Matúš Drobuliak
Doc. RNDr. Jiří Tuma, DrSc.
13 September 2015

RSA šifra

Ronald Rivest, Adi Shamir a Leonard Adlemann

1 Úvod

1.1 Asymetrická šifra

V tejto práci sa budem snažiť popísať RSA šifru, ktorá je pomenovaná podľa priezvisk jej autorov (vid. vyššie). Bola vymyslená už v roku 1977 a aj v súčasnej dobe patrí medzi najpoužívanejšie a najobľúbenejšie asymetrické šifry. Čo je úžasné na asymetrickej šifre je, že na rozdiel od symetrickej, odosielateľ a prijímateľ sa nemusia stretnúť a dohodnúť sa na jednotnom spoločnom kľúči, ktorý slúži na šifrovanie aj dešifrovanie správ. To môže byť často fyzicky nemožné, poprípade v dnešnej internetovej dobe, ak kľúč šifry nie je pri prenose dostatočne zabezpečený, tak sa veľmi ľahko môže ocitnúť v zlých rukách. Pri asymetrickej šifre máme dva rôzne kľúče, ktoré sú spolu úzko matematicky prepojené, a kde jeden je verejný a druhý je súkromný - je tajomstvom len príjemcu, nie oboch partnerov. Jeden slúži na šifrovanie správy, zatiaľ čo druhý na jej dešifrovanie. Šifrovacie a dešifrovacie algoritmy musia byť rýchle, zatiaľ čo dešifrovanie bez znalosti súkromného kľúča musí byť početne nezvládnuteľné. Poďme si ukázať niekoľko jednoduchých príkladov ako sa dá takáto šifra využiť.

Pri hľadaní informácií na napísanie tejto práce, som sa pri kryptografických príkladoch dookola stretal s menami Bob a Alica, teda aj ja sa budem držať tejto konvencie. Nasledujúce vysvetlenie je prevzaté a pekne graficky ukázané na videu [1]. Bob si vygeneruje súkromný a verejný kľuč. Verejný zverejní a teda hocikto má k nemu prístup, aj Alica. Alica chce napísať Bobovi tajnú správu tak použije Bobov verejný kľuč na zašifrovanie správy. Tým že nikto iný nemá prístup k Bobovmu súkromnému kľúču Bob je jediný, kto si môže Alicinu zašifrovanú správu dešifrovať a následne prečítať.

Samozrejme toto nie je jediné využitie asymetrickej šifry. Poďme si ukázať druhý príklad. Bob má stále vygenerovaný svoj súkromný a verejný kľuč. No teraz Bob napíše správu a zašifruje ju svojim súkromným kľúčom a tú následne zverejní. Tým, že k jeho verejnému kľúču majú všetci prístup, tak hocikto si môže jeho správu dešifrovať verejným kľúčom a teda prečítať. To, že Alica dešifrovala túto správu implikuje, že ju musel napísať Bob, lebo on jediný má prístup k svojmu súkromnému kľúču. Na tomto princípe funguje digitálny podpis.

Predstavme si, že chcem vydať dôležité prehlásenie, nechcem aby s ním niekto iný manipuloval, ale zároveň chcem, aby si každý mohol overiť, že ja som jeho pravý autor.

Ak by chceli Bob a Alica spolu bezpečne komunikovať bez toho aby sa spolu stretli alebo sa nejako dohodli na kľúči šifry, najlepšia možnosť je, aby tieto dve metódy skombinovali. Ak Bob zašifruje správu svojim privátnym kľúčom zaručí tým, že tá správa je originálne jeho. Následne, ak ju ešte zašifruje Aliciným verejným kľúčom zaručí jej bezpečie. Týmto Bob a Alica majú zabezpečené, že nikto iný si ich správu nemôže prečítať a ani ju modifikovať, lebo k tomu je potrebný Alicin súkromný kľúč. Alica vie, že správa prišla priamo od Boba, lebo po použití Bobovho verejného kľúču správa dáva zmysel.

