Problem Set 3

GMP: General Use and RSA

Problem I

We are intersted here in finding prime numbers suitable for secured cryptographic operations. Most cryptographic operations require large prime numbers ranging from 1024 bits to 4096 bits. Finding such prime number is not a trivial operation. However, with the help of the GMP library and its associated predicate, life can be made easier.

1. With the help of the `mpz_probab_prime_p` predicate, write a program to test if a number is a probable prime.

2. Test your program with 199 and 583.

3. Write a program which computes the mean distribution of prime numbers between two integer bounds.

4. Write a program which finds a probable prime number given its size in bits. How many trials are needed to find a such prime ?

Problem II

We consider $n = 77$, $e = 13$, $d = 37$ and we note:

$$K_{\text{PUBLIC}} = (n, e)$$

$$K_{\text{PRIVATE}} = (n, d)$$

1. Encrypt, by hand (or with a calculator), the message $M = 26$, using the Rivest-Shamir[1]-Adleman algorithm (RSA), with key K_{PUBLIC}.

2. Decrypt the ciphered message you obtained in the previous question with key $K_{\text{PRI VATE}}$.

3. The numbers involved in the keypair are incredibly small. Recover K_{PRIVATE} f rom K_{PUBLIC}.

[1] Adi Shamir, an Israeli cryptographer, is one of the inventors of the RSA algorithm (and the S of RSA) and one of the inventors of differential cryptanalysis.

Adi Shamir (1952–)

Problem III

We are intersted here in the implementation of RSA algorithm with the help of the GMP library. The implementation will be done in two steps. The first step is the key generation, the second step is the message encryption/decryption itself. Each function designed in this exercise must be associated with a function tester.

1. With the help of the GMP library, write a function that generates a prime number of a given size in bits.

2. With the help of the previous question, write a function that generates a RSA key. It is recommended to store the key components into a structure.

3. Given a RSA key, write a function that encrypts a message. You are supposed to verify that the message is acceptable with respect to the generated key.

Problem IV

In this supplementary part, you are asked to design a program that integrates the previously designed functions. Such a program must be able to generate a key of a given size, if requested to do so, and encrypt or decrypt a message. The message might be represented as an octer string if desired.