Assignment 5

Maxime CHAMBREUIL
McGill ID: 260067572
maxime.chambreuil@mail.mcgill.ca

Contents

1 Exercises from Stinson's book **1**
1.1 Exercise 5.11 p 220 . . . 1
1.1.1 Inverse operations . . . 1
1.1.2 Computation of a . . . 2
1.1.3 In Z_n . . . 2
1.1.4 Maple^{TM} . . . 3
1.2 Exercise 4.5 p 151 . . . 3
1.2.1 Polynom of degree 2 . . . 3
1.2.2 Polynom of degree d . . . 4
1.3 Exercise 7.6 p 312 . . . 4
1.3.1 Verification . . . 4
1.3.2 Computational advantage . . . 4
1.3.3 Security comparison . . . 4

2 Blum-Goldwasser **5**
2.1 Decryption / Verification . . . 5
2.2 Cyphertext attack . . . 5

3 Jacobi symbol and least significant bit **5**
3.1 Conditions on p and q . . . 5
3.2 Entropy . . . 5
3.3 Maple^{TM} . . . 5

4 Goldwasser-Micali **5**

1 Exercises from Stinson's book

1.1 Exercise 5.11 p 220

1.1.1 Inverse operations

To prove that encryption and decryption are still inverse operations with:

$$\lambda(n) = \frac{(p-1)(q-1)}{gcd(p-1,q-1)} = \frac{\phi(n)}{gcd(p-1,q-1)}$$

we will compute $\left(x^b\right)^a$:

$$\left(x^b\right)^a \equiv x^{t\lambda(n)+1} \text{ mod n } \equiv x^{t\lambda(n)} \times x \text{ mod n}$$

$$\left(x^b\right)^a \equiv \left(x^{\lambda(n)}\right)^t \times x \text{ mod n}$$

$$\left(x^b\right)^a \equiv (1)^t \times x \text{ mod n} \equiv x \text{ mod n}$$

So we have to prove that $x^{\lambda(n)} = 1$. We know that p and q are primes, so thanks to Fermat, we have:

$$x^{p-1} \equiv 1 \text{ mod p}$$

$$x^{q-1} \equiv 1 \text{ mod q}$$

By definition of $\phi(n)$, gcd (p - 1 , q - 1) appears twice in his factorisation : in the one of p-1 and the one of q-1. So if we consider the factorisation of $\lambda(n)$, we are going to find the complete one of p and the one of q-1 without the gcd. So we can deduce that:

$$x^{\lambda(n)} \equiv x^{(p-1)\times\frac{q-1}{gcd(p-1,q-1)}} \equiv 1 \text{ mod p}$$

With the same reasoning, we deduce:

$$x^{\lambda(n)} \equiv 1 \text{ mod q}$$

p and q are relatively primes so we can use the Chinese Reminder Theorem to set:

$$\exists! x, x^{\lambda(n)} \equiv 1 \text{ mod pq} \equiv 1 \text{ mod n}$$

So we have proved that $x^{\lambda(n)} \equiv 1$ mod n and that enables us to finish the original demonstration.

To conclude, we have found the original message so encryption and decryption are still inverse operations.

1.1.2 Computation of a

Given p = 37 and q = 79, we can find $\phi(pq) = 2808$ and with b = 7, we have:

$$7 \times a \equiv 1 \text{ mod } 2808 \Rightarrow 7a = 2808 \times t + 1$$

With t = 6, we obtain a = 2407 using the original RSA.

Using the modified cryptosystem, we have $\lambda(n) = 468$, so:

$$7 \times a = 468 \times t + 1$$

With t = 1, we obtain a = 67.

1.1.3 In Z$_n$

Let me remind you that $Z_n^* = \{x/gcd(x,n) = 1\}$ and $Z_n = Z_n^* \cup \{x/x = kp\}$

So we need to prove that the encryption/decryption is still valid for multiples of p (it would be the same for q):

$$\begin{aligned} d(e(kp)) &= d\left[(kp)^b \text{ mod n}\right] \\ &= \left[(kp)^b \text{ mod n}\right]^a \text{ mod n} \\ &= (kp)^{ab} \text{ mod n} \\ &= kp \text{ mod n} \end{aligned}$$

We can obtain the last line because of the first question. Therefore, the encyption/decryption is still valid in Z$_n$.