Verím, že som dostatočne načrtol aká dôležitá a zaujímavá je asymetrická šifra. Jej hlavným reprezentantom je RSA šifra, ktorá je veľmi rozšírená. [2] Využíva ju napríklad SSL protokol čo je nekomerčný otvorený protokol a jedna z najpoužívanejších metód na zabezpečenie dátových prenosov v rámci internetu medzi serverom s webovou prezentáciou a prehliadačom (používateľom). [3] Takisto sa využíva v systéme PGP (Pretty Good Privacy), ktorý slúži najmä na digitálny podpis - teda podpis nahradzujúci vlastnoručný podpis - a overuje identitu odosielateľa. [4] Ako klient Slovenskej Sporiteľne som si zistil, že z dôvodu väčšej bezpečnosti mám možnosť si v zmluve vybrať zasielanie elektronických výpisov šifrované práve týmto systémom. V tejto práci sa dozviete, ako tento šifrovací algoritmus funguje.

2 Matematické pozadie

2.1 Množina $\mathbb{Z}_n$

[5] Množinu modulo n budeme nazývať množinu celých čísel od 0 do $n-1$. Píšeme $\mathbb{Z}_n = \{0,1,2,\dots,n-1\}$.

2.2 Zvyšok po delení

[5] Formálnym spôsobom ako vysvetliť zvyšok po delení iným číslom je ekvivalentný vzťah: $\forall x,y,z,k \in \mathbb{Z}, x \equiv y \bmod z \iff x = k \cdot z + y$, tento vzťah nám hovorí, že keď x je rovný zvyšku y po delení celým číslom z, potom x môže byť napísané ako $x = k \cdot z + y$, kde k je celé číslo.

2.3 Prevrátená hodnota

[5] Prevrátená hodnota (inverzný prvok) čísla x označuje to číslo, ktoré po vynásobení číslom x dáva výsledok 1. Prevrátenú hodnotu čísla x označujeme ako $\frac{1}{x}$ alebo x^{-1} a je definovaná ako

$$x \cdot x^{-1} = 1$$

2.4 Najväčší spoločný deliteľ

[5] Najväčší spoločný deliteľ (NSD) dvoch celých čísel je najväčšie číslo také, že bez zvyšku delí obidve čísla. Napríklad, $NSD(10,15) = 5$. $x \in \mathbb{Z}_p, x^{-1} \in \mathbb{Z}_p \iff NSD(x,p) = 1$ tento

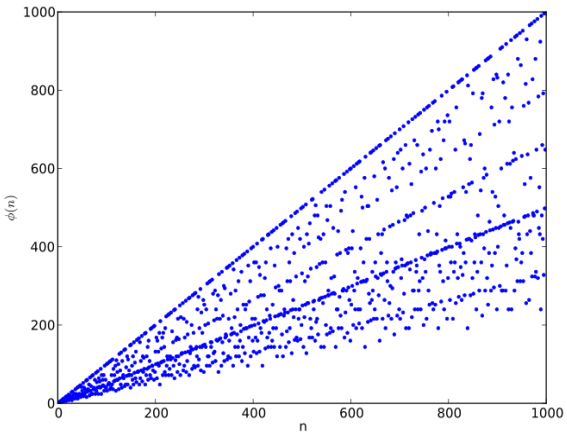
vzťah nám hovorí, že ak dve čísla majú najväčší spoločný násobok 1, potom menšie z nich má prevrátenú hodnotu v množine modulo väčšieho.

Napríklad, NSD(3,4) = 1, takže vieme že $3^{-1} \epsilon \mathbb{Z}_4$ čo je zhodou okolností $3 \Longrightarrow 3 * 3 = 1 \bmod 4$. V opačnom prípade $2 \epsilon \mathbb{Z}_4$ vieme, že 2^{-1} neexistuje v $\mathbb{Z}_4$, lebo NSD(2,4) = $2 \neq 1$.

2.5 Prvočíslo

[5] Prvočíslo, teda prirodzené číslo ktoré je bez zvyšku deliteľné práve dvomi rôznymi prirodzenými číslami, a to číslom jedna a samou sebou, je veľmi dôležité pre RSA. Pre každé prvočíslo p a každé číslo od 1 do $p-1$ majú *NSD* rovný 1 a teda má prevrátenú hodnotu v modulo *p*.

