Discrete logarithm and related cryptosystems - III

Diego F. Aranha

Institute of Computing
UNICAMP

Pohlig-Hellman Algorithm

As before, suppose we have a group $\mathbb{G}$ and elements $\alpha, \beta \in \mathbb{G}$ such that α has order $n > 1$ and $\beta \in \langle \alpha \rangle$. Our task is to find $a = \log_\alpha \beta$, which is unique modulo n.

Pohlig-Hellman Algorithm

As before, suppose we have a group $\mathbb{G}$ and elements $\alpha, \beta \in \mathbb{G}$ such that α has order $n > 1$ and $\beta \in \langle \alpha \rangle$. Our task is to find $a = \log_\alpha \beta$, which is unique modulo n.

Now let

- $n = \prod_{i=1}^{k} p_i^{c_i}$ be the factorization of n, and
- suppose that we can find the values

$$a \bmod p_i^{c_i}, 1 \leq i \leq k;$$

Pohlig-Hellman Algorithm

As before, suppose we have a group $\mathbb{G}$ and elements $\alpha, \beta \in \mathbb{G}$ such that α has order $n > 1$ and $\beta \in \langle \alpha \rangle$. Our task is to find $a = \log_\alpha \beta$, which is unique modulo n.

Now let

- $n = \prod_{i=1}^{k} p_i^{c_i}$ be the factorization of n, and
- suppose that we can find the values

$$a \bmod p_i^{c_i}, 1 \leq i \leq k;$$

then, a mod n can be computed from these values using the Chinese Remainder Theorem. See Thm. 2.31 on the IMC book.

Pohlig-Hellman Algorithm

As before, suppose we have a group $\mathbb{G}$ and elements $\alpha, \beta \in \mathbb{G}$ such that α has order $n > 1$ and $\beta \in \langle \alpha \rangle$. Our task is to find $a = \log_\alpha \beta$, which is unique modulo n.

Now let

- $n = \prod_{i=1}^{k} p_i^{c_i}$ be the factorization of n, and
- suppose that we can find the values

$$a \bmod p_i^{c_i}, 1 \leq i \leq k;$$

then, a mod n can be computed from these values using the Chinese Remainder Theorem. See Thm. 2.31 on the IMC book.

The overall time complexity of the algorithm would be $O\left(\sum_{i=1}^{k} S(p_i^{c_i}) + \log n\right)$, where

- $S(p_i^{c_i})$ is the complexity of finding a mod $p_i^{c_i}$, and
- $\log N$ is the cost of the Chinese Remainder Algorithm.

Pohlig-Hellman Algorithm

Finding $a = \log_\alpha \beta \bmod q^c$, q prime

Suppose now that we are given a group $\mathbb{G}$ with order q^c, for q prime, and generator α for $\mathbb{G}$. We would like to find $a = \log_\alpha \beta$, which is unique modulo q^c.

Pohlig-Hellman Algorithm

Finding $a = \log_\alpha \beta \bmod q^c$, q prime

Suppose now that we are given a group $\mathbb{G}$ with order q^c, for q prime, and generator α for $\mathbb{G}$. We would like to find $a = \log_\alpha \beta$, which is unique modulo q^c.

We reduce this problem to that of finding discrete logs in a prime order group. So, we assume that an algorithm for this task exists.

Pohlig-Hellman Algorithm

Finding $a = \log_\alpha \beta \bmod q^c$, q prime

Suppose now that we are given a group $\mathbb{G}$ with order q^c, for q prime, and generator α for $\mathbb{G}$. We would like to find $a = \log_\alpha \beta$, which is unique modulo q^c.

We reduce this problem to that of finding discrete logs in a prime order group. So, we assume that an algorithm for this task exists.

First express $x = a \bmod q^c$ in "base" q as the sum

$$x = x_0 + x_1 q + x_2 q^2 + \ldots + x_{c-1} q^{c-1}.$$

Pohlig-Hellman Algorithm

Finding $a = \log_\alpha \beta \bmod q^c$, q prime

Suppose now that we are given a group $\mathbb{G}$ with order q^c, for q prime, and generator α for $\mathbb{G}$. We would like to find $a = \log_\alpha \beta$, which is unique modulo q^c.

We reduce this problem to that of finding discrete logs in a prime order group. So, we assume that an algorithm for this task exists.

First express $x = a \bmod q^c$ in "base" q as the sum

$$x = x_0 + x_1 q + x_2 q^2 + \ldots + x_{c-1} q^{c-1}.$$

Now note that, since $\alpha^{q^c} = 1$, $\alpha^{q^{c-1}}$ has order q, since

$$1 = \alpha^{q^c} = (\alpha^{q^{c-1}})^q,$$

and any smaller power of $\alpha^{q^{c-1}}$ cannot divide q, a prime number.

