The Pohlig-Hellman Algorithm

D. R. Stinson

We provide here a bit of further explanation concerning the workings of the Pohlig-Hellman algorithm. We use the same notation as in the text: p is prime, α is a primitive element in $\mathbb{Z}_p$, and $\beta \in {\mathbb{Z}_p}^*$. Our goal is to determine $a = \log_\alpha \beta$, where, without loss of generality, $0 \leq a \leq p-2$.

The prime power factorization of $p-1$ is

$$p - 1 = {p_1}^{c_1} {p_2}^{c_2} \ldots {p_k}^{c_k},$$

where the p_i's are distinct primes. The main step is to compute $a \bmod {p_i}^{c_i}$, $1 \leq i \leq k$. So suppose that $q = p_i$ and $c = c_i$ for some i, $1 \leq i \leq k$. We will show how to compute $x = a \bmod q^c$.

First, x is expressed as

$$x = \sum_{i=0}^{c-1} a_i q^i,$$

where $0 \leq a_i \leq q-1$ $(0 \leq i \leq c-1)$. From this it follows that

$$a = a_0 + a_1 q + \ldots + a_{c-1} q^{c-1} + s q^c,$$

for some integer s.

The computation of a_0 follows from the fact that

$$\beta^{\frac{p-1}{q}} \equiv \alpha^{\frac{a_0(p-1)}{q}} \bmod p. \tag{1}$$

Here is a proof of Equation (1) that is simpler than the one given in the text:

$$\begin{aligned}
\beta^{\frac{p-1}{q}} &\equiv (\alpha^a)^{\frac{p-1}{q}} \bmod p \\
&\equiv \left(\alpha^{a_0 + a_1 q + \ldots + a_{c-1} q^{c-1} + s q^c}\right)^{\frac{p-1}{q}} \bmod p \\
&\equiv \left(\alpha^{a_0 + Kq}\right)^{\frac{p-1}{q}} \bmod p \qquad \text{(where } K \text{ is an integer)} \\
&\equiv \alpha^{\frac{a_0(p-1)}{q}} \alpha^{K(p-1)} \bmod p \\
&\equiv \alpha^{\frac{a_0(p-1)}{q}} \bmod p.
\end{aligned}$$

From this, it is a simple matter to determine a_0.

The next step would be to compute $a_1, \ldots, a_{c-1}$ (if $c > 1$). These computations can be done from a suitable generalization of Equation (1).

First, define $\beta_0 = \beta$, and

$$\beta_j = \beta\alpha^{-(a_0+a_1q+\ldots+a_{j-1}q^{j-1})} \bmod p,$$

for $0 \leq j \leq c-1$. We make use of the following generalization of Equation (1):

$$(\beta_j)^{\frac{p-1}{q^{j+1}}} \equiv \alpha^{\frac{a_j(p-1)}{q}} \bmod p. \tag{2}$$

(Observe that when $j = 0$, Equation (2) reduces to Equation (1).)

The proof of Equation (2) is much the same as that of Equation (1):

$$\begin{aligned}
(\beta_j)^{\frac{p-1}{q^{j+1}}} &\equiv \left(\alpha^{a-\left(a_0+a_1q+\ldots+a_{j-1}q^{j-1}\right)}\right)^{\frac{p-1}{q^{j+1}}} \bmod p \\
&\equiv \left(\alpha^{a_jq^j+\ldots+a_{c-1}q^{c-1}+sq^c}\right)^{\frac{p-1}{q^{j+1}}} \bmod p \\
&\equiv \left(\alpha^{a_jq^j+K_jq^{j+1}}\right)^{\frac{p-1}{q^{j+1}}} \bmod p \qquad \text{(where } K_j \text{ is an integer)} \\
&\equiv \alpha^{\frac{a_j(p-1)}{q}}\alpha^{K_j(p-1)} \bmod p \\
&\equiv \alpha^{\frac{a_j(p-1)}{q}} \bmod p.
\end{aligned}$$

Hence, given β_j, it is straightforward to compute a_j from Equation (2).

To complete the description of the algorithm, it suffices to observe that β_{j+1} can be computed from β_j by means of a simple recurence relation, once a_j is known. This follows from the following relation, which is proved easily:

$$\beta_{j+1} = \beta_j\alpha^{-a_jq^j} \bmod p. \tag{3}$$

Now, we can compute a_0, β_1, a_1, β_2, $\ldots$, β_{c-1}, a_{c-1} by alternately applying Equations (2) and (3).