2.6 Eulerova funkcia



[6] Eulerova funkcia, ktorá sa značí $\varphi(n)$ je zobrazenie $\varphi : \mathbb{N}^+ \rightarrow \mathbb{N}^+$ ktoré nám udáva počet všetkých prirodzených čísel *k* takých, že $1 \leq k \leq n$ a *NSD(k,n)* = 1, teda *k* a *n* sú nesúdeliteľné čísla. Čo nás prinieslo k dôležitej rovnici: $\varphi(p) = p - 1,\ p \epsilon \mathbb{P}$ kde $\mathbb{P}$ značí množinu všetkých prvočísel. Napríklad: $\varphi(5) = |\{1,2,3,4\}| = 4$.

([7] Toto môžeme pozorovať aj na grafe Eulerovej funkcie pre celé čísla od 1 do 1000.Všimnite si tú rovnú čiaru bodov na vrchole, ktorá reprezentuje všetky prvočísla.)

2.7 Rozšírený Euklidov algoritmus

[8] Tento algoritmus nám umožňuje nájsť inverzný prvok čísla x na telese $\mathbb{Z}_p$, kde p je prvočíslo, za predpokladu, že $NSD(x,p) = 1$. Princíp spočíva v tom, že sa snažíme rozpísať zvyšky tak, aby boli vyjadrené ako násobky čísiel p a x. Ukážeme si to na príklade: $x = 20$ a $p = 127$. Postupujeme nasledovne, až pokiaľ dostaneme rovnosť $a \cdot 20 + b \cdot 127 = 1$.

$$127 = 6 \cdot (20) + 7 \rightarrow 7 = 127 - 6 \cdot 20$$

$$20 = 2 \cdot (7) + 6 \rightarrow 6 = 20 - 2 \cdot 7 \rightarrow 6 = 20 - 2 \cdot (127 - 6 \cdot 20) \rightarrow 6 = 13 \cdot 20 - 2 \cdot 127$$

$$7 = 1 \cdot (6) + 1 \rightarrow 1 = 7 - 6 \rightarrow 1 = 127 - 6 \cdot 20 - (13 \cdot 20 - 2 \cdot 127) = 3 \cdot 127 - 19 \cdot 20$$

Pretože sme v telese $\mathbb{Z}_{127}$ tak všetky násobky čísla 127 sú rovné 0 a teda dostávame rovnosť $-19 \cdot 20 = 1$, čo upravíme na $108 \cdot 20 = 1$ v $\mathbb{Z}_{127}$ $\Rightarrow$ 108 je inverzný prvok ku 20.

2.8 Eulerova veta

[9] Eulerova veta hovorí: $a^{\varphi(n)} = 1 \bmod n$, kde $\varphi(n)$ je eulerova funkcia a $NSD(a,n) = 1$

Príklad: $a = 4, n = 9,\ \varphi(9) = 6 \Rightarrow 4^6 = 1 \bmod 9 \Leftrightarrow 4096 = 1 \bmod 9$

2.9 Malá Fermatova veta

[10] Je to veta z teórie čísel, ktorá hovorí že pre každé prvočíslo p nesúdelitelné s číslom a platí:

$$a^{p-1} = 1 \bmod p$$

Je to špeciálny prípad Eulerovej vety (2.8).

3 RSA

3.1 Úvod

Štruktúra a informácie v nasledujúcej kapitole sú prevzaté z [5], no sú doplnené ďalšími informáciami z rôznych zdrojov, ktoré budú spomenuté v texte.

RSA šifra sa skladá z dvoch častí:

- Vytvorenie kľúča: algoritmus na vygenerovanie kľúča
- Vyhodnotenie RSA funkcie: funkcia F, ktorá dostane na vstupe text *m* a kľuč *k* a vytvorí zakódovaný text *c*.

