Master 1 – Crypto - TP : RSA coding

1 Preliminaries

1.1 Using GMP

GMP (GNU Multiprecision Library)[1] is a `C` library for multi-precision integers, and rationals.

Mulitiprecision integers are represented by arrays of `unsigned long int`. The corresponding `C` type is `mpz_t`. Beware that the declaration and initialization of an array are always separated.

```
mpz_t x,y,z;
mpz_init (x);      //x is initialized
mpz_init_ui(y,10); //y is initialized and receives the value 10
mpz_init_ui(z,30); //y est initialisé and receives the value 10
mpz_add (x,y,z);
```

You can refer to the manual of the library, available on-line[2].

2 Conversions

Download the file `https://www-fourier.ujf-grenoble.fr/~viva/teaching/RSA_init.c` and explain the two following functions: `string_to_int` and `int_to_string`

3 Generating large prime numbers

A common way of generating large prime numbers is to pick random integers in a target size range and test if they are primes. We propose two approaches to primality testing.

1. Program your own version of the sieve of Eratosthenes. What is the memory complexity of this algorithm ? Give a sensible upper bound on the number of prime numbers that can be found that way.

2. Program a primality testing algorithm based on trial division, and test it on numbers of increasing size. Give a sensible upper bound on the size of numbers that can tested for primality that way.

Since this naïve approach is unsuitable for numbers of cryptographic size, we need more efficient primality tests. We will implement the widespread Miller-Rabin test, which is probabilistic.

[1]http://gmplib.org/
[2]http://gmplib.org/manual/

3. Let $n \geq 3$ be an odd integer. Show that n is prime if and only if the only square roots of 1 modulo n (i.e. integers x such that $x^2 = 1 \mod n$) are -1 and 1.

4. Let s, m be the two integers such that $n - 1 = 2^s m$ and m is odd. Show that if n is prime, then $a^m = 1 \mod n$ or there exists $i \in \{0, \ldots, s-1\}$ such that $a^{2^i m} = -1 \mod n$. Hint : factorise the polynomial $X^{n-1} - 1$ in $\mathbb{Z}/n\mathbb{Z}[X]$.

5. Let a be an integer. We say that a is a Miller-Rabin witness for n if $a^m \neq 1 \mod n$ and $a^{2^i m} \neq -1 \mod n$ for all $i \in \{0, \ldots, s-1\}$.
   Write a program that takes as input two integers a and n and determines if a is a Miller-Rabin witness for n. What is its complexity ?

   We will admit that if n is not prime, then at least one half of the elements of $\mathbb{Z}/n\mathbb{Z}^\times$ are Miller-Rabin witnesses. In particular, if a is chosen uniformly randomly in $\{1, \ldots, n-1\}$ then we can detect that n is composite with a probability greater than $1/2$.

6. Implement the Miller-Rabin primality testing. The program will take as input an integer n and a threshold probability p. It will choose random integers and test if they are Miller-Rabin witnesses for n, repeatedly until either n is proven composite or is prime with probability greater than $1 - p$.
   For random number generations, use the functions `gmp_randstate_t`, `gmp_randinit_default` and `mpz_urandomm`.

4 RSA

1. Implement a function to generate a pair of RSA keys, using Miller-Rabin primality test.

2. Implement the functions `RSA-encrypt` and `RSA-decrypt` acting on a ASCII text in ECB mode.

3. Demonstrate your code on some examples of your choice.