + 1$$

so the class of $X^7 + X^6 + X^5 + X^4 + X^2 + X$ *is an inverse of* $X + 1$ *modulo* f.

1.4 Primitive element

Recall that the group $\mathbb{F}_q^*$ is cyclic.

Définition 1.2. *A generator of* $\mathbb{F}_q^*$ *is called a* primitive element *of* $\mathbb{F}_q$.

There is no deterministic algorithm to produce a primitive element without assuming the Generalized Riemann Hypothesis. However, in practice it is not a problem as they are $\phi(q-1)$ of them which is a large proportion of the elements (greater than $e^{-\gamma}/\log\log(p-1)$ for p big enough). Still in order to check that a random element is a primitive element, one has to compute its order and for this one needs to know the factorization of $q-1$, which can be hard to obtain. Indeed assume that the decomposition of $q-1 = \prod q_i^{\alpha_i}$. We pick a a and for each i we compute $a^{(q-1)/q_i}$. If this element is different from 1 this means that $q_i^{\alpha_i}$ divides the order of a. Indeed if m is the order of a and if $q_i^{\beta_i} | a$ with $\beta_i < \alpha_i$ then

$$a^{(q-1)/q_i} = (a^n)^{(q-1)/(nq_i)} = 1.$$

As these numbers are coprime, if all $a^{(q-1)/q_i}$ are different from 1 then a is a primitive element, otherwise we pick another a.

1.5 Frobenius morphism

Définition 1.3. *Let* $\mathbb{F}_{p^b}/\mathbb{F}_{p^a}$ *be a field extension (i.e.* $a|b$*). The* Frobenius morphism *of* $\mathbb{F}_{p^b}$ *relative to* $\mathbb{F}_{p^a}$ *is the morphism* $F_{p^a} : x \to x^{p^a}$.

This is an important morphism : basic Galois theory implies that $x \in \mathbb{F}_{p^b}$ is in $\mathbb{F}_{p^a}$ if and only if $F_{p^a}(x) = x$. It is important de be able to compute its action in a fast way. One possibility is to consider a basis of $\mathbb{F}_{p^b}/\mathbb{F}_{p^a}$ generated by $\beta, \beta^{p^a}, \beta^{p^{2a}}, \ldots, \beta^{p^{(b-1)a}}$ (this is called a normal basis and always exists).

1.6 Square root computations

We will restrict to the case where $q = p$ prime.

1.6.1 When $p = 4k - 1$

In this case let $a \equiv x^2 \pmod{p}$. Let us check that $a^{\frac{p+1}{4}}$ is a square root of a :

$$(a^{\frac{p+1}{4}})^2 \equiv a^{(p-1)/2} \cdot a \equiv x^{p-1} \cdot a \equiv a \pmod{p}.$$

1.6.2 When $p = 4k + 1$: Shanks algorithm

Let us write $p - 1 = 2^s t$ with t odd and $s \geq 2$. Let $a \equiv x^2 \pmod{p}$. Let assume that we know a b which is not a quadratic residue modulo p. Let us write $z = b^t$.
Set $B = a^t$, $X = a^{(t+1)/2}$, $Y = z$ and $R = s - 1$ and run the following algorithm.

```
while R ≥ 1 do
      if B^{2^{R-1}} ≡ 1 (mod p) then
         Y := Y^2 ;
      else
         B := BY^2 ;
         X := XY ;
         Y := Y^2 ;
      end if ;
      R := R − 1 ;
end while ;
```

One can check that the output X is a square root of a. To do so, we check that the following conditions are loop invariants :

$$\begin{cases} aB & = X^2 \\ Y^{2^R} & \equiv -1 \pmod{p} \\ B^{2^R} & \equiv 1 \pmod{p} \\ R & \geq 0 \end{cases}$$

As initial conditions

$$\begin{cases} aB & = aa^t = a^{t+1} = (a^{(t+1)/2})^2 \\ Y^{2^R} & \equiv z^{2^{s-1}} \equiv b^{(p-1)/2} \equiv -1 \pmod{p} \\ B^{2^R} & \equiv a^{t2^{s-1}} \equiv a^{(p-1)/2} \equiv 1 \pmod{p} \\ R = s - 1 & \geq 0 \end{cases}$$

After a loop, let us denote B', X', Y', R' the new values.

