KRY: Implementace a prolomení RSA

Katarína Grešová (xgreso00)

26. apríla 2020

1 Úvod

Táto správa sa venuje popisu zvolených algoritmov pre projekt do predmetu KRY - Implementace a prolomení RSA. Cieľom projektu bolo zoznámiť sa s algoritmom RSA a implementovať program, ktorý bude vedieť generovať parametry RSA, šifrovať, dešifrovať a prelomiť RSA pomocou faktorizácie slabého verejného modulu. Program bol implementovaný v jazyku C++ s využitím knižnice GMP pre prácu s veľkými číslami.

2 Generovanie parametrov RSA

Generovanie parametrov RSA je implementované v triede `KeyGenerator`. Vstupom je veľkosť verejného modulu v bitoch. Výstupom je potom súkromný a verejný kľúč. Samotné generovanie kľúčov je znázornené v algoritme 1.

Algorithm 1: Generovanie parametrov RSA

input : požadovaná veľkosť verejného modulu v bitoch - B
output: súkromný kľúč (d,n) a verejný kľúč (e,n)
e = b > 2048 ? 65537 : 3;
do
  generuj prvočíslo **p** s veľkosťou B // 2;
  generuj prvočíslo **q** s veľkosťou B - veľkosť(p);
  phi = (p - 1) * (q - 1);
while *gcd(e, phi) != 1*;
n = p*q;
d = inverse(e, phi);

3 Šifrovanie a dešifrovanie

Šifrovanie zadanej správy pomocou verejného kľúču je implementované v triede `Encryptor` a to podľa vzťahu [1]:

$$c = m^e \pmod{n} \tag{1}$$

Dešifrovanie správy pomocou súkromného kľúča je implementované v triede `Decryptor` a to podľa vzťahu [1]:

$$m = c^d \pmod{n} \tag{2}$$

4 Prelomenie RSA

Prelomenie RSA je implementované v triede `Breaker`. Základom prelomenia je faktorizácia verejného modulu. Pri faktorizácii je najskôr použité triviálne delenie až do hranice 1 000 000. Na urýchlenie je použité Eratostenovo sito a skúša sa delenie len prvočíslami.

V prípade neúspechu sa použije Pollardova rho metóda [2]. Táto metóda využíva narodeninový paradox. Namiesto generovania jedného náhodného čísla, sa generujú náhodné čísla dva, čím sa zvyšujú šance na správne uhádnutie. Ak generujeme náhodné číslo z intervalu $< 0, 1000 >$, potom pravdepodobnosť, že dostaneme 42 je 0.001. Ak však generujeme čísla dva i a j a chceme, aby $i - j = 42$, potom je pravdepodobnosť 0.002 [3]. Pollardova faktorizačná metóda bola zvolená kvôli jej nízkej priestorovej zložitosti a kvôli jej rýchlosti v porovnaní s Fermatovou metódov. Princíp je znázornený v algoritme 2.

Algorithm 2: Pollardova rho faktorizačná metóda

```
vygeneruj náhodné čísla x a c;
polož y rovné x;
polož f(x) = x^2 + c;
while deliteľ nie je získaný do
    x = f(x)  (mod n);
    y = f(f(y))  (mod n);
    vypočítaj GCD |x − y| a n;
    if GDC != 1 then
        if GDC == n then
            opakuj algoritmus s inými náhodnými číslami;
        else
            GDC je výsledkom;
        end
end
```

Výsledom je jeden faktor verejného modulu. Druhý faktor je jednotucho získaný ako $q = n/p$. Na dešifrovanie správy ďalej potrebujeme získať hodnotu súkromného exponentu, ktorú môžeme vypočítať ako $d = Inverse(e, phi)$, kde $phi = (p-1) * (q-1)$. Následné dešifrovanie správy môže byť vykonané podľa rovnice (2).

5 Záver

V rámci tohto projektu bol naštudovaný algoritmus RSA. Ďalej bol implementovaný program, ktorý je schopný generovať parametry RSA, šifrovať, dešifrovať a prelomiť RSA pomocou faktorizácie verejného modulu.

Literatúra

[1] James Nechvatal. Public-key cryptography. Technical report, NATIONAL COMPUTER SYSTEMS LAB GAITHERSBURG MD, 1991.

[2] John M Pollard. Theorems on factorization and primality testing. In *Mathematical Proceedings of the Cambridge Philosophical Society*. Cambridge University Press, 1974.

[3] A quick tutorial on pollard's rho algorithm. `https://www.cs.colorado.edu/~srirams/courses/csci2824-spr14/pollardsRho.html`. Accessed: 2020-04-26.