Baby-stp giant-step attack

Tanja Lange

Eindhoven University of Technology

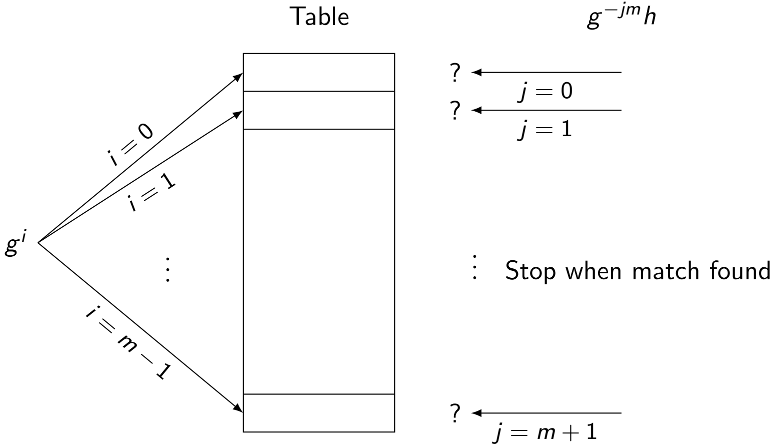
2WF80: Introduction to Cryptology

Attack on the discrete-logarithm problem

Let $g \in G$ with $\mathrm{ord}(g) = \ell$. Let $m = \lfloor \sqrt{\ell} \rfloor$. Let $h = g^a$ for secret a.

Write $a = a_0 + a_1 m$ and split the search space.
(Typical meet-in-the-middle attack).

Search for a match of g^i with $g^{-jm}h$, find it at at $i = a_0, j = a_1$.



Stop when match found

Adapted from Jérémy Jeam

The BSGS algorithm

Baby steps

- Compute table with (g^i, i) for $0 \leq i < m$;
- Sort by first element while computing.

Preparation

- Reach g^m, invert: $S = g^{-m}$.

Giant steps

- Starting at $j = 0, 1, 2, 3, \ldots$,
  compute $S^j h$ and compare with table entries.
  Match instantly gives $g^{-jm} h = g^i$,
  thus $a = i + jm$.

Cost

- Each BS or GS costs 1 MULT.
- Total cost $(\leq 2m + 2)$ MULTs $+1$INV.

The BSGS algorithm

Baby steps

- Compute table with (g^i, i) for $0 \leq i < m$;
- Sort by first element while computing.

Preparation

- Reach g^m, invert: $S = g^{-m}$.

Giant steps

- Starting at $j = 0, 1, 2, 3, \ldots$,
  compute $S^j h$ and compare with table entries.
  Match instantly gives $g^{-jm} h = g^i$,
  thus $a = i + jm$.

Cost

- Each BS or GS costs 1 MULT.
- Total cost $(\leq 2m + 2)$ MULTs $+1$INV.

Example: see exercise sheet.
Optimizations
Using $g^{jm} h$ in the giant steps avoids inversion
but needs reduction mod ℓ to get the result.

Can optimize by interleaving baby and giant steps
(needs $\log_2 n$ MULTs for exponentiation again).