Mathematics 3159A Introduction to Cryptography

Group Assignment 5 Due Date: December 8th, 2020

Report

Group members.

- Manager: Ao Xu axu33@uwo.ca
- Programmer: Dat Huu Vo dvo9@uwo.ca
- Reporter: Terry Zeng, tzeng23@uwo.ca
- Scribe: Jay Jigneshbhai Shah jshah48@uwo.ca
- Secretary: Morteza Al Rabya, malraby@uwo.ca

List all group members with their roles and email addresses.

Group meetings. The group met 2 times:

- Friday, Dec 4 8:00PM - 8:52PM
  Attendance: Ao (8:00 - 8:52), Dat (8:00 - 8:52), Terry (8:00 - 8:52), Jay (8:05 - 8:52), Morteza (8:05 - 8:52)
- Sunday, Dec 6 5:00PM - 7:05PM
  Attendance: Ao (5:00 - 7:05), Dat (5:15 - 7:05), Terry (5:05 - 7:05), Jay (5:15 - 7:05), Morteza (5:00 - 7:05)

Timeline. The meetings were scheduled in a Discord channel, and the notes were shared within the channel as well. In Meeting 1, we discussed Part 1 and we were able to come up with an idea for the solution using Remark 6.20 in the textbook. Our programmer wrote the program based on Part 1. In Meeting 2 we finalized our solution for Part 1, and fixed an issue with Part 2. The solutions were written in a shared LaTeX document.

Problem

Given a quintuple (p, A, B, P, Q) we know that p is a large prime, $A, B \in \mathbb{F}_p$ such that $4A_3 + 27B^2 \neq 0$, and $P, Q \in E(\mathbb{F}_p)$ where E is a curve given by $y^2 = x^3 + Ax + B$. Choose a secret element $s_0 \in \mathbb{F}_p$ and compute

$$r = x(s_0 P)$$
$$s_1 = x(rP)$$
$$t_1 = x(rQ)$$

If we are told that for some $e \in \mathbb{N}, P = eQ$, then show that given t_1, both s_1 and t_2 can be found

Solution

We know $t_1 = x(rQ)$ and $P = eQ$ and we have the A, B that satisfies the equation

$$Y^2 = X^3 + AX + B$$

We can solve for $y(rQ)$ by the following

$$y(rQ) \equiv (X^3 + AX + B)^{(k+1)} \mod p$$

given p satisfies $p \equiv 3 \mod 4$.
Now that we have $t_1 = x(rQ)$ and $y(rQ)$, we can find erQ using the double and add algorithm. We now have,

$$erQ \equiv reQ \equiv rP \mod p$$

Since $s_1 = x(rP)$ we just need the x coordinate for the above rP. Note that regardless of which solution we get from

$$y(rQ) \equiv (X^3 + AX + B)^{(k+1)} \mod p$$

for $y(rQ)$, it does not affect $t_1 = x(rQ)$ and thus does not affect the x-coordinate of rP

Now, to find t_2, we first find $r' = x(s_1 P)$ and $s_2 = x(r'P)$, so we also know $x(r'P)$ and $y(r'P)$. Finally, $t_2 = x(r'Q)$

Program

```
 ### HELPER FUNCTIONS ###
def dblP(pX,pY,p):
    if(pY == 0):
        return 'inf'
    # lambda
    gL = (3 * pow(pX,2) + A) * pow(2 * pY, p-2, p)
    # Nue
    gN = pY - (gL * pX)
    # P coordinates
    x3 = pow(gL,2) - (2 * pX)
    y3 = (-gL * x3) - gN
    return (x3 % p,y3 % p)

def dblAdd(pX,pY,qX,qY,p):
    if pX == qX and pY == (-qY % p):
        return 'inf'
    # lambda
    gL = (qY - pY) * pow(qX-pX,p-2,p)
    # Nue
    gN = pY - gL * pX
    # P coordinates
    x3 = pow(gL,2) - pX - qX
    y3 = (-gL * x3) - gN
    return (x3 % p,y3 % p)

### Double and Add Algorithm ###
def nPAlgo(pX,pY,n,p):
    # convert e to binary and flip
    nBin = [int(d) for d in f'{n:b}']
    # create list of Ps and put rQ in
    Ps = (pX,pY)
    # add all Ps for each binary digit in n
    for i in range(1,len(nBin)):
        if nBin[i] == 1:
            dbl = dblP(Ps[0],Ps[1],p)
            Ps = dblAdd(pX,pY,dbl[0],dbl[1],p)
        else:
            Ps = dblP(Ps[0],Ps[1],p)
        if Ps == 'inf':
            return (-1,-1)
    return Ps
```

```python
### SOLVE FUNCTION ###
def solve(p, A, B, Px, Py, Qx, Qy, e, t1):
    if (4*A**3 + 27*B**2) % p == 0:
        return "invalid A,B"
    # calculate k
    k = (p - 3) // 4
    # calculate rQy^2
    rQysq = pow(t1,3) + A * t1 + B % p
    # find rQy using y = q^(k+1) mod p for p = 4k+3
    rQy = pow(rQysq, k + 1,p)
    # use double and add algorithm to find erQ
    # recall: t1 = rQx
    erQ = nPAlgo(t1,rQy,e,p)
    # s1 = x(erQ)
    s1 = erQ[0]
    # r = x(s1P)
    r = nPAlgo(Px,Py,s1,p)[0]
    # t2 = x(rQ)
    t2 = nPAlgo(Qx,Qy,r,p)[0]
    return t2
```

Output

» solve(32416188691,103245,992349,8909974598,12810966706,
17766069436,28295926988,16789908933,20309635644)
27353310855
» solve(32416189987,1004,9998,21776342460,26884854746,
22858809622,16175884785,29713279001,18366438538)
10645293074
» solve(32416188127,19,33,16475848216,5118271045,
19262187694,2854205065,13332545241,9721955507)
23287589974