— if $B^{2^{R-1}} \equiv 1 \pmod p$, then

$$\begin{cases} aB' & = aB = a^{t+1} = (a^{(t+1)/2})^2 \\ (Y')^{2^{R'}} & = Y^{2^R} \equiv -1 \pmod p \\ (B')^{2^{R'}} & \equiv B^{2^{R-1}} \equiv 1 \pmod p \\ R' = R-1 & \geq 0 \end{cases}$$

— if $B^{2^{R-1}} \equiv -1 \pmod p$, then

$$\begin{cases} aB' & = aBY^2 = X^2Y^2 = X'^2 \\ (Y')^{2^{R'}} & = Y^{2^R} \equiv -1 \pmod p \\ (B')^{2^{R'}} & \equiv B^{2^{R-1}}Y^{2^R} \equiv 1 \pmod p \\ R' = R-1 & \geq 0 \end{cases}$$

Let us point out that this algorithm gives the correct answer as long as we have picked a non quadratic residue b. It is then a Las Vegas algorithm.

1.7 Some other facts

Polynomial functions over finite fields : be careful that there is no bijection with the polynomials (the kernel is generated by the $x_i^q - x_i$).

Order of certain matrix groups over finite fields : a classical question is about the order of $\mathrm{GL}_n(\mathbb{F}(q))$.

Let $\mathbb{F}_{q^n}/\mathbb{F}_q$ be a field extension. One define the trace of an element $x \in \mathbb{F}_{q^n}$ as $x + x^q + \ldots x^{q^{n-1}}$. One can show that this is a surjective linear form onto $\mathbb{F}_q$ (shows that there is an element with non-zero trace). Similarly the norm $x \cdot x^q \cdots x^{q^{n-1}}$ is a surjective morphism between the multiplicative groups (show that it sends a primitive element onto a primitive element).

2 Discrete Logarithms

Let $(G, \times)$ be a finite cyclic group of order n, α a generator. We recall that the DLP is to find x in the equation $\alpha^x = y$.

We will present several attacks on the DLP. The first four are exponential. The last one is subexponential. Note that the biggest DLP which has been solved is in $\mathbb{F}_{2^{613}}$.

2.1 Enumeration

The simplest method for computing the DL x from $\alpha^x = y$ in G is to test whether $x = 0, 1, 2, \ldots$ satisfies the equation. Of course, as soon as the size of the group is important (60 bits), this method is not possible anymore.

2.2 Shanks Baby-Step Giant-Step

We set $m = \lceil \sqrt{n} \rceil$ and write $x = qm + r$ with $0 \leq r, q < m$. We have

$$\alpha^{qm+r} = y \Rightarrow (\alpha^m)^q = y\alpha^{-r}.$$

First we compute the set of *baby-steps* (**pas de bébé**)

$$B = \{(y\alpha^{-r}, r),\ 0 \leq r < m\}.$$

If we find a pair $(1, r)$ then $y = \alpha^r$. If we do not find such a pair, we determine $\delta = \alpha^m$. Then we test for $q = 1, 2, \ldots, m$ whether the element δ^q is the first component of an element of B. As soon as it is true we have a solution for the DLP. The elements δ^q are called *giant steps* (**pas de géant**).
It is easy to see that this algorithm is in $\mathcal{O}(\sqrt{\#G})$. Note that it requires also a storage for $\mathcal{O}(\sqrt{\#G})$ elements.

2.3 The Pollard ρ-algorithm

This algorithm has the same running time as the previous one but it only requires constant storage.
We need a partition $G_1 \amalg G_2 \amalg G_3 = G$. Let $f : G \to G$ be defined by

$$f(\beta) = \begin{cases} \alpha\beta & \text{if } \beta \in G_1, \\ \beta^2 & \text{if } \beta \in G_2, \\ y\beta & \text{if } \beta \in G_3. \end{cases}$$

We choose a random $x_0 \in \{1, \ldots, n\}$ and compute $\beta_0 = \alpha^{x_0}$. Then we compute the sequence

$$\beta_{i+1} = f(\beta_i).$$

The elements of this sequence can be written as

$$\beta_i = \alpha^{x_i} y^{\delta_i}$$

where $\delta_0 = 0$ and

$$x_{i+1} = \begin{cases} x_i + 1 \pmod n & \text{if } \beta_i \in G_1, \\ 2x_i \pmod n & \text{if } \beta_i \in G_2, \\ x_i & \text{if } \beta_i \in G_3, \end{cases}$$

and

$$\delta_{i+1} = \begin{cases} \delta_i \pmod n & \text{if } \beta_i \in G_1, \\ 2\delta_i \pmod n & \text{if } \beta_i \in G_2, \\ \delta_i + 1 & \text{if } \beta_i \in G_3. \end{cases}$$

At some points, two elements in the sequence (β_i) must be equal, say $\beta_{i+k} = \beta_i$. This implies

$$\alpha^{x_i} y^{\delta_i} = \alpha^{x_{i+k}} y^{\delta_{i+k}}$$

and therefore

$$\alpha^{x_i - x_{i+k}} = y^{\delta_{i+k} - \delta_i}.$$

We obtain a congruence

$$x_i - x_{i+k} \equiv x(\delta_{i+k} - \delta_i) \pmod n.$$

The solution is unique if $\delta_{i+k} - \delta_i$ is invertible modulo n. If the solution is not unique then the discrete logarithm can be found by testing the different possibilities modulo n. If there are too many possibilities then the algorithm is applied with a different x_0.

