Jméno a příjmení: Jan Kubiš
Login: xkubis13

Dokumentace k 2. projektu KRY 17/18

(Implementace a prolomení RSA)

- **Obecné informace**

  Pro práci s velkými čísly byla využita doporučená knihovna GMP a odpovídající typ pro celá čísla mpz_t. Jelikož se nejedná o běžný datový typ, nýbrž o ukazatele, při jejich předávání do funkcí je potřeba dbát na to, že jejich změna uvnitř funkce se projeví i navenek. Na určitých místech jsou tedy vytvářeny lokální kopie, se kterými se dál pracuje, abychom těmto side-effectům předešli.

- **Generování parametrů RSA**

  Pro šifrování a dešifrování zprávy pomocí algoritmu RSA je nutné znát veřejný a soukromý klíč. Veřejný klíč je dvojice (e,n), soukromý klíč dvojice (d,n). V této části bude popsáno, jak se tyto klíče generují. Nejprve vypočteme n – veřejný modulus. Veřejný modulus je součinem dvou náhodných, dostatečně velkých prvočísel, běžně označených jako p a q. Jelikož je zadána velikost veřejného modulu v bitech, p a q ???budou zabírat polovinu velikosti???, abychom jejich součinem získali číslo o požadované velikosti. Pro získání p si nejprve pomocí funkce mpz_urandomb vygenerujeme číslo (nikoliv prvočíslo) reprezentováno odpovídajícím počtem bitů. Pokud nemá msb tohoto čísla hodnotu 1, generujeme znovu (požadavek ze zadání). Pokud ano, vlastní funkcí nalezneme prvočíslo, které je větší než tato hodnota (poté ještě provedeme kontrolu, zda má nalezené prvočíslo odpovídající počet bitů, pokud ne – mohlo se rozšířit – provedeme celé generování znovu). Funkce pro hledání následujícího prvočísla funguje následovně: pokud je vstupní číslo sudé, inkrementuj jej o 1 (prvočísla jsou lichá); dále v cyklu ověřuj, zda je toto číslo prvočíslo; pokud ano, vrať jej, pokud ne, inkrementuj jej o 2 (na další liché). Pro ověření, zda je dané číslo prvočíslo, byl implementován algoritmus Miller-Rabin. Algoritmus je zmíněn v zadání projektu a používá jej i funkce knihovny GMP pro hledání následujícího prvočísla, její použití je však v tomto projektu zakázáno. Jednotlivé kroky algoritmu si lze prohlédnout ve zdrojovém kódu, případně v literatuře zmíněné v zadání. Prvočíslo q vygenerujeme analogicky; veřejný modulus n získáme jejich vynásobením.

  Pro výpočet parametrů d a e nejprve vypočítáme proměnnou phi. Phi získáme jednoduše vynásobením hodnot p a q dekrementovaných o 1. Parametr e musí být zvolen tak, že největší společný dělitel e a phi je číslo 1. Abychom minimalizovali počet společných dělitelů, za e budeme postupně dosazovat různá prvočísla – ta jsou dělitelná pouze sebou samým a číslem 1. Tím pádem stačí kontrolovat, zda phi není násobkem zvoleného e. V zadání se píše, že e má většinou hodnotu 3. Čím vyšší prvočíslo zvolíme, tím menší pravděpodobnost je, že phi bude jeho násobkem: např číslem 3 je dělitelné každé třetí číslo, čísla dělitelná prvočíslem 65537 mají periodu mnohem větší (právě každé 65537. číslo). Po výpočtu phi a vhodné volbě e můžeme vypočítat parametr d. K tomu využijeme funkci označovanou jako multiplicative inverse, která je popsána v literatuře zmíněné v zadání. Výpočet využívá rozšířeného eukleidova algoritmu, přesný postup pro toto řešení lze nalézt ve zdrojovém kódu.

  V této fázi jsou již vypočteny všechny požadované parametry, jejichž hexadecimální hodnota je následně vytisknuta na výstup.

- **Šifrování a dešifrování RSA**

  Výpočet ciphertextu respektive plaintextu je založen na znalosti veřejného respektive privátního klíče a jednoduchém vzorci. Knihovna GMP obsahuje funkci mpz_powm, jež je přesnou realizací tohoto vzorce, stačí do ní tedy poslat zvolené parametry.

  Ověření správnosti tohoto řešení bylo testováno s různými referenčními parametry (internet, výukové materiály předmětu KRY) a samozřejmě s parametry vygenerovanými vlastním řešením z předchozího kroku.

- **Prolomení RSA**

  Prolomení RSA není v tomto řešení implementováno.