3.1 Generovanie kľúču

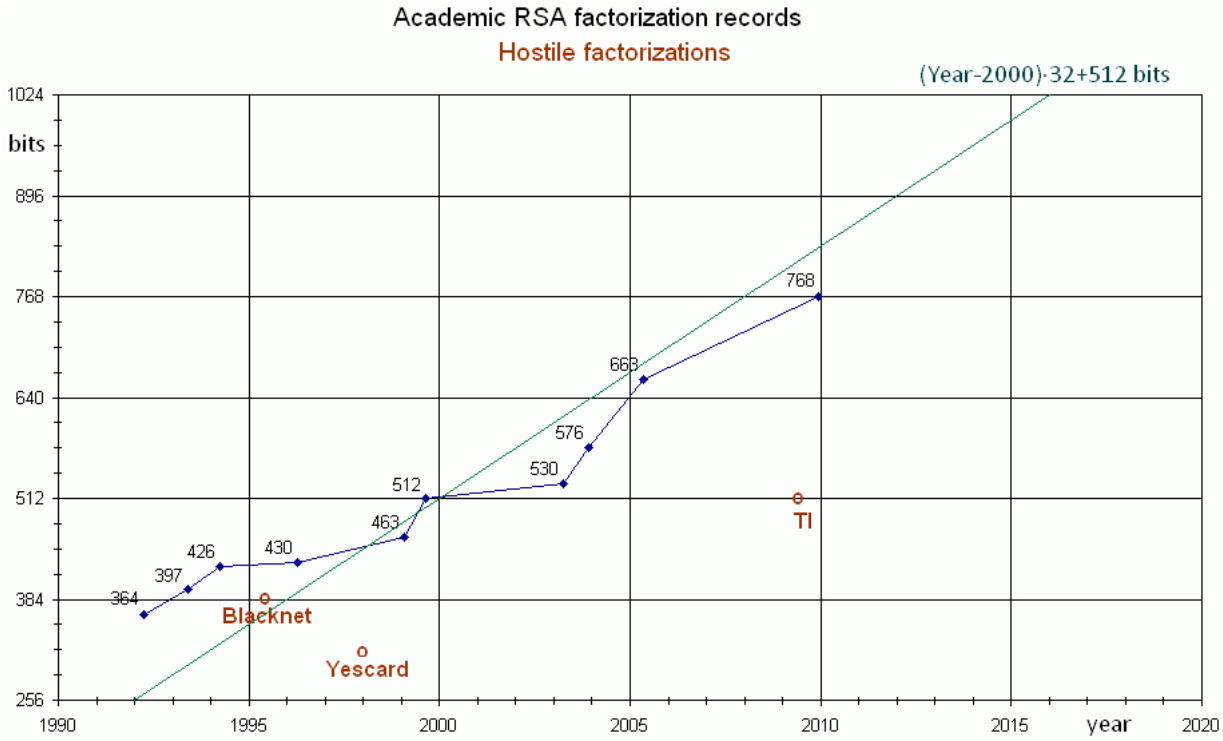
Algoritmus na vytvorenie kľúča RSA šifry je veľmi priamy a jednoduchý. Cieľom je vytvoriť ako verejný tak aj súkromný kľuč. [12] Generovanie kľúča RSA prebieha v troch hlavných krokoch:

1. Voľba *n* (modulus): vyberieme dve rôzne prvočísla *p* a *q*, potom položíme $n = p \cdot q$.
2. Voľba *e* (verejný exponent): *e* zvolíme tak, aby patrilo do oboru hodnôt $[3, \varphi(n)), kde\ \varphi(n) = (p-1) \cdot (q-1)$ a zároveň, aby *e* bolo nesúdeliteľné s číslom $\varphi(n)$.
3. Voľba *d* (privátny exponent): *d* zvolíme tak, aby platilo $e \cdot d \bmod \varphi(n) = 1$

Privátnym kľúčom sa stáva dvojica čísel (n,d) a verejným kľúčom dvojica (n,e).

