

# DTU



TECHNICAL UNIVERSITY OF DENMARK

01410 CRYPTOLOGY 1

---

## Homework 2

---

Authors:

Andreas Hallberg KJELDSSEN  
s092638@student.dtu.dk

Morten Chabert ESKESEN  
s133304@student.dtu.dk

April 7, 2014

## Exercise 2.1

### 2.1.1

We have to show that  $m^{e\tilde{d}} \equiv m \pmod n$  for all  $m \in \mathbb{Z}_n$ .  
The keys  $e$  and  $\tilde{d}$  are chosen such that.

$$e\tilde{d} \equiv 1 \pmod{\frac{(p-1)(q-1)}{\gcd(p-1, q-1)}}$$

This means that for some positive integer  $k$

$$e\tilde{d} = 1 + k \frac{(p-1)(q-1)}{\gcd(p-1, q-1)}$$

We can rewrite this expression and get

$$m^{e\tilde{d}} \pmod n = m^{1+k \frac{(p-1)(q-1)}{\gcd(p-1, q-1)}} \pmod n$$

for some integer  $k$ .

If two integers  $x$  and  $y$  are congruent modulo  $n$  then they are also congruent modulo  $p$  and modulo  $q$  because both  $p$  and  $q$  divide  $n$ . The Chinese Remainder Theorem tells us that the reverse implication is also true. This means that if  $x$  and  $y$  are congruent modulo  $p$  and congruent modulo  $q$ , then they are also congruent modulo  $n$ .

We want to show that  $m^{e\tilde{d}} \equiv m \pmod n$  so if it will be sufficient to show that:

$$m^{e\tilde{d}} \equiv m \pmod p \text{ and } m^{e\tilde{d}} \equiv m \pmod q$$

First we will show that  $m^{e\tilde{d}} \equiv m \pmod p$ . We have therefore have two cases to consider:

1.  $p$  divides  $m$
2.  $p$  does not divide  $m$ .

Case 1: If  $p$  divides  $m$ , then  $m \equiv 0 \pmod p$ , but also  $m^{e\tilde{d}} \equiv 0 \pmod p$ , therefore  $m^{e\tilde{d}} \equiv m \pmod p$ .

Case 2: If  $p$  does not divide  $m$  then  $m \in \mathbb{Z}_p^*$ . By Fermat's Little Theorem we have  $m^{p-1} \equiv 1 \pmod p$ . Since  $e\tilde{d} \equiv 1 \pmod{\psi(n)}$ , we have that  $\psi(n)$  divides  $e\tilde{d} - 1$ . This equivalent with some integer  $k$ :  $k\psi(n) = e\tilde{d} - 1$ , so  $e\tilde{d} = k\psi(n) + 1$  for some integer  $k$ . We therefore have:

$$\begin{aligned} m^{e\tilde{d}} &= m^{k\psi(n)+1} = m * m^{k \frac{(p-1)(q-1)}{\gcd(p-1, q-1)}} \\ m^{e\tilde{d}} &= m * (m^{p-1})^{k \frac{(q-1)}{\gcd(p-1, q-1)}} \\ m^{e\tilde{d}} &\equiv m * 1^{k \frac{(q-1)}{\gcd(p-1, q-1)}} \pmod p \\ m^{e\tilde{d}} &\equiv m \pmod p. \end{aligned}$$

We can do similar calculations to show that  $m^{e\tilde{d}} \equiv m \pmod q$  by replacing  $p$  by  $q$ .  
Therefore we have now shown for all  $m \in \mathbb{Z}_n$  that

$$m^{e\tilde{d}} \equiv m \pmod p \text{ and } m^{e\tilde{d}} \equiv m \pmod q$$

Hence we can conclude that  $m^{e\tilde{d}} \equiv m \pmod n$  for all  $m \in \mathbb{Z}_n$ .

### 2.1.2

Let  $p = 881$ ,  $q = 461$ , and let  $n = pq = 405141$ . We have to show that  $e = 3$  is an allowed encryption exponent for an RSA encryption system with modulus  $n$ . By the definition of RSA  $e$  must be chosen such that  $e$  and  $\phi(n)$  are co-prime. Formally this means that  $e \in \mathbb{Z}_{\phi(n)}^*$ , where  $\phi(n) = (p-1)(q-1)$ .

$$\gcd(3, (881-1)(461-1)) = 1$$

This means that  $e$  and  $\phi(n)$  are co-prime and therefore  $e = 3$  is an allowed encryption exponent.

### 2.1.3

We have to find  $d_1$  such that  $ed_1 \equiv 1 \pmod{\phi(n)}$ .

```
> p := 881; q := 461; e := 3;  
d1 := mod(e^-1, (p-1)*(q-1))
```

Using the maple code above we find that  $d_1 = 269867$

### 2.1.4

We have to find  $d_2$  such that  $ed_2 \equiv 1 \pmod{\psi(n)}$

```
> p := 881; q := 461; e := 3;  
d2 := mod(e^-1, (p-1)*(q-1)/gcd(p-1, q-1))
```

Using the maple code above we find that  $d_2 = 6747$ .

### 2.1.5

Choosing  $\psi(n)$  instead of  $\phi(n)$  in the congruence for  $d$  means that the decryption becomes faster since

$$\text{lcm}(p-1, q-1) = \frac{(p-1)(q-1)}{\gcd(p-1, q-1)} \leq \frac{(p-1)(q-1)}{2}$$

Because  $p$  and  $q$  are odd primes  $\gcd(p-1, q-1) \geq 2$ .

## Exercise 2.2

### 2.2.a

We have implemented trial division in maple with the following code:

```
> TrialDivision := proc (n::integer)
local i;
if n ≤ 1 then false
elif n = 2 then true
elif type(n, 'even') then false
else for i from 3 by 2 while i*i ≤ n do
if irem(n, i) = 0 then return false end if
end do;
true end if
end proc:
> result := 0;
for n from 25 to 25000 do
if TrialDivision(n) then result := result+1 end if
end do;
result;
```

Using this code we find that the number of primes  $s$  between 25 and 25000 is 2753.

$$s = 2753$$

## 2.2.b

We have implemented the Miller-Rabin algorithm with  $k$  iterations in maple with the following code:

```
> MillerRabin := proc (n::integer, k::integer)
local x, r, roll, s, d, i, a;
s := n-1; d := 0;
while mod(s, 2) = 0 do
s := (1/2)*s; d := d+1
end do;
for i to k do
roll := rand(2 .. n-1);
a := roll(); x := mod(as, n);
if x = 1 or x = n-1 then next end if;
for r to d-1 do x := mod(x2, n);
if x = 1 then return false end if;
if x = n-1 then break end if
end do;
if x ≠ n-1 then return false end if
end do;
return true
end proc;
```

## 2.2.c

We use this maple code below and define  $k = 1, 2, 3, \dots$  to find the smallest number of iterations needed such that we get the correct answer  $s$ .

```
> result := 0;
for n from 25 to 25000 do
  if MillerRabin2(n, k) then result := result+1 end if
end do;
result;
```

This gives us the following table:

k	1	2	3	4
s	2792	2755	2754	2753

With  $k = 4$  iterations we get the correct answer for  $s$  which is 2753.