S111503
Ran Wang
14/09/2015

01415 Computational Tools for Discrete Mathematics

Day2 Homework

Ran Wang Day 2 Home Work

S111503
Ran Wang
14/09/2015

Question 1: Construct a field of 8 elements.
This question can be considered as a finite field:

$$F2^3 = F2[x] / f(x)$$

where $f(x) = x^3 + x^2 + 1$ is an irreducible polynomial of degree 3 over F2, the order of $F2^3$ is namely{0,1,2}. The elements are $bx^2 + cx + d$ for all b,c,d ϵ {0,1}, thus we can construct its elements $F2^3 = \{0, 1, x, x+1, x^2, x^2+1, x^2+x, x^2+x+1\}$.

Question 2: Left to right binary exponentiation
Left-to-right binary exponentiation is also called Most Significant Binary exponentiation, which requires the binary format of a 10-based integer. $|F2^8| = 256$, with irreducible polynomial
$f := x^8 + x^4 + x^3 + x + 1 :$
I choose randomly one of its elements being $g := x^7 + x^4 + x^3 + 1$, and test in Maple using both in built function and my own algorithm, and I get the same inversion $(x^4 + x^2)$ (see appendix A). since the binary format of the exponentiation 256-2 is $[0, 1, 1, 1, 1, 1, 1, 1]$, so we have 8 squaring and 7 multiplications. Whilst right-to-left needs to factorize integer n by n/2 each time until it goes down to 0, but binary format don't need to worry this.

Question 3: Diffie-Hellman key exchange in F^*_{107}
The given group F^*_{107} has order 106 and its elements are{1,2,…,105,106}. The order of any element a in this group should divide 106 being {1,2,53,106} with their corresponding number of elements {1,1,52,52} (calculated using Euler's Phi function).

Diffies-Hellman key exchange is a method concerns securely exchanging keys over a public channel. There are two parties in this construction, both prior knowing the public information g and p with regards to g ϵ F^*_p , p =107, let's assume g is 35. a and b are randomly chosen by themselves.

1. Alice chooses a secret key a = 57 and sends Bob $A = g^a \bmod p$:
   $A = 35^{57} \bmod 107 = 57$
2. Bob chooses a secret key b = 73 and sends Alice $B = g^b \bmod p$:
   $B = 35^{73} \bmod 107 = 27$
3. Alice compute $k = B^a \bmod p$
   $K = B^a = (g^b)^a = 27^{57} \bmod 107 = 79$
4. Bob compute $K' = A^b$
   $k' = A^b = (g^a)^b = 57^{73} \bmod 107 = 79$
5. k = k' = 79
   When raise the bases to the power of their private keys, Bob and Alice share the same key.

Appendix

A.

$p := 2 : n := 8 :$
$f := x^8 + x^4 + x^3 + x + 1 :$
$F := GF(p, n, f)$

$$\mathbb{F}_{2^8} \tag{1.1}$$

$g := F{:-}ConvertIn\left(x^7 + x^4 + x^3 + 1\right)$

$$\left(x^7 + x^4 + x^3 + 1\right) \bmod 2 \tag{1.2}$$

$gm1 := F{:-}inverse(g)$

$$\left(x^4 + x^2\right) \bmod 2 \tag{1.3}$$

$pow := 2^8$

$$256 \tag{1.4}$$

```
l2rBinExp := proc(b :: zppoly, exp :: integer) :: zppoly;
    local i, nsq, nmult, bb, rb, m, A;
    m := exp;
    bb := convert(m, base, 2);
    with(ListTools) :
    rb := Reverse(bb) :
    A := F :-ConvertIn(1);
    nsq := 0;
    nmult := 0;

   for i in rb do
       A := F:-`*`(A, A);
       nsq := nsq + 1;
       if  i = 1 then
         A := F:-`*`(A, b);
         nmult := nmult + 1;
        end if
   end do;
   A;
 end proc;
```

```
proc(b::zppoly, exp::integer)::zppoly;                                    (2.1)
    local i, nsq, nmult, bb, rb, m, A;
    m := exp;
    bb := convert(m, base, 2);
    with(ListTools);
    rb := ListTools:-Reverse(bb);
    A := F:-ConvertIn(1);
    nsq := 0;
```

```
    nmult := 0;
    for i in rb do
        A := A * A; nsq := nsq + 1; if i = 1 then A := A * b; nmult := nmult + 1 end if
    end do;
    A
end proc
```

$gm1l2r := l2rBinExp(g, pow - 2)$

$$\left(x^4 + x^2\right) \bmod 2 \qquad \textbf{(3.1)}$$