[13] Pri voľbe p a q treba podotknúť, že bezpečnosť RSA šifry je založená na predpoklade, že rozložiť veľké číslo na súčin prvočísel (faktorizácia) je veľmi náročná úloha a v rozumnom čase je prakticky nemožné ju vyriešiť. Je to z toho dôvodu, že v súčasnej dobe nie je známy žiadny efektívny algoritmus ktorý by riešil tento problém, dokonca sa predpokladá že ani neexistuje. Je teda veľmi dôležité aby p a q boli veľmi veľké prvočísla a aby neboli príliš blízko seba. V súčasnosti sa považuje za bezpečný a odporúča používať kľúč o veľkosti 1024 bitov, čo znamená, že modulus sa skladá z 309 cifier. Dostaneme ho tak, že pri tvorbe modulusu zoberieme prvočísla o polovičnej dĺžky. Samozrejme kvôli zvýšeniu bezpečnosti veľa firiem prechádza aj na väčšiu veľkosť a to poväčšine na 2048 bitový kľúč (617 cifier). [14] Pre zaujímavosť na rozlúštenie RSA o veľkosti 1024 bitov je vypísaná odmena 100,000 USD. Podľa môjho názoru, kým nepríde niekto s novou matematickou myšlienkou ako riešiť problém prvočíselnej faktorizácie, tak jediné čo stačí robiť s rastúcou výkonnosťou počítačov, ktoré budú vedieť urobiť viac operácií za jednotku, zvyšovať veľkosť modulusu a tým sťažovať prelomenie

šifry. *(*[15] *Na obrázku je vidno vývoj faktorizácie za čas – aktuálny rekord je 768 bitové číslo rozlúštené koncom roku 2009.)*



Otázkou zostáva, ako niekto vytvorí tak veľké prvočíslo? Treba si vybrať veľmi veľké náhodné číslo a otestovať ho, či je to prvočíslo. Ak neprejde týmto testom, potom pridáme jedna a začneme odznovu. Poznáme viacero testov, ktoré nám toto vedia zabezpečiť. Jeden rýchly prvočíselný test sa nazýva Millerov-Rabinov test prvočíselnosti, ktorý nám s pravdepodobnosťou $1 - \frac{1}{4^k}$, kde k je počet iterácií testu, určí či dané číslo je prvočíslo.

Následne po tom čo máme naše prvočísla, modulus zistíme veľmi jednoducho, položíme $n = p \cdot q$. S prvočíselným rozkladom čísla *n*, vieme veľmi jednoducho vypočítať Eulerovu funkciu $\varphi(n) = (p-1) \cdot (q-1)$. Čo je odvodené z rovnice (2.6) $\varphi(n) = \varphi(p \cdot q) = \varphi(p) \cdot \varphi(q) = (p-1) \cdot (q-1)$. Ak by niekto vedel určiť prvočíselný rozklad modulusu *n*, potom sa RSA šifra stane veľmi zraniteľnou.

Následne je určený verejný kľuč. Značíme ho *e*, je to prvočíslo vybrané z rozsahu $[3, \varphi(n))$. Často býva zvolené číslo 65537. Toto číslo je prvočíslo, a teda je celkom veľká šanca že splňuje druhú podmienku - že je nesúdelitelné s $\varphi(n)$. Ak by to tak nebolo, treba vybrať iné. A teda verejným kľúčom sa stane dvojica (n,e).

Pretože verejný kľúč *e* je nesúdelitelné s $\varphi(n)$ a teda ich NSD je 1, podľa (2.4) vieme, že inverzný prvok existuje a môže byť rýchlo spočítaný *rozšíreným Euklidovým algoritmom* (2.7). Tento inverzný prvok *d* sa teda stáva naším súkromným kľúčom spolu s číslom *n*.

3.2 Vyhodnotenie RSA funkcie

Toto je proces, kde šifrujeme a dešifrujeme text správy. V prípade správy *m* a kľúču *k* správu zašifrujeme RSA funkciou *F* nasledovne:

$$F(m,k) = m^k \bmod n$$

Máme dva prípady:

1. zašifrovanie verejným kľúčom a dešifrovanie privátnym kľúčom.

Šifrovanie: $F(m,e) = m^e \bmod n = c$, kde *m* je správa, *e* je verejný kľúč a *c* je zašifrovaná správa. Dešifrovanie**:** $F(c,d) = c^d \bmod n = m$

2. zašifrovanie privátnym kľúčom a dešifrovanie verejným kľúčom.

Šifrovanie: $F(m,d) = m^d \bmod n = c$, kde *m* je správa, *d* je súkromný kľúč a *c* je zašifrovaná správa. Dešifrovanie**:** $F(c,e) = c^e \bmod n = m$

Všimnite si, že tieto dva prípady sú k sebe navzájom zrkadlové.

