Vysoké učení technické v Brně
Fakulta informačních technologií
KRY projekt 2 – RSA
Pavel Bednář (xbedna73)

1 Algoritmus RSA

Jedná se o asymetrický šifrovací algoritmus, který byl objeven pány Rivest, Shamir, Adelman v roce 1978. Je založen na problému faktorizace velkých čísel. Soukromý klíč představuje dvojce $(n, d), n = p*q$, kde n je veřejný modulus, veřejný klíč představuje dvojce $(n, e), n = p*q$, kde p a q jsou prvočísla, n je veřejný modulus, d je soukromý exponent a e je veřejný exponent. Zároveň musí platit rovnost $d * e \mod (p-1)(q-1) = 1$. Šifrování veřejným klíčem probíhá dle vzorce $c = m^e \mod n$, opačná operace dešifrování soukromým klíčem je definována vztahem $m = c^d \mod n$. Takovéto nastavení slouží pro utajení [1].

2 Návrh a implementace

Implementační jazyk je C++ za využití knihovny GMP pro práci s velkými čísly. Šifrování a dešifrování je implementováno přímočaře dle výše uvedených vzorců. Následuje popis dvou hlavních částí, a to generování soukromého a veřejného klíče a faktorizace, nebo-li prolomení šifrovacího algoritmu.

Generování klíčů

Nejprve se vygenerují dvě náhodná prvočísla p a q. Generování náhodného prvočísla probíhá následovně. Použije se generátor náhodných čísel z knihovny GMP. Pro větší bezpečnost je inicializován C++ funkcí *random_device*, která vrací náhodnou hodnotu z operačního systému. V Unix systémech je to hodnota z */dev/urandom*. U takto vygenerovaného čísla ověřím jestli má nejvýznamnější bit hodnotu 1 a jestli se jedná o prvočíslo, což je realizováno metodou Miller-Rabin s 10 iteracemi, což zajišťuje pravděpodobnost správně určeného prvočísla přes 99 % [2]. Veřejný exponent zvolím náhodný, ale s maximální hodnotou $\phi(n) = (p-1)(q-1)$ a $NSD(e, \phi(n)) = 1$. Soukromý exponent spočítám jako $d = e^{-1} \mod \phi(n)$, kde e^{-1} naleznu pomocí Rozšířeného Euklidova algoritmu[1]. Veřejný exponent vypočtu jako $n = p * q$.

Faktorizace

Pro faktorizaci je využita Pollard Rho metoda[2]. Ta z definice může vrátit neúspěch, proto pro tyto případy je ještě implementována Fermatova metoda[3]. Z provedených experimentů ovšem plyne, že první zmíněná metoda je výrazně rychlejší. Ve všech případech je pro první milion dělitelů použita metoda triviálního dělení.

3 Závěr

Závěrem lze konstatovat, že jsem vytvořil funkční program pro práci s RSA, který byl řádně otestován.

[1] https://cs.wikipedia.org/wiki/Rozš%C3%ADřený_Eukleidův_algoritmus
[2] https://en.wikipedia.org/wiki/Pollard%27s_rho_algorithm
[3] https://en.wikipedia.org/wiki/Fermat%27s_factorization_method

Literatura

[1] Hanáček, P.: Studijní podklady k předmětu KRY – Asymetrické algoritmy. online, 2022.

[2] Rogalewicz, A.: Studijní podklady k předmětu SLOa – Randomized Computation. online, 2022.