We estimate the number of β_i that must be computed before a match is found. By the birthday paradox (see **??**) if we compute $\mathcal{O}(\sqrt{\#G})$ elements then a match is found with a probability greater than $1/2$.
Thus far, our algorithm is less good than the previous one. The advantage is that we do not need to store as many elements. Initially (β_1, x_1, δ_1) is stored. Now suppose that at a certain point in the algorithm (β_i, x_i, δ_i) is stored. Then (β_j, x_j, δ_j) is computed for $j = i+1, i+2, \ldots$ until either a match is found or $j = 2i$. In the latter case we delete β_i and store β_{2i}. Hence we only store the triplets with $i = 2^k$. This works for the following reason : the sequence (β_i) is periodic after a certain number s of iterations (with the first match as end point). If l is the length of the period then if $2^j \geq \max(s, l)$ then a period is contained in the interval $[2^j, \ldots, 2^{j+1}]$ and a match can be found.

Remarque 1. *In [?], it is showed that the Pollard Rho method for finding the discrete logarithm on a cyclic group G requires $O(\sqrt{|G|}(\log |G|)^{3/2})$ steps until a collision occurs and discrete logarithm is possibly found.*

2.4 The Pohlig-Hellman algorithm

We now show that the DLP can be reduced to DLPs in cyclic groups of prime order if we know the factorization

$$n = \#G = \prod_p p^{e(p)}.$$

1. Reduction to prime powers. For each prime divisor p of n, we set

   $$n_p = n/p^{e(p)}, \quad \alpha_p = \alpha^{n_p}, \quad y_p = y^{n_p}.$$

   Then the order of α_p is exactly $p^{e(p)}$ and

   $$\alpha_p^x = y_p.$$

   Assume we can solve the DLP in the prime powers subgroups and call $x(p)$ the results. Then the Chinese Remainder Theorem shows that x is the unique solution of the congruences

   $$x \equiv x(p) \pmod{p^{e(p)}}.$$

2. Reduction to prime order. Let now assume that $\#G = p^e$ for a prime p. We want to solve the DLP in this group. We have $x < p^e$ so let us write (in base p)

   $$x = x_0 + x_1 p + \ldots x_{e-1} p^{e-1}, \quad 0 \leq x_i < p, \quad 0 \leq i \leq e-1.$$

   We show that the x_i are DLP in a group of order p. Indeed, one has

   $$p^{e-1} x = x_0 p^{e-1} + p^e (x_1 + x_2 p + \ldots + x_{e-1} p^{e-2}).$$

   Now

   $$(\alpha^{p^{e-1}})^{x_0} = y^{p^{e-1}}.$$

   This equation shows that x_0 is the DL in a group of order p. The other coefficients are determined recursively. Suppose that $x_0, \ldots, x_{i-1}$ have been determined. Then

   $$\alpha^{x_i p^i + \ldots + x_{e-1} p^{e-1}} = y x^{-(x_0 + x_1 p + \ldots + x_{i-1} p^{i-1})}.$$

Denote the right-hand side by y_i, one has by raising to the power p^{e-i-1}

$$(\alpha^{p^{e-1}})^{x_i} = y_i^{p^{e-i-1}}.$$

We have then reduce the problem of the DLP in G to e DLPs in a group of prime order.

3. Prime order. One applies one of the two previous algorithms (i.e. 2.2 or 2.3).

We see easily that the running time is dominated by the square root of the largest prime divisor of $\#G$.

Exemple 5. *Let us solve* $5^x \equiv 3 \pmod{2017}$. *The order of the multiplicative group is* $n = 2016 = 2^5 \cdot 3^2 \cdot 7$. *First we determine* $x(2) \equiv x \pmod{2^5}$. *We obtain* $x(2)$ *as a solution of the congruence*

$$(5^{3^2 \cdot 7})^{x(2)} \equiv 3^{3^2 \cdot 7} \pmod{2017}.$$

To solve this congruence, we write

$$x(2) = x_0(2) + \ldots + x_4(2) \cdot 2^4.$$

The coefficient $x_0(2)$ *is solution of*

$$2016^{x_0(2)} \equiv 1 \pmod{2017}.$$

We obtain $x_0(2) = 0$. *Now* $y_1 = y$. *Then* $x_1(2)$ *is solution of*

$$2016^{x_1(2)} \equiv 2016 \pmod{2017}.$$

We obtain $x_1(2) = 1$ *and* $y_2 \equiv 1579 \pmod{2017}$. *Hence* $x_2(2)$ *is solution of*

