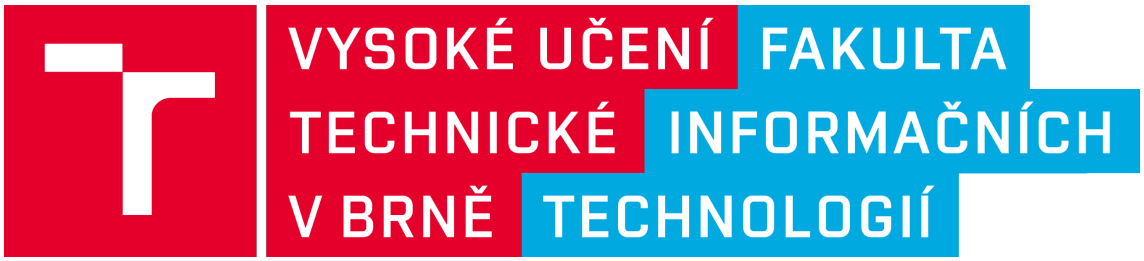

Cryptography
RSA Implementation and Cracking

3. května 2020

Jiří Matějka (xmatej52)

Obsah

1 Introduction

The goal of this document is to explain how the RSA algorithm was implemented in the second project and what algorithm was used to crack weak RSA keys.

RSA (Rivest–Shamir–Adleman) is one of the most commonly used algorithms for data encryption. It is an asymmetric algorithm, so there are two different types of keys – private key and public key. The algorithm is based on a problem called prime factorization.

2 RSA Implementation

In these section, we will explain:

- RSA key creation (Section 2.1),
- RSA encryption (Section 2.2),
- RSA decryption (Section 2.2).

2.1 RSA Key Creation

The first step to generate RSA keypair is to generate two random primes – p and q. In this works, program access file */dev/urandom* to generate a random number, then the first and the last bits of the number are set to 1 and finally the number the primality is tested using the Fermat primality test. This process is repeated until two primes are generated.

Then modulus $n = p * q$ is computed. This modulus is part of both private and public keys. We also compute $phi(n) = (q-1)/times(p-1)$

Then random number e is generated. Number e is generated again, until condition $e > 1 and gcd(e, phi(n)) = 1$ is fullfiled.

Once e is known, we compute the multiplicative inverse of e and $phi(n)$ and store it into number d. The pairs (e, n) and (d, n) can be used as private or public keys.

2.2 RSA Encryption and Decryption

The process of encrypting and decrypting data is the same. For data encryption (or decryption) algorithm called modular exponentiation.

3 RSA Cracking

To crack the RSA, we need to compute numbers p and q from modulus n. This can be done with prime factorization. Since we need to find only one divisor of modulus n, we used in the project Pollard's Rho algorithm for prime factorization. The concept of this algorithm is:

1. Two numbers x and y are said to be congruent modulo n ($x = ymodn$) if one of the following conditions is true:
   (a) Each of them has the same remainder if they are divided by n
   (b) Their difference is an integer multiple of n
2. The Greatest common divisor is the largest number that divides into each of the original numbers.
3. Birthday Paradox.
4. Floyd's cycle-finding algorithm.

And the implemented algorithm works as follows:

```
primeFactorPollard( n, p, q ) {
    if ( n == 1 ) {
        p = 1;
        q = 1;
        return;
    }

    if ( isPrime( n ) ) {
        p = n;
        q = 1;
        return;
    }

    if ( n % 2 == 0 ) {
        p = 2;
        q = n / 2;
        return;
    }

    while ( true ) {
        x = random();
        c = random();

        // Birthday paradox
        x = ( x % ( n - 2 ) ) + 2;
        c = ( c % ( n - 1 ) ) + 1;
        y = 1
        d = 1;
        while ( d == 1 && d != n ) {
            // Tortoise Move x(i+1) = f(x(i))
            x = powerModulus( x, 2, n )
            x = x + c + n
            x = x % n

            // Hare Move y(i+1) = f(f(y(i)))
            y = powerModulus( y, 2, n )
            y = y + c + n
            y = y % n
            y = powerModulus( y, 2, n )
            y = y + c + n
            y = y % n

            if ( x > y ) {
                d = gcd( x - y, n )
            }
```

```
46                  else {
47                      d = gcd( y - x, n )
48                  }
49
50                  if ( d != 1 && d != n ) {
51                      p = d;
52                      q = d / 2;
53                      return;
54                  }
55              }
56          }
57      }
```

After the p and q is found, the private key can be computed the same way as described in section 2.1. And using the private key, the message can decrypted.

Information needed to implement and describe Pollard's Rho algorithm for prime factorization was gained from [1].

4 Conclusion

The goal of this project was to implement an application that will be able to generate RSA keypair, encrypt messages, decrypt messages and crack RSA algorithm. The software is capable to generate keypairs, encrypt and decrypt messages and can crack the 96 bit RSA key under 10 seconds on intel i5-7300HQ processor using only one core.

Reference

[1] CHITRANAYAL. *Pollard's Rho Algorithm for Prime Factorization* [online]. [vid. 2020-05-03]. Dostupné z: < <https://www.geeksforgeeks.org/pollards-rho-algorithm-prime-factorization/>>.