4 Fungovanie RSA

4.1 Úvod

V tejto časti si ukážeme, prečo je dôležité, aby kľúče boli k sebe inverzné a potom si dokážeme správnosť RSA. Nasledujúce vysvetlenie je prevzaté z [11].

4.2 Kľúče a zámena

Vieme, že môžeme šifrovať súkromným, ale aj verejným kľúčom, pričom dešifrovať budeme opačným k tomu ktorý si vyberieme. Ideme si teda ukázať, že platí: keď správu *m* zašifrujem verejným kľúčom a následne dešifrujem súkromným, dostanem originálnu správu, ktorú by som tak isto dostal ak by som ju zašifrovať súkromným a následne dešifroval verejným. Matematicky to zapíšeme takto:

$$D\big(d,E(e,m)\big) = m = D(e,E(d,m)) \qquad \textbf{(a)}$$

Kde funkcia E (encription) nám značí šifrovanie a D (decription) dešifrovanie správy.

Vieme, že v poslednom kroku pri generácii kľúču musíme zvoliť súkromný kľúč tak, aby platila rovnosť: $e \cdot d = 1 \bmod \varphi(n)$, ktorá nám hovorí, že po vynásobení súkromného a verejného kľúča dostaneme 1 v mod $\varphi(n)$. Násobenie je komutatívne, čo znamená že keď zameníme *e* a *d*, rovnosť: $d \cdot e = 1 \bmod \varphi(n)$ bude stále platiť. To je veľmi kľúčové, lebo keď zakódujeme našu správu *m* súkromným kľúčom *d* dostaneme zašifrovaný text *c*.

$$m^d \bmod n = c$$

Nato, aby sme dostali späť originálnu správu *m*, chceme docieliť aby sme *m* umocnili na 1, čím dostaneme *m*. Teda, keď umocníme $c^e = m^{d\cdot e} = m^1 = m$, dostaneme našu originálnu správu. Toto teraz využijeme nato aby sme ukázali rovnosť (a).

$$D\big(d,E(e,m)\big) = D(d,m^e \bmod n) = m^{\boldsymbol{e\cdot d}} \bmod n = m^{\boldsymbol{d\cdot e}} \bmod n$$
$$= D(e,m^d \bmod n) = D(e,E(d,m))$$

Kde zvýraznený text je vyššie spomínaná komutatívna vlastnosť.

4.3 Dôkaz, že RSA funguje

Ideme dokázať fundamentálnu rovnicu RSA šifry, ktorá nám hovorí, že ak máme správu m, po zašifrovaní kľúčom k dostaneme šifrovanú správu c, ktorú keď dešifrujeme opačným kľúčom, dostaneme originálnu správu m. Dôkaz je prevzatý z [16]. Pre dôkaz použijeme verejný kľúč e ako šifrovací. Zapíšeme takto: $\forall\, m \in \mathbb{Z}_n : D\big(d, E(e,m)\big) = m$, kde $E(e,m) = c$. Ideme dokázať jej správnosť: Potrebujeme ukázať, že $(m^e \bmod n)^d \bmod n = m$, pre všetky $m \in \mathbb{Z}_n$. Špeciálny prípad je m = 0. Vtedy $E(e,m) = 0$ a $D(d,0) = 0$, preto tvrdenie platí. Pre $m \in \mathbb{Z}_n \setminus \{0\}$ budeme uvažovať dva prípady: $NSD(n,m) = 1$ a $NSD(n,m) \neq 1$. Vieme, že $e \cdot d = 1 \bmod \varphi(n) \Leftrightarrow e \cdot d = 1 + k \cdot \varphi(n), k \in \mathbb{Z}$ (2.2)

1. $NSD(n,m) = 1$. Počítame:

$$D\big(d, E(m)\big) = (m^e \bmod n)^d \bmod n =$$
$$= m^{e\cdot d} \bmod n = m^{1+k\cdot\varphi(n)} \bmod n =$$
$$= m \cdot \big(m^{\varphi(n)}\big)^k \bmod n = m \bmod n = m$$

Predposledná rovnosť vyplýva z Eulerovej vety (2.8)

