GEORGIA INSTITUTE OF TECHNOLOGY
School of Electrical and Computer Engineering

ECE 6280
Cryptography

Due: Thursday, 5:00PM, April 24, 2013

Solving Discrete Log in a Cyclic Group using the Index calculus algorithm

In this project, you need to solve the discrete log problem as followings: given that $b = g^a \ mod \ p$ for some $1 < a < q$ find "a" provided that you know "b", "g" and "p".

Let q be a prime. Let p also be a prime such that $p = kq + 1$ where k is an even number. Define a set $G = \{x^k \bmod p \ \text{ for all } \ x \in \mathbb{Z}_p^*\}$. You can easily verify that the size of the set is equal to q. You can also verify that the set G is a cyclic group of order q (Note that although the group order is q, the multiplication in the group is mod p). To find the generator of the group, we pick a random number $x \in \mathbb{Z}_p^*$ and compute $g = x^k \bmod p$. If $g \neq 1$, then you can show that g is the generator of the group G. We study the discrete log problem over this cyclic group G with the generator g.

Below, I provided the primes p (about 31 bits), q (about 23 bits) and the generator of the group G: These are the parameters you will need to use in your final algorithm. Again, $p = kq + 1$, Use $p = 2382933803$ and $q = 10930889$, and $k = 218$. According to the above discussion, the elements of the cyclic group G (of order q) is constructed by taking any x in Z_p^* and computing $y = x^k \ mod \ p$. I have found the generator of the cyclic group G as $g = 2084483647$ (by choosing $x = 2$). The generator g has order q.

For testing your algorithm, I will be sending you the parameter b and will ask you to report me a, where $g^a = b \bmod p$. Note that to avoid overflow in computing $s^d \ mod \ p$, you need to perform s^d in iterative way. For example, you can compute $s^2 \ mod \ p$ first and then reapply over and over until you get $s^d \ mod \ p$.

EX: $s^5 \ mod \ p = \{[(s^2 \ mod \ p)(s^2 \ mod \ p)] \ mod \ p\} \ (s \ mod \ p) \ mod \ p.$

Hint: In matlab, you may use fixed-point arithmetic toolbox, which would allow you to have positive integers as large as 63 bits. There is a "fi" built in matlab function which would represent integers in fixed-point arithmetic. You need to write your own modulo-m operation as the existing matlab mod function would not work with fixed-point representation.