Dokumenácia k 2. projektu z KRY

Implementácia a prelomenie RSA

Matej Marušák, Apríl 2018

1 Úvod

V tejto správe je popísaná implementácia šifry RSA a jej prelomenie. Jedná sa o aplikáciu v jazyku C za použitia knižnice GMP. V prvej kapitole je popísaný obecný princíp šifry RSA, druhá kapitola detailne popisuje implementáciu výslednej aplikácie.

2 Šifra RSA

Asymetrická šifra s verejným kľúčom RSA je založená na zložitosti rozkladu čísla na súčín prvočísel. Tento problém označujeme ako faktorizácia. Dnes neexistuje žiadna metóda, ktorá by dokázala faktorizovať v polynomiálnom čase. Preto pre dostatočne veľké kľúče sa metóda považuje za bezpečnú.

2.1 Tvorba kľúčového páru

Algoritmus tvorby kľúčového páru je nasledujúci:

1. Vygeneruj dve náhodne prvočísla p a q
2. Vypočítaj $n = p * q$
3. Vypočítaj $\phi(n) = (p - 1) * (q - 1)$
4. Vyber celé číslo e z rozsahu $(1, \phi(n))$
5. Nájdi číslo d, aby platilo $(d * e)\%\phi(n) = 1$

Verejným kľúčom je dvojica (n, e) a súkromným (n, d).

2.2 Šifrovanie a dešifrovanie

Šifrovanie a dešifrovanie sa obecne nelíši, jediný rozdiel je v použití verejného alebo súkromného kľúča. Šifrovanie verejným kľúčom vyzerá nasledovne: $c = m^e\%d$. m je šifrovaná správa. Dešifrovanie súkromným kľúčom vyzerá nasledovne: $m = c^d\%n$.

3 Implementácia aplikácie

Táto kapitola popisuje implementáciu jednotlivých častí aplikácie. Prvá podkapitola sa venuje generovaniu kľúčov, druhá šifrovaniu a dešifrovaniu a tretia podkapitola popisuje spôsoby lámania šifry.

3.1 Generovanie kľúčov

Prvým krokom je potrebné vygenerovať dve prvočísla, p a q. Je vyžadované aby ich súčin mal konkrétnu dĺžku. Pri generovaní náhodných čísel nevieme zaručiť, že ich súčin bude mať konkrétnu dĺžku, tak preto generujeme prvočísla, ktoré majú dĺžku približne polovičnú očakávaného súčinu. Nakoľko nájsť prvočíslo určitej veľkosti je výrazne zložitý problém sam o sebe, je použité generovanie náhodných čísel a za pomoci pravdepodobnostnej metódy Miller-Rabin je overené, že sa jedná o prvočíslo. Pri overovaní sa používa 10 kôl, ktoré podľa nástroja openssl ako aj z knižnice GMP sú dostačujúce. Následne je vypočítané n vynásobením získaných prvočísel. Ak n nespĺňa podmienku dĺžky, menšie prvočíslo je vygenerované znova a tento cyklus je opakovaný, až kým nie sú nájdene dve prvočísla, ktorých súčin je správnej dĺžky.

Pokračuje sa hľadaním e. Nakoľko veľkosť verejného exponentu nijak nepridáva na bezpečnosti, je vybrané číslo z prvej desiatky (ak také neexistuje, pokračuje sa až po $\phi(n)$, a ak neexistuje vôbec, sú vygenerované nové prvočísla), ktoré je nesúdeliteľné s n.

Posledné generované číslo, d, je multiplikatívny inverz. Na počítanie je využité rekurzívne zanorenie, spojené s hľadaním zvyšku po delení a delenca predchádzajúceho volania. Pri vynorovaní je spočítaný inverz. Ak sa takéto číslo nepodarí nájsť, sú vygenerované nové prvočísla.

3.2 Šifrovanie a dešifrovanie

Šifrovanie a dešifrovanie je vykonané za pomoci volania funkcie mpz_powm, ktorá vykonáva umocnenie vo zvyškovej triede. Je len potrebné argumenty predať v správnom poradí.

3.3 Lámanie šifry

Prelomenie širy sa skladá z dvoch krokov:

1. Triviálne delenie - Pri nevhodne zvolených číslach p a q je možné rýchlo a ľahko nájsť číslo, ktoré delí daný modulus. Preto je možné skúšať deliteľnosť prvočíslami. Táto metóda je extrémne rýchla pre nevhodné moduly, avšak pre lámanie dobrých modulov s väčším počtom bitov je nepoužiteľná. Implementácia uvažuje skúšanie prvočísalmi do miliónu. Aby nebolo potrebné tieto čísla počítať, sú staticky uložené v súbore `primes.h`.

2. Faktorizačná metóda Pollard rdo - Na faktorizáciu čísel, ktoré nebolo možné prelomiť za pomoci triviálneho delenia je použitá metóda Pollard rdo. Je založená na generovaní pseudo náhodnej postupnosti z polynómu, konkrétne bol použitý $(x^2 + 1)modn$. 96 bitový kľúč je schopná táto metóda zlomiť za približne 90 sekúnd.

4 Záver

Správa prezentovala vytvorenú aplikáciu, ktorá dokáže vygenerovať súkromný a verejný kľúč. Okrem toho dokáže zašifrovať a dešifrovať správu. V neposlednom rade aplikácia umožňuje prelomenie šifry faktorizáciou verejného modulu. Prelomenie v rozumnom čase je možné len slabých a krátkych (do 100 bitov) kľúčov. Aplikácia bola testovaná na sade testov, dostupných v súbore `tester.sh`.