Cryptanalytic world records, summer 2014

Discrete Logarithms

Dr. Daniel Loebenberger

ALGORITHM. Baby-step giant-step algorithm for the discrete logarithm.

Input: A cyclic group $G = \langle g \rangle$ with d elements, and a group element $x \in G$.
Output: $\mathrm{dlog}_g\, x$.

1. $m \longleftarrow \lceil \sqrt{d}\, \rceil$.
2. Baby steps: compute and store $x, xg, xg^2, \ldots, xg^m$ in a table.
3. Giant steps: compute $g^m = xg^m \cdot x^{-1}$, g^{2m}, $g^{3m}, \ldots$ until one of them, say g^{im}, equals an element in the table, say xg^j.
4. Return $im - j$ in $\mathbb{Z}_d$.

Example

We take a group G with $d = 20$ elements. We might have $G = \mathbb{Z}_{20}$ with addition, or $G = \mathbb{Z}_{25}^{\times}$ with multiplication, since $\phi(25) = 4 \cdot 5$. Let us take the latter representation. Now $g = 2 \in G = \mathbb{Z}_{25}^{\times}$ is a generator, since $2^{20/2} = 2^{10} = 24 \neq 1$ and $2^{20/5} = 2^4 = 16 \neq 1$ in G. In order to compute the discrete logarithm of $x = 17$, we have $m = \lceil \sqrt{20} \rceil = 5$, and perform the following computations.

k	baby steps xg^k	giant steps g^{km}
0	17	1
1	9	7
2	18	24
3	11	18
4	22	1
5		7
		$\cdots$

In the third giant step, we find the collision $xg^2 = 18 = g^{3 \cdot 5}$, and hence $\mathrm{dlog}_2\, 17 = 3 \cdot 5 - 2 = 13$. We check that indeed $2^{13} = 17$ in $\mathbb{Z}_{25}^{\times}$.

Theorem

For any group G with d elements, the baby-step giant-step method solves DL_G with at most $2m$ group operations and space for m elements of G, where $m = \lceil \sqrt{d}\, \rceil$.

Algorithm. Birthday algorithm for discrete logarithm.

Input: A cyclic group $G = \langle g \rangle$ with d elements, and a group element $x \in G$.

Output: $\mathrm{dlog}_g\, x$.

1. $X, Y \longleftarrow \emptyset$.
2. Do step 3 until a collision of X and Y occurs.
3. Choose uniformly at random a bit $b \xleftarrow{} \{0,1\}$ and $i \xleftarrow{} \{0,\ldots,d-1\}$. Add xg^i to X if $b = 0$ and g^i to Y if $b = 1$, and remember the index i.
4. If $xg^i = g^j$ for some $xg^i \in X$ and $g^j \in Y$, then return $j - i$ in $\mathbb{Z}_d$.

Theorem

The algorithm works correctly as specified. Its expected time is $O(\sqrt{d}\log d)$ multiplications in G, with expected space for $O(\sqrt{d})$ elements of G.

We have a cyclic group $G = \langle g \rangle$ with d elements, and an element $x = g^{a}$ of G. Our task is to calculate $a = \mathrm{dlog}_g\, x$ from g and x. Choose a sequence $b_0, b_1, \ldots \longleftarrow \{0, 1, 2\}$ of uniformly and independently distributed random "trits" b_k, choose u_0, $v_0 \longleftarrow \mathbb{Z}_d$ at random and start with $y_0 = x^{u_0} g^{v_0}$. Then we calculate $y_1, y_2, \ldots$ in G by

$$y_k = \begin{cases} x \cdot y_{k-1} & \text{if } b_{k-1} = 0, \\ y_{k-1}^2 & \text{if } b_{k-1} = 1, \\ g \cdot y_{k-1} & \text{if } b_{k-1} = 2, \end{cases}$$

until we find a collision $y_i = y_j$ with $i \neq j$.

ALGORITHM. The Pollard rho algorithm for discrete logarithms.

Input: A cyclic group $G = \langle g \rangle$ of order d, a partition $G = S_0 \cup S_1 \cup S_2$ into three disjoint parts of roughly equal size, and $x \in G$.

Output: $\text{dlog}_g\, x$, or "failure".

