The University of Sydney
Math3925 Public Key Cryptography

Semester 2 **Exercises and Solutions for Week 4** 2004

In order to implement an index calculus algorithm, we need a smoothness algorithm:

```
function IsSmooth(m,prms)
   // Returns true if and only if m factors over the prime
   // sequence prms, and if so, returns the exponent vector.
   error if m eq 0, "Argument 1 must be nonzero.";
   v := Vector([ 0 : i in [1..#prms] ]);
   for k in [1..#prms] do
       p := prms[k];
       if p eq -1 then
           if m lt 0 then
               v[k] +:= 1; m *:= -1;
           end if;
       else
           while m mod p eq 0 do
               v[k] +:= 1; m div:= p;
           end while;
       end if;
   end for;
   if m ne 1 then return false, _; end if;
   return true, v;
end function;
```

A smoothness base of t elements can be generated with a simple function:

```
function SmoothnessBase(t)
    prms := [ -1 ];
    p := 2;
    for i in [2..t] do
        Append(~prms,p);
        p := NextPrime(p);
    end for;
    return prms;
end function;
```

With these two functions, we can search for relations in the multiplication group $\mathbb{Z}/n\mathbb{Z}^*$. A simple index calculus algorithm is realised in the following lines of code:

```
function ModularRelations(n,prms,b,t)
    Z := Integers();
    R := ResidueClassRing(n);
    rels := [ RSpace(Z,#prms) | ];
    for k in [1..t] do
        u := Vector([ Random([0..b]) : i in [1..#prms] ]);
        m := Z!&*[ R!prms[i]^u[i] : i in [1..#prms] ];
        bool, v := IsSmooth(m,prms);
        if bool then
            Append(~rels,u-v);
        end if;
    end for;
    return rels;
end function;
```

1. **a.** Use the above functions to determine a set of prime generators and the complete sets of relations among them in $\mathbb{Z}/n\mathbb{Z}^*$ for $n = 2^{29} - 1$.

b. Use the relations to realise a factorization of n.

c. How does this method compare to a Pollard ρ factorization?

Solution

a. The function `SmoothnessBase(40)` sets up a factor base of size 40 (including -1). To eliminate numbers having a small prime factor, you can first do a GCD with each element of the factor base. `ModularRelations` is called to generate relations. The parameter t determines how many trials are carried out. Between 0 and t relations will be returned, and results from multiple trials can be concatenated. Putting the relations in echelon form reduces the relations to upper triangular form such that the $m \times m$

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -15 & 3 & 1 & 7 & 6 & 1 \\ 1 & 3 & 6 & 9 & 2 & -2 & 10 & -11 \\ 1 & -7 & -8 & -11 & 2 & -2 & 14 & -5 \\ 0 & 3 & -14 & -2 & -12 & -12 & 0 & -4 \\ 1 & 17 & 0 & -6 & 9 & -7 & 11 & 3 \\ 0 & 9 & -12 & -8 & 16 & 16 & -20 & -12 \\ 1 & 12 & 5 & -16 & -10 & 20 & 3 & -21 \end{bmatrix}$$

b. In order to factor n, one needs to iterate this relation collecting phase, then to solve a complete set of relations to determine the 2-torsion subgroup, then use this to factor n.

c. The running time for numbers of this size is much longer for this simple index calculus method than for a Pollard ρ. One expects a cross-over point, where the runtime coincides, to occur for integers of much larger size. Eventually, however, an optimal index calculus algorithm will outperform Pollard ρ.

2. **a.** Similarly find a set of generators and relations for the group $\mathbb{Z}/p\mathbb{Z}^*$ for the prime $p = 2^{31} - 1$.

b. Solve the discrete logarithm $\log_3(5)$ in this group using these relations.

Solution

a. For the set of generators $\{-1, 2, 3, 5, 7, 11, 13, 17\}$ we find a generator matrix

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & -2 & 0 & 1 & 14 & -1 & -7 \\ 0 & 8 & 1 & 11 & -11 & -1 & 2 & 0 \\ 1 & 8 & 4 & -9 & -5 & 5 & 12 & 6 \\ 1 & -2 & 0 & -8 & -15 & -2 & -2 & -10 \\ 0 & 15 & -15 & -6 & -8 & -7 & 6 & 0 \\ 1 & 6 & 14 & -13 & 4 & 6 & -10 & -10 \\ 0 & 5 & 6 & -6 & 5 & -15 & 11 & -18 \end{bmatrix}$$

b. If we permute the columns to move the columns corresponding to 3 (the third) and 5 (the fourth) to the last and next-to-last, respectively, and put the matrix in Echelon form, we find a lower right-hand submatrix:

$$\begin{bmatrix} 3 & 263334115 \\ 0 & 715827882 \end{bmatrix}$$

This implies that $5^3 3^{263334115} = 3^{715827882} = 1$, and thus $5^3 = 3^{715827882-263334115} = 3^{452493767}$. However, note that 452493767 is not divisible by 3 (the exponent of 5 in this relation) but $3|p-1$. So 5 is not in the subgroup generated by 3, and the discrete logarithm $\log_3(5)$ does not exist! This relation is as close as we can get – 5^3 is the first power of 5 in $\langle 3 \rangle \subset \mathbb{Z}/p\mathbb{Z}^*$.