Practical Work 2

The RSA Cryptosystem

Abstract

The aim of this practical work is to implement an RSA key generation, as well as the encryption/decryption and signature computation/signature verification functions. You must write your programs in C language and use the GMP library for computations on large integers.

Good work!

1 RSA Cryptosystem (standard mode)

1.1 Key generation

Write a program that takes as inputs two integers k and e, and generates a (standard mode) RSA key of size k bits with e as its public exponent.

The modulus n must be computed as the product of two primes p and q of same bit-size. Take care that n bit-size itself must be exactly k.

Remark 1 *Note that the inverse of e modulo $\phi(n)$ does exist only if e and $\phi(n)$ are coprime.*

The program will display the generated key on standard output in hexadecimal form on three lines like this:

```
e = 0x......
n = 0x......
d = 0x......
```

Remark 2 *The key can easily be written into a file by simply piping the standard output into this file.*

1.2 Encryption and Decryption

- Write an encryption function `encrypt_rsa`(c, m, n, e) which takes as inputs a message $m \in \mathbb{Z}_n$ the modulus n and the public exponent e, and computes the ciphertext $c = m^e \bmod n$.
- Write a decryption function `decrypt_rsa`(m, c, n, d) which decrypts the ciphertext c using the modulus and the private exponent d by computing $m = c^d \bmod n$.
- Write a program which allows to encrypt and/or decrypt a small text – which must be converted into an integer $m \in \mathbb{Z}_n$ beforehand – rather than an integer which has no particular meaning. The key elements are to be read from a key file created at Section 1.1.
- Play with your classmate by sending encrypted secret messages, and decrypting them.

1.3 Signature and Verification

Write a program which takes a filename and an RSA private key (n, d) as inputs, and computes the signature s of the file as:

$$s = h^d \bmod n$$

where h is the integer representation of the MD5 hash of the file.

2 RSA Cryptosystem (CRT mode)

Do the same as in Section 1 but this time the key is generated in CRT mode. Which functions or programs need to be modified?, which do not?