1. Define the iteration function $\mathcal{P}$ by $\mathcal{P}(z, \rho) = (z^*, \rho^*)$, where $z, z^* \in G$, $\rho, \rho^* \in \mathbb{Z}_d[t]$, and
$$z^* = \begin{cases} x \cdot z & \text{if } z \in S_0, \\ z^2 & \text{if } z \in S_1, \\ g \cdot z & \text{if } z \in S_2. \end{cases} \qquad \rho^* = \begin{cases} \rho + t & \text{if } z \in S_0, \\ 2\rho & \text{if } z \in S_1, \\ \rho + 1 & \text{if } z \in S_2. \end{cases}$$
2. $u_0, v_0 \longleftarrow \mathbb{Z}_d$, x_0, $y_0 \longleftarrow x^{u_0} g^{v_0}$, σ_0, $\tau_0 \longleftarrow u_0 t + v_0$, $k \longleftarrow 0$.
3. Do step 4 until $x_k = y_k$.
4. $k \longleftarrow k + 1$. Calculate $x_k, y_k \in G$ and $\sigma_k, \tau_k \in \mathbb{Z}_d[t]$ by
$$(x_k, \sigma_k) \leftarrow \mathcal{P}(x_{k-1}, \sigma_{k-1}), \quad (y_k, \tau_k) \leftarrow \mathcal{P}(\mathcal{P}(y_{k-1}, \tau_{k-1})).$$
5. Let $\sigma_k = ut + v$ and $\tau_k = u't + v'$, with $u, u', v, v' \in \mathbb{Z}_d$. If $\gcd(u - u', d) = 1$ in $\mathbb{Z}$, then return $(v' - v) \cdot (u - u')^{-1}$ in $\mathbb{Z}_d$, else return "failure".

Theorem

Let G be a cyclic group of order d. Then the Pollard rho algorithm, with Floyd's trick, finds a discrete logarithm in G with an expected number of $O(\sqrt{d})$ group operations, provided that the sequence $x_0, x_1, x_2, \ldots$ behaves randomly. Space is required for two elements of G and four elements of $\mathbb{Z}_d$.

Example

We have $g = 2 \in G = \mathbb{Z}_{25}^{\times}$, $d = 20$, and $x = 17$. As suggested above, we use the partition $S_0 = \{1, 2, 3, 4, 6, 7, 8\}$, $S_1 = \{9, 11, 12, 13, 14, 16, 17\}$, and $S_2 = \{18, 19, 21, 22, 23, 24\}$ of G, with $7, 7$, and 6 elements, respectively. Our random choice is $u_0 = 12$ and $v_0 = 7$, so that $\sigma_0 = 12t + 7$.

k	x_k	y_k	σ_k	τ_k
0	8	8	$12t + 7$	$12t + 7$
1	6	23	$4t + 16$	$9t + 14$
2	23	6	$9t + 14$	8
3	16	16	$10t + 14$	$2t + 18$

We find the collision $x_3 = y_3 = 16$ with $\sigma_3 = 10t + 14$ and $\tau_3 = 2t + 18$ in $\mathbb{Z}_{20}[t]$. Then $u - u' = 10 - 2$ and $w = \gcd(8, 20) = 4 \neq 1$. The algorithm as stated returns “failure”.

Example

But we persist and compute $a = \text{dlog}_2 17$ as a root of $\sigma_3 - \tau_3$. Namely, $d' = d/w = 20/4 = 5$, and dividing $\sigma_3 - \tau_3$ by 4, we have

$$(10t + 14 - (2t + 18))/4 = (8t + 16)/4 = 2t + 4.$$

The quantity called u in the Remark is now $\tilde{u} = 8$, and $(\tilde{u}/w)^{-1} = (8/4)^{-1} = 2^{-1} = 3$ in $\mathbb{Z}_{d'} = \mathbb{Z}_5$. Then

$$b = \frac{-v}{w} \cdot (\frac{u}{w})^{-1} = \frac{-16}{4} \cdot 3 = -12 = 3 \text{ in } \mathbb{Z}_5,$$

and the possible values for a are $a = b + id' = 3 + 5i$ in $\mathbb{Z}_{20}$, for $0 \leq i < 4$. Thus $a \in \{3, 8, 13, 18\}$, and a check reveals that $\text{dlog}_2 17 = 13$.

