

$\dots, x_{\alpha-1}$. Namely, for each $i = 1, 2, \dots, \alpha - 1$ set

$$y_i = y/b^{x_0+x_1p+\dots+x_{i-1}p^{i-1}},$$

which has discrete log congruent $\text{mod } p^\alpha$ to $x_i p^i + \dots + x_{\alpha-1} p^{\alpha-1}$. Since y_i is a p^i -th power, we have $y_i^{(q-1)/p^i} = 1$ and $y_i^{(q-1)/p^{i+1}} = b^{(x_i+x_{i+1}p+\dots)(q-1)/p} = b^{x_i(q-1)/p} = r_{p,x_i}$. So we set x_i equal to the value of j for which $y_i^{(q-1)/p^{i+1}} = r_{p,j}$.

When we are done we will have $x \text{ mod } p^\alpha$. After doing this for each $p|q-1$, we finally use the Chinese Remainder Theorem to find x .

This algorithm works well when all of the primes dividing $q-1$ are small. But clearly the computation of the table of $\{r_{p,j}\}$ and the comparison of the $y_i^{(q-1)/p^{i+1}}$ with this table will take a long time if $q-1$ is divisible by a large prime. (By “large” we mean of at least about 20 digits. If $p|q-1$ is smaller than about 10^{20} , then one can combine the Silver–Pohlig–Hellman algorithm with Shanks’ “giant step — baby step” method; see pp. 9, 575–576 of Knuth, Vol. 2.)

Example 4. Find the discrete log of 28 to the base 2 in \mathbf{F}_{37}^* using the Silver–Pohlig–Hellman algorithm. (2 is a generator of \mathbf{F}_{37}^* .)

Solution. Here $37-1=2^2 \cdot 3^2$. We compute $2^{18} \equiv 1 \pmod{37}$, and so $r_{2,0}=1$, $r_{2,1}=-1$. (For $p=2$, always $\{r_{2,j}\}=\{\pm 1\}$.) Next, $2^{36/3} \equiv 26$, $2^{2 \cdot 36/3} \equiv 10 \pmod{37}$, and so $\{r_{3,j}\}=\{1, 26, 10\}$. Now let $28 \equiv 2^x \pmod{37}$. We first take $p=2$ and find $x \text{ mod } 4$, which we write as $x_0 + 2x_1$. We compute $28^{36/2} \equiv 1 \pmod{37}$, and hence $x_0=0$. We then compute $28^{36/4} \equiv -1 \pmod{37}$, and hence $x_1=1$, i.e., $x \equiv 2 \pmod{4}$. Next we take $p=3$ and find $x \text{ mod } 9$, which we write as $x_0 + 3x_1$. (Of course, for each p the x_i are defined differently.) To find x_0 , we compute $28^{36/3} \equiv 26 \pmod{37}$, and so $x_0=1$. We then compute $(28/2)^{36/9} = 14^4 \equiv 10 \pmod{37}$; thus, $x_1=2$, and so $x \equiv 1+2 \cdot 3 = 7 \pmod{9}$. It remains to find the unique $x \text{ mod } 36$ such that $x \equiv 2 \pmod{4}$ and $x \equiv 7 \pmod{9}$. This is $x=34$. Thus, $28=2^{34}$ in \mathbf{F}_{37}^* .

The index–calculus algorithm for discrete logs. The reader may want to skip this subsection for now, or read it lightly, and come back to it for a closer examination while reading §V.3, since the index–calculus algorithm for computing discrete logs in finite fields has much in common with the factor–base method for factoring large integers.

Here we shall suppose that $q=p^n$ is a fairly large power of a small prime p , and b is a generator of \mathbf{F}_q^* . The index–calculus algorithm finds for any $y \in \mathbf{F}_q^*$ the value of $x \text{ mod } q-1$ such that $y=b^x$.

Let $f(X) \in \mathbf{F}_p[X]$ be any irreducible polynomial of degree n ; then \mathbf{F}_q is isomorphic to the residue ring $\mathbf{F}_p[X]/f(X)$. Any element $a \in \mathbf{F}_q = \mathbf{F}_p[X]/f(X)$ can be written (uniquely) as a polynomial $a(X) \in \mathbf{F}_p[X]$ of degree at most $n-1$. In particular, our base $b=b(X)$ is such a polynomial. The “constants” are the elements of $\mathbf{F}_p \subset \mathbf{F}_q$.