A $O(\sqrt{p}\log p)$ Algorithm for Discrete Log

Bill Gasarch and Douglas Ulrich

January 10, 2014

Let p be a prime; let g be a generator for Z_p^*. (Z_p^* is the set $\{1, 2, \ldots, p-1\}$.) All arithmetic will be mod p. We consider p and g to be fixed and known. We also think of p as being large so any computation taking p steps is not feasible.

A MULTIPLICATION or COMPARISON between two elements in Z_p^* takes roughly $O(\log p)$ or $O(\log^2 p)$. We ignore such factors until the very end and just call them STEPS.

The Discrete Log Problem: Given x find $L \in [0, p-1]$ such that $g^L = x$.

We could do this problem as follows:

1. For $L = 0$ to $p - 1$
   (a) If $g^L = x$ then BREAK

   Answer is L

This takes $O(p)$ steps and hence is not feasible. Can we do better? YES- we will show a way to do this problem in $O(\sqrt{p}\log p)$.

Let $m = \mathsf{Floor}(\sqrt{p-1})$. Here is the KEY IDEA. We know that $0 \le L \le p-1$. If L was divided by m it would be $L = am + b$ where $0 \le b \le m - 1$. (this is standard division) but ALSO note that $0 \le a \le m$ since $L \le p - 1$. Hence our goal is to find a and b.

The algorithm is in two phases. In phase ONE we do some preprocessing (computations independent of x). Note that if you need to find many discrete logs using p, g then the preprocessing need only be done once.

PHASE ONE:

1. For $i = 1$ to m compute g^i. From a list $(1, g^1)$, $(2, g^2)$, ..., (m, g^m). (This step takes $O(m) = O(\sqrt{p})$ steps.)
2. SORT the list of ordered pairs based on the SECOND coordinate. (This step takes $O(m \log m) = O(\sqrt{p}\log p)$ steps. We call this ordered list THE TABLE.
3. For $i = 1$ to m compute g^{im}. From a list $(1, g^m)$, $(2, g^{2m})$, ..., (m, g^{mm}). (This step takes $O(m) = O(\sqrt{p})$ steps.)

4. For $i = 1$ to m compute g^{-im}. From a list $(1, g^{-m})$, $(2, g^{-2m})$, $\ldots$, (m, g^{-mm}). (This step takes $O(m) = O(\sqrt{p})$ steps.)

PHASE ONE takes $O(\sqrt{} \log p)$ steps.

PHASE TWO (we now use x).

First the intuition. We want a, b such that $g^{am+b} = x$. And recall that $0 \le a \le m-1$. Lets say a was the answer. Then $xg^{-am} = g^b$ where $0 \le b \le m-1$. We will TRY all such a. Note that there are only m of them.

1. For $i = 0$ to m
   (a) Compute $z = xg^{-am}$.
   (b) Look for z on the TABLE (the TABLE is sorted so this only takes $O(\log m) = O(\log p)$ steps). Note that you may or may not find it.
   (c) IF we find z on the TABLE then we have found (b, z) so we know that $z = g^b$ where $0 \le b \le m-1$. We KNOW that $DL(x) = am + b$. Here is why:

$$\begin{aligned} z &= xg^{-am} \\ g^b &= xg^{-am} \\ g^{am+b} &= x \end{aligned}$$

The loop has at most m iterations and each one takes $O(\log p)$ steps. So PHASE TWO takes $O(m \log p) = O(\sqrt{p} \log p)$ steps.

Phase ONE and TWO together take $O(\sqrt{p} \log p)$ steps. Each step is at most $(\log p)^2)$ real steps. So the algorithm is $O(\sqrt{p} \log^3 p)$ steps.

1. This works really well in practice.
2. The log factors do not matter in practice.
3. This is called baby-step giant-step algorithm. I'll let you figure out why.