Lemma

Suppose that $d = q_1 q_2$ with coprime q_1 and q_2, and that $a_i = \mathrm{dlog}_{g_i} x_i$ in $\mathbb{Z}_{d/q_i}$, where $g_i = g^{d/q_i}$ and $x_i = x^{d/q_i} \in \pi_{d/q_i}(G)$, for $i = 1, 2$. Then $\mathrm{dlog}_g x = a_i$ in $\mathbb{Z}_{q_i}$ for $i = 1, 2$.

$$
\begin{tikzcd}
 & & \mathbb{Z}_d \arrow[dll, "\mu_{e_1}"'] \arrow[drr, "\mu_{e_2}"] \arrow[dd, shift right, "\mathsf{dexp}_g"'] & & \\
\mathbb{Z}_{q_1} \arrow[dd, shift right, "\mathsf{dexp}_{g_1}"'] & & & & \mathbb{Z}_{q_2} \arrow[dd, shift right, "\mathsf{dexp}_{g_2}"'] \\
 & & G \arrow[uu, shift right, "\mathsf{dlog}_g"'] \arrow[dll, "\pi_{e_1}"] \arrow[drr, "\pi_{e_2}"'] & & \\
\pi_{e_1}(G) \arrow[uu, shift right, "\mathsf{dlog}_{g_1}"'] & & & & \pi_{e_2}(G) \arrow[uu, shift right, "\mathsf{dlog}_{g_2}"']
\end{tikzcd}
$$

Example

We have $G = \mathbb{Z}_{25}^{\times} = \langle 2 \rangle$ with $d = \#G = 20 = 4 \cdot 5$, so that $q_1 = 4$ and $q_2 = 5$, and $x = 17 \in G$. Additively, μ_5 maps $\mathbb{Z}_{20}$ to $5 \cdot \mathbb{Z}_{20} = \{0, 5, 10, 15, 20, \ldots, 95\} = \{0, 5, 10, 15\} \cong \mathbb{Z}_4$ as a subgroup of $\mathbb{Z}_{20}$. Multiplicatively, we have $g_1 = 2^{20/4} = 7$ and $g_2 = 2^{20/5} = 16$, and the two subgroups

$$S_1 = \langle 2^{20/4} \rangle = \{1, 7, 24, 18\} \text{ and } S_2 = \langle 2^{20/5} \rangle = \{1, 16, 6, 21, 11\}$$

have 4 and 5 elements, respectively. The Chinese remainder algorithm for the discrete logarithm of 17 first computes the two constituents of x in S_1 and S_2: $x_1 = 17^{20/4} = 7$ and $x_2 = 17^{20/5} = 21$. We can read off the discrete logarithms in S_1 and S_2: $a_1 = \mathrm{dlog}_{g_1} x_1 = 1$ and $a_2 = \mathrm{dlog}_{g_2} x_2 = 3$. With the Chinese Remainder Algorithm, we find $a = 13$, which satisfies $a = 1$ in $\mathbb{Z}_4$ and $a = 3$ in $\mathbb{Z}_5$. We are quite happy to have found the same result as with baby and giant steps and Pollard's rho method.

Lemma

Let $d = q_1 \cdots q_r$ be a factorization of $d = \#G$ into pairwise coprime factors, with $G = \langle g \rangle$ a cyclic group as above, let $x \in G$ and for $i \leq r$, let $S_i = \{x^{d/q_i} \colon x \in G\}$ and $T_i = \{x \in G \colon x^{q_i} = 1\}$. Then the following hold.

1. $S_i = T_i$ is a subgroup with q_i elements, generated by g^{d/q_i}, and the map

$$\begin{aligned} G &\to S_1 \times S_2 \times \cdots \times S_r, \\ y &\mapsto (y^{d/q_1}, \ldots, y^{d/q_r}), \end{aligned}$$

   is an isomorphism.

2. If $x = g^{a}$, $i \leq r$, and $a = a_i$ in $\mathbb{Z}_{q_i}$, then $x^{d/q_i} = (g^{d/q_i})^{a_i}$.
3. If $a_i = \operatorname{dlog}_{g^{d/q_i}} x^{d/q_i}$ and $a \in \mathbb{Z}_d$ satisfies $a = a_i$ in $\mathbb{Z}_{q_i}$ for all $i \leq r$, then $a = \operatorname{dlog}_g x$.

Algorithm. Chinese remaindering for discrete logarithms.

Input: A cyclic group $G = \langle g \rangle$ of order $d = \#G$, and $x \in G$.
Output: $a = \mathrm{dlog}_g x$.

