Final Project Source Code

Serdar HALİLOĞLU

20120602019

```
import hashlib
## Choose a prime number p, which is called the prime divisor.
p = 1;
while (p < 2^15): p = random_prime(2^16)

##Choose another primer number q. q is called the prime modulus.
q=1;
while (not is_prime(q)): q = ( randint(1,2^46)*2*p)+1
q;p

h = randint(2,q-2)
F=GF(q);F
g=F(h)^((q-1)/p)

## Choose an integer, such that 0 < x < p.
x = randint(0,p)

## Compute y as g**x mod q.
y = (g**x)%q

h;g;x;y

## Package the public key as {p,q,g,y}.
## Package the private key as {p,q,g,x}.

pubkey = (p,q,g,y)
prikey = (p,q,g,x)

pubkey;prikey

## PlainText - ConvertAscii -

plainText=open('/home/serdar/Desktop/CE340-\
   FinalProject_serdarhaliloglu/plainText.txt','r')
PT = plainText.read()
```

```
h=hashlib.sha1(PT)
a = h.hexdigest()
h;a

def convert_to_ascii(hash):
    asci='1'
    for x in range(0,len(PT)):
        if ord(hash[x])<100:
            asci += '0' + str(ord(hash[x]))
        else:
            asci += str(ord(hash[x]))
    return asci

M= convert_to_ascii(a)
M = int(M)
M

k = randint(0,p)
r = mod((g**k)%q,p)
s=mod(((M+x*r)/k),p)
k;r;s

signText=open('/home/serdar/Desktop/CE340-\
   FinalProject_serdarhaliloglu/signText.txt','w')
signText.write(str(r))
signText.write(str(s))
signText.write(str(PT))
signText.close()
signText

## Package the digital signature as {r,s}.

digital_signature = (r,s)
digital_signature

## Verify
Verify=open('/home/serdar/Desktop/CE340-FinalProject_serdarhaliloglu\
   /signText.txt','r')
Verify.read()
h = hashlib.sha1(PT)
b = h.hexdigest()
b

N = convert_to_ascii(b)
N = int(N)
N
```

```
## w is called the modular multiplicative inverse of s modulo p.
w=(s^(-1))%p
w

u1 = mod((N*w),p)
u2 = mod((r*w),p)
u1;u2

v=mod((((g**u1)*(y**u2))%q),p)
v

v==r
```