2. $NSD(n,m) \neq 1$. Potom buď p delí m alebo q delí m (nie však obe súčastne, lebo($0 < m < n$). Bez ujmy na všeobecnosti budeme predpokladať, že $m = l \cdot p^s$, kde $s \geq 1$ a $NSD(l,n) = 1$ $(s, l \in \mathbb{N})$. Potom:

$$D\big(d, E(e,m)\big) = m^{e\cdot d} \bmod n =$$
$$= (l \cdot p^s)^{1+k\cdot\varphi(n)} \bmod n =$$
$$= l \cdot (p^{1+k\cdot\varphi(n)})^s \bmod n \quad \textbf{(b)}$$

Z malej Fermatovej vety (2.9) vieme, že $q \in \mathbb{P}$, $p^{q-1} = 1 \bmod q$. Odtiaľ:

$$p^{(q-1)\cdot(p-1)} \equiv 1 \pmod q$$
$$p^{k\cdot\varphi(n)} = 1 + a \cdot q, \text{ pre nejaké } a \geq 1$$
$$p^{k\cdot\varphi(n)+1} = p + a \cdot p \cdot q = p + a \cdot n$$
$$p^{k\cdot\varphi(n)+1} \equiv p \pmod n.$$

Po dosadení do (b) dostaneme:

$$D\big(d, E(e,m)\big) = l \cdot p^s \bmod n = m.$$

Dokázali sme fundamentálnu rovnicu RSA.

5 Príklad

5.1 Výpočet kľúču

Alica a Bob chcú spolu tajne komunikovať a zvolia si protokol RSA. Alica si náhodne zvolí dve prvočísla, vypočíta modulo a hodnotu Eulerovej funkcie. (Aby bola ukážka názorná

zvolíme malé prvočísla, v reálnom prípade by sme zvolili prvočísla, ktoré majú aspoň 160 cifier).

$$p = 13, q = 29$$

$$n = 13 \cdot 29 = 377$$

$$\varphi(377) = \varphi(13 \cdot 29) = (13 - 1) \cdot (29 - 1) = 12 \cdot 28 = 336$$

Teraz si Alica zvolí číslo $e = 43$ a pomocou rozšíreného Euklidovho algoritmu spočíta jeho inverzný prvok v $\mathbb{Z}_{\varphi(n)}$.

$$336 = 7 \cdot (43) + 35 \rightarrow 35 = 336 - 7 \cdot (43)$$

$$43 = 1 \cdot (35) + 8 \rightarrow 8 = 43 - 1 \cdot (35)$$

$$35 = 4 \cdot (8) + 3 \rightarrow 3 = 35 - 4 \cdot (8)$$

$$8 = 2 \cdot (3) + 2 \rightarrow 2 = 8 - 2 \cdot (3)$$

$$3 = 1 \cdot (2) + 1 \rightarrow 1 = 3 - 1 \cdot (2)$$

Po dosadení dostaneme rovnosť $1 = 16 \cdot (336) - 125 \cdot (43) \Longrightarrow 1 = -125 \cdot 43$

$$d = -125^{-1} \bmod (336) = 211 \bmod (336)$$

Alica teraz zverejní svoj verejný kľúč (377,43). A svoj súkromný (377,211) si ponechá.

Bob si zvolí náhodne dve prvočísla, vypočíta modulo a hodnotu Eulerovej funkcie.

$$p = 17, q = 31$$

$$n = 17 \cdot 31 = 527$$

$$\varphi(527) = \varphi(17 \cdot 31) = (17 - 1) \cdot (31 - 1) = 16 \cdot 30 = 480$$

Bob volí číslo $e = 19$ a pomocou rozšíreného Euklidovho algoritmu spočíta jeho inverzný prvok v $\mathbb{Z}_{\varphi(n)}$.

$$d = 19^{-1} \bmod (480) = 43 \bmod (480)$$

Bob teraz zverejní svoj verejný kľúč (527,19). A svoj súkromný (527,379) si ponechá.

5.2 Komunikácia

Pri nasledujúcich výpočtoch som využil online kalkulačku pre veľké čísla [17]. Bob chce teraz poslať Alice správu. Správou *m* bude pre jednoduchosť 15 čo môžu byť súradnice, ale napríklad aj znak z ASCII tabuľky, to už záleží na predošlej dohode Alice a Boba. Bob má