Pohlig-Hellman Algorithm

Finding $a = \log_\alpha \beta \bmod q^c$, q prime

We now find $x_0, x_1, \ldots$ in order, using the following procedure.

Compute

$$\begin{aligned}
\beta^{q^{c-1}} &= (\alpha^x)^{q^{c-1}} \\
&= (\alpha^{x_0 + x_1 q + x_2 q^2 + \ldots + x_{c-1} q^{c-1}})^{q^{c-1}} \\
&= \alpha^{x_0 q^{c-1}} (\alpha^{q^c})^{x_1 + x_2 q + x_3 q^2 + \ldots + x_{c-1} q^{c-2}} \\
&= \alpha^{x_0 q^{c-1}} \\
&= (\alpha^{q^{c-1}})^{x_0}
\end{aligned}$$

Pohlig-Hellman Algorithm

Finding $a = \log_\alpha \beta \bmod q^c$, q prime

We now find $x_0, x_1, \ldots$ in order, using the following procedure.

Compute

$$\begin{aligned}
\beta^{q^{c-1}} &= (\alpha^x)^{q^{c-1}} \\
&= (\alpha^{x_0 + x_1 q + x_2 q^2 + \ldots + x_{c-1} q^{c-1}})^{q^{c-1}} \\
&= \alpha^{x_0 q^{c-1}} (\alpha^{q^c})^{x_1 + x_2 q + x_3 q^2 + \ldots + x_{c-1} q^{c-2}} \\
&= \alpha^{x_0 q^{c-1}} \\
&= (\alpha^{q^{c-1}})^{x_0}
\end{aligned}$$

Since $\alpha^{q^{c-1}}$ has prime order q, the equation

$$(\alpha^{q^{c-1}})^{x_0} = \beta^{q^{c-1}}$$

can be solved for x_0, by our initial assumption.

Pohlig-Hellman Algorithm

Finding $a = \log_\alpha \beta \bmod q^c$, q prime

Similarly, we now proceed to find x_1. Compute

$$\begin{aligned}
\beta^{q^{c-2}} &= (\alpha^x)^{q^{c-2}} \\
&= \left(\alpha^{x_0 + x_1 q + x_2 q^2 + \ldots + x_{c-1} q^{c-1}}\right)^{q^{c-2}} \\
&= \alpha^{x_0 q^{c-2}} \alpha^{x_1 q^{c-1}} \left(\alpha^{q^c}\right)^{x_2 + x_3 q + x_4 q^2 + \ldots + x_{c-1} q^{c-3}} \\
&= \alpha^{x_0 q^{c-2}} \alpha^{x_1 q^{c-1}}
\end{aligned}$$

Pohlig-Hellman Algorithm

Finding $a = \log_\alpha \beta \bmod q^c$, q prime

Similarly, we now proceed to find x_1. Compute

$$\begin{aligned}
\beta^{q^{c-2}} &= (\alpha^x)^{q^{c-2}} \\
&= (\alpha^{x_0+x_1 q+x_2 q^2+\ldots+x_{c-1}q^{c-1}})^{q^{c-2}} \\
&= \alpha^{x_0 q^{c-2}} \alpha^{x_1 q^{c-1}} (\alpha^{q^c})^{x_2+x_3 q+x_4 q^2+\ldots+x_{c-1}q^{c-3}} \\
&= \alpha^{x_0 q^{c-2}} \alpha^{x_1 q^{c-1}}
\end{aligned}$$

Since we already know x_0, we can solve the discrete log equation for x_1:

$$(\alpha^{q^{c-1}})^{x_1} = (\beta \cdot \alpha^{-x_0})^{q^{c-2}}.$$

Similarly, knowing x_0 and x_1, we can solve for x_2:

$$(\alpha^{q^{c-1}})^{x_2} = (\beta \cdot \alpha^{-x_0-x_1 q})^{q^{c-3}}.$$

Pohlig-Hellman Algorithm

Finding $a = \log_\alpha \beta \bmod q^c$, q prime

Similarly, we now proceed to find x_1. Compute

$$\begin{aligned}
\beta^{q^{c-2}} &= (\alpha^x)^{q^{c-2}} \\
&= (\alpha^{x_0+x_1 q+x_2 q^2+\ldots+x_{c-1}q^{c-1}})^{q^{c-2}} \\
&= \alpha^{x_0 q^{c-2}} \alpha^{x_1 q^{c-1}} (\alpha^{q^c})^{x_2+x_3 q+x_4 q^2+\ldots+x_{c-1}q^{c-3}} \\
&= \alpha^{x_0 q^{c-2}} \alpha^{x_1 q^{c-1}}
\end{aligned}$$