1.1.4 Maple^{TM}

This is my first algorithm when I read the questions, but it doesn't return any p and q :

```
find := proc():
     roll := rand(10^99..10^100):
     p := roll():
     q := roll():
     while (gcd(p-1,q-1)< 10^74 ) do:
           p := roll():
           q := roll():
     end do:
     return p,q:
end proc:
```

So I took the problem upside down :

```
find := proc():
     roll := rand(10^74..10^75):
     gcd := roll():
     p := 3*10^25*gcd:
     q := 4*10^25*gcd:
     while (not (isprime(p+1)) or not (isprime(q+1)) ) do:
          gcd := roll():
          p := 3*10^25*gcd:
          q := 4*10^25*gcd:
     end do:
     return p+1,q+1:
end proc:
```

and I finally find:
p = 588660476800131775105978461149584613361071240656615403980188348579939200766
0000000000000000000000001
q = 784880635733509033474637948199446151148094987542153871973584464773252267688
0000000000000000000000001

1.2 Exercise 4.5 p 151

1.2.1 Polynom of degree 2

To solve the SecondPreimage problem, assume we have:

$$\begin{cases} x \Rightarrow y = x^2 + ax + b \text{ mod } 2^m \\ x' \Rightarrow y = x'^2 + ax' + b \text{ mod } 2^m \end{cases}$$

So

$$\begin{aligned} y = x^2 + ax + b &= x'^2 + ax' + b \\ \Rightarrow x^2 - x'^2 + ax - ax' &= 0 \\ \Rightarrow (x - x')(x + x' + a) &= 0 \\ x = x' \text{ or } x' &= -x - a \end{aligned}$$

We can find a second preimage by computing $-x - a$, which is really easy.

1.2.2 Polynom of degree d

I first try to solve the SecondPreimage problem in the same way as before, but I finally obtained:

$$\sum_{j=0}^{i} x^i x'^{i-j} \equiv 0 \text{ mod } 2^m$$

which doesn't help us to find a second preimage to x.

I was told that the problem is easy to solve by using the fact that $n > m$, but I was not able to figure out the problem with this hint.

1.3 Exercise 7.6 p 312

1.3.1 Verification

To verify this new signature scheme, we have to compute $\beta^\delta \gamma^\gamma$:

$$\begin{aligned} \beta^\delta \gamma^\gamma &= (\alpha^a)^{(x-k\gamma)a^{-1}} \gamma^\gamma \text{ mod p} \\ &= (\alpha)^{(x-k\gamma)} \gamma^\gamma \text{ mod p} \\ &= (\alpha)^{(x-k\gamma)} \left(\alpha^k\right)^\gamma \text{ mod p} \\ &= \alpha^x \text{ mod p} \end{aligned}$$

1.3.2 Computational advantage

In the original El Gamal Signature Scheme, we have to compute the multiplicative inverse of k ofr each message.

In this modified scheme, we can invert a once for all message. So it is computaionally faster.

1.3.3 Security comparison

We have seen in class that behind the original El Gamal Signature Scheme, there is always the Discrete Logarithm Problem (DLP). Let's now see what there is behind the modified scheme :

Given x, set a γ and then try to find δ with:

$$\beta^\delta = \alpha^x \gamma^{-\gamma} \text{ mod p}$$

which is equivalent to the DLP.

Given x, set a δ and then try to find γ with:

$$\gamma^\gamma = \alpha^x \beta^{-\delta} \text{ mod p}$$

which seems to be easier than the DLP.

Given x, try to simultaneously find δ and γ with:

$$\beta^\delta \gamma^\gamma = \alpha^x \text{ mod p}$$

which is equivalent to the DLP.

So we can conclude that the modified scheme is less secure than the original El Gamal.

2 Blum-Goldwasser

2.1 Decryption / Verification

Concerning the decryption, the modification does not affect encryption so the decryption is the same as in the original cryptosystem.

The verification consists in reproducing the same operations described in the exercise and compare w obtained with the one transmitted.

2.2 Cyphertext attack

In the cyphertext attack, Oscar has an temporary access to the decryption machinery and he can compute dec (y) = x. If this modified cryptosystem fails under this attack, it means that Oscar can find the private key p and q. But he can also break the system with the least significant bit of x^2, but I was not able to deal with that information.

3 Jacobi symbol and least significant bit

3.1 Conditions on p and q

We know that:

$$\left(\frac{n-1}{n}\right) = \left(\frac{n-1}{p}\right)\left(\frac{n-1}{q}\right) = +1$$

So

$$(n-1) \in QR_p \Rightarrow (n-1) \in QR_q$$
$$(n-1) \in QNR_p \Rightarrow (n-1) \in QNR_q$$

3.2 Entropy

To clarify the formulae, we have to prove that the proportion of odd and even number doesn't change because of the Jacobi Symbol of M^e.

We know that if $\left(\frac{n-1}{n}\right) = +1$, the number of odd and even number are equally distributed in the set of x such that $\left(\frac{x}{n}\right) = +1$.

So let's consider a bijection, which associates an odd to an even number and an even to an odd. Therefore we keep the same proportion of even and odd number. If both number in each pair has the same Jacobi Symbol, we can conclude that the Jacobi Symbol of M^e has no incident on the proportion.

3.3 Maple^{TM}

Finding a counter example should be enough to prove that it is false with the Jacobi Symbol -1.

4 Goldwasser-Micali