$$2016^{x_2(2)} \equiv 2016 \pmod{2017}.$$

We obtain $x_2(2) = 1$ *and* $y_3 \equiv 1 \pmod{2017}$ *so* $x_3(2) = x_4(2) = 0$. *Concluding those computations, we obtain* $x(2) = 6$.
Now we compute

$$x(3) = x_0(3) + x_1(3) \cdot 3.$$

We obtain $x_0(3)$ *as the solution of*

$$294^{x_0(3)} \equiv 294 \pmod{2017},$$

so $x_0(3) = 1$ *and* $y_1 \equiv 294 \pmod{2017}$. *Hence* $x_1(3) = 1$ *and* $x(3) = 4$.
Finally we compute $x(7)$ *as the solution of the congruence*

$$1879^{x(7)} \equiv 1879 \pmod{2017},$$

so $x(7) = 1$. *We obtain* x *as the solution of the simultaneous congruence*

$$x \equiv 6 \pmod{32}, \quad x \equiv 4 \pmod{9}, \quad x \equiv 1 \pmod{7}.$$

The solution is $x = 1030$.

2.5 Index calculus

When $G = (\mathbb{Z}/n\mathbb{Z})^*$ or more generally the unit group of a finite field, there are more efficient DL algorithms, the so called *index calculus* algorithms (**méthode de l'index**). They are closely related to integer factoring algorithms such as the quadratic sieve. We describe a simple index calculus algorithm.

The idea. Let p be a prime number, α a primitive element modulo p and $y \in \{1, \ldots, p-1\}$. We want to solve $\alpha^x \equiv y \pmod{p}$. We choose a bound B and determine the set

$$F(B) = \{q \in \mathbb{P},\ q \leq B\}.$$

This is the factor base. An integer b is called B-smooth if it has only prime factor in $F(B)$. We proceed in two steps. First we compute the discrete logarithm of the factor base elements, i.e. we solve

$$\alpha^{x(q)} \equiv q \pmod{p}$$

for all $q \in F(B)$. Then we determine an exponent $\delta \in \{1, \ldots, p-1\}$ such that $y\alpha^\delta \pmod{p}$ is B-smooth. We obtain

$$y\alpha^\delta \equiv \prod_{q \in F(B)} q^{e(q)} \pmod{p}.$$

Together

$$y\alpha^\delta \equiv \prod_{q \in F(B)} q^{e(q)} \equiv \prod_{q \in F(B)} \alpha^{x(q)e(q)} \equiv \alpha^{\sum_{q \in F(B)} x(q)e(q)} \pmod{p},$$

and hence

$$y = \alpha^{\sum_{q \in F(B)} x(q)e(q) - \delta} \pmod{p}.$$

Therefore,

$$x \equiv \sum_{q \in F(B)} x(q)e(q) - \delta \pmod{p-1}. \tag{1}$$

Discrete logarithms of the factor base elements. To compute the discrete logarithms of the factor base elements, we choose random numbers $z \in \{1, \ldots, p-1\}$ and compute $\alpha^z \pmod{p}$. We check whether those numbers are B-smooth. If they are, we compute the decomposition

$$\alpha^z \pmod{p} = \prod_{q \in F(B)} q^{f(q,z)}.$$

Each exponent vector $(f(q,z))_{q \in F(B)}$ is called a *relation*. If we find as many relations as there are factor base elements, then we try to find the discrete logarithms by solving a linear system. We obtain

$$\alpha^z \equiv \prod_{q \in F(B)} q^{f(q,z)} \equiv \prod_{q \in F(B)} \alpha^{x(q)f(q,z)} \equiv \alpha^{\sum_{q \in F(B)} x(q)f(q,z)} \pmod{p}.$$

This implies

$$z \equiv \sum_{q \in F(B)} x(q)f(q,z) \pmod{p-1}$$

for all z, so each relation yields one linear congruence. The system is solved with standards methods.

Individual logarithms. If the discrete logarithm of the factor base elements are computed, then the discrete logarithm of y to the base α is determined. We choose a random $\delta \in \{1, \ldots, p-1\}$. If $y\alpha^{\delta}$ is B-smooth, then 1 is applied. Otherwise, we choose a new δ.

Remarque 2. *It can be shown that the running time is $L_p(1/2, C)$ for some constant C. In principle the index calculus algorithm works in any group. However the factor base must be chosen such that relations can be found efficiently. (Un)fortunetaly, for some groups, such that elliptic curves over finite fields, it is not known how to choose the factor base and how to compute relations.*