Exercises on Modulo Arithmetic and RSA

Víctor Alcázar Kosmas Palios Albert Ribes

April 2, 2017

Exercise 1

The idea is to use Euler's theorem (generalization of Fermat's Little Theorem).

$$a^{\phi(n)} = 1 mod n$$

where a is coprime to n.

This helps us compute powers modulo n. E.g, in the exercise $n = 10, \phi(n) = 4, a = 3$. It is true that $28 = 4 * 7$, therefore we can be sure that $3^{28} = 1(mod 10)$

The second case is different: $n = 15, \phi(n) = 8, a = 3$. Here we notice that lcd(3,15)=3, and therefore we cannot use Euler's Theorem.

However, we happily discover that $3^4 mod 15 = 3$. That means that we have actually found a repetition pattern in the chain $3^1, 3^2, 3^3, 3^4 \ldots$ and this means that $3^{3k+1} = 3$. But we know that $200 = 3*66+2$, therefore : $3^{200} = 3^{3*66+1} * 3 = 3^2 = 9$.

Exercise 2

We first compute $x = b^c mod \phi(p)$

Then we compute $a^x mod p$.

Why is this correct? Simply because for every p,a, as Euler tells us that$a^{\phi(p)} = 1 mod p$, we have the following $a^k = a^{k+\phi(p)*l} mod p$.

Exercise 3

Solution.

Exercise 4

SOLUTION.

Exercise 5

In the given cryptosystem, to decrypt a ciphertext we must use the private key d, in the following way $m = c^d mod p$. The problem here is that the private key can be easily computed. We know that the relation between e and d is $ed = 1 mod \phi(p)$. But now $\phi(p) = p - 1$, because p is prime!

So all we have to do is compute the multiplicative inverse of e modulo p-1. This can be easily done using the Extented Euclidean Algorithm, which takes $O(log(e)2)$ time. Then it is only a matter of exponentiation of c to the power of d modulo p. This in itself takes log(d) steps if done with repeated squaring.

In total we have $O(log(e)2 + log(d))$ time.