Since we already know x_0, we can solve the discrete log equation for x_1:

$$(\alpha^{q^{c-1}})^{x_1} = (\beta \cdot \alpha^{-x_0})^{q^{c-2}}.$$

Similarly, knowing x_0 and x_1, we can solve for x_2:

$$(\alpha^{q^{c-1}})^{x_2} = (\beta \cdot \alpha^{-x_0-x_1 q})^{q^{c-3}}.$$

We repeat this procedure, solving equations

$$(\alpha^{q^{c-1}})^{x_i} = (\beta \cdot \alpha^{-x_0-x_1 q-\ldots-x_{i-1}q^{i-1}})^{q^{c-i-1}},$$

for $i = 3, \ldots, c-1$, thus determining $x = a \bmod q^c$ entirely.

Pohlig-Hellman

Algorithm

Input: $\mathbb{G}, n, \alpha, \beta, q, c$

1. $j \leftarrow 0$
2. $\beta_j \leftarrow \beta$
3. **While** $j < c$
   - $\delta \leftarrow \beta_j^{n/q^{j+1}}$
   - Compute $a_j = \log_{\alpha^{n/q}} \delta$
   - $\beta_{j+1} \leftarrow \beta_j \alpha^{-a_j q^j}$
   - $j \leftarrow j + 1$
4. **Return** $(a \bmod q^c)$ from $(a_0, \ldots, a_{c-1})$

Invariant: $\beta_j^{n/q^{j+1}} = \alpha^{a_j n/q}$. See Example 2.35 from the IMC book.

Important: Cost of c_i discrete logarithms for each subgroup of order p_i with $\sqrt{p_i}$ iterations. Choose $p - 1 = 2q$, with prime q as a defense.

The Index Calculus Method

This is a subexponential algorithm, that works only for computing discrete logarithms $\log_\alpha \beta$ over $\mathbb{Z}_p^*$, for p prime and α a primitive element modulo p.

It assumes the existence of a *factor base* $\mathcal{B} = \{p_1, p_2, \ldots, p_B\}$, that is, a set of small primes.

The first step (precomputing phase) in the algorithm is to obtain discrete logarithms of the B primes $p_1, p_2, \ldots, p_B$ in the factor base.

The second step is computing the logarithm of an element β using the factor-base discrete logarithms.

The Index Calculus Method

In the precomputing phase, $C > B$ congruences of the form

$$\alpha^{x_j} \equiv p_1^{a_{1,j}} p_2^{a_{2,j}} \cdots p_B^{a_{B,j}} \pmod{p}$$

are constructed.

Alternatively, taking logarithms on both sides gives

$$x_j \equiv a_{1,j} \log_\alpha p_1 + \cdots + a_{B,j} \log_\alpha p_B \pmod{p-1}.$$

Given the C congruences, we can solve this system for the unknowns $\log_\alpha p_i$, with $1 \leq i \leq B$. This has to be done using the Chinese Remainder Theorem.

The Index Calculus Method

Now, given the discrete logarithms in the factor base, choose a random integer $1 \leq s < p - 1$ and compute $\gamma = \beta\alpha^s \bmod p$.

The Index Calculus Method

Now, given the discrete logarithms in the factor base, choose a random integer $1 \leq s < p-1$ and compute $\gamma = \beta\alpha^s \bmod p$.

If it is possible to factor γ using only primes in the base $\mathcal{B}$, we obtain

$$\beta\alpha^s \equiv p_1^{c_1} p_2^{c_2} \dots p_B^{c_B} \pmod{p}, \text{ or,}$$

$$\log_\alpha \beta + s \equiv c_1 \log_\alpha p_1 + \dots + c_B \log_\alpha p_B \pmod{p-1}.$$

Since all other terms are known, we can find $\log_\alpha \beta$.

See example 3.5.8 from the IMC book.

The Index Calculus Method

Now, given the discrete logarithms in the factor base, choose a random integer $1 \leq s < p-1$ and compute $\gamma = \beta\alpha^s \bmod p$.

If it is possible to factor γ using only primes in the base $\mathcal{B}$, we obtain

$$\beta\alpha^s \equiv p_1^{c_1} p_2^{c_2} \dots p_B^{c_B} \pmod{p}, \text{ or,}$$

$$\log_\alpha \beta + s \equiv c_1 \log_\alpha p_1 + \dots + c_B \log_\alpha p_B \pmod{p-1}.$$

Since all other terms are known, we can find $\log_\alpha \beta$.

See example 3.5.8 from the IMC book.

Important: Subexponential complexity because it reduces to integer factoring. There are limitations when applying the approach to elliptic curves.