1. Compute the prime power factorization of d.
2. For each $i \leq r$, do steps 2 and 3.
3. Compute $g_i = g^{d/q_i}$ and $x_i = x^{d/q_i}$, with the repeated squaring.
4. Compute the discrete logarithm $a_i = \mathrm{dlog}_{g_i} x_i \in \mathbb{Z}_{q_i}$ in $S_i = \langle g_i \rangle$.
5. Combine these "small" discrete logarithms via the Chinese Remainder Theorem to find the unique $a \in \mathbb{Z}_d$ so that $a = a_i$ in $\mathbb{Z}_{q_i}$ for all $i \leq r$.

Theorem

Let G be a cyclic group of n-bit order d. Then Algorithm computes discrete logarithms in G at the following cost:

1. factoring the integer d,
2. one discrete logarithm in each of the groups $S_1, \ldots, S_r$,
3. $O(n^2)$ operations in G,
4. $O(n^2)$ bit operations.

$$
\begin{tikzcd}
 & G \arrow[dl, "\pi_5"'] \arrow[dr, "\pi_4"] & \\
S_1 \cong \mathbb{Z}_4 \arrow[d, "\pi_2"'] & & S_2 \cong \mathbb{Z}_5 \\
S_{10} \cong \mathbb{Z}_2 & &
\end{tikzcd}
$$

ALGORITHM. Pohlig-Hellman.

Input: A cyclic group $G = \langle g \rangle$ with p^e elements, where p is a prime and $e \geq 2$ an integer, and $x \in G$.

Output: $\mathrm{dlog}_g\, x$.

1. Compute $h = g^{p^{e-1}}$ and set $y_{-1} = 1 \in G$.
2. For i from 0 to $e-1$ do steps 3 – 5.
3. $\quad x_i \leftarrow (x \cdot y_{i-1})^{p^{e-i-1}}$. [Then $x_i \in H = \langle h \rangle$.]
4. $\quad a_i \leftarrow \mathrm{dlog}_h\, x_i$.
5. $\quad y_i \leftarrow y_{i-1} \cdot g^{-a_i p^i}$.
6. Return $a = a_{e-1}p^{e-1} + \cdots + a_0$.

Theorem

The algorithm correctly computes $\mathsf{dlog}_g\, x$. It uses $O(e^2 \log p)$ operations in G, plus e calls to a subroutine for discrete logarithms in the group H with p elements.

Example

We illustrate the Pohlig-Hellman algorithm in an example with $p^e = 3^4 = 81$. The group G is the subgroup $G = \langle 4 \rangle \subseteq \mathbb{Z}_{163}^\times$ generated by $g = 4$. We note that 163 is prime and $\# G_{163}^\times = \phi(163) = 162 = 2 \cdot 81$. Furthermore, $2^2 = 4$ and $2^{81} = -1$ in $\mathbb{Z}_{163}$. Thus 2 is a generator of $\mathbb{Z}_{163}^\times$, so that the order of 4 in $\mathbb{Z}_{163}^\times$ is 81. We have $p = 3$, $e = 4$, and $H = \langle 4^{27} \rangle = \langle h \rangle = \{1, 104, 58\}$ with $h = 104$. We trace the computation of the discrete logarithm $a = \mathsf{dlog}_4 60 = 2 \cdot 3^3 + 0 \cdot 3^2 + 1 \cdot 3 + 2 = 59$ of $x = 60 = 4^{59}$. Discrete logarithms in H are found by inspection.

$$x_0 = x^{p^3} = x^{27} = 58 \in H, a_0 = \mathsf{dlog}_h x_0 = 2, y_0 = y_{-1} \cdot g^{-a_0} = 51,$$

$$x_1 = (xy_0)^9 = 104 \in H, a_1 = \mathsf{dlog}_h x_1 = 1, y_1 = y_0 \cdot g^{-a_1 \cdot 3} = 39,$$

$$x_2 = (xy_1)^3 = 1 \in H, a_2 = \mathsf{dlog}_h x_2 = 0, y_2 = y_1 \cdot g^{-a_2 \cdot 9} = 39,$$

$$x_3 = xy_2 = 58 \in H, a_3 = \mathsf{dlog}_h x_3 = 2.$$

We now have computed
$a = a_3 p^3 + a_2 p^2 + a_1 p + a_0 = 2 \cdot 27 + 0 \cdot 9 + 1 \cdot 3 + 2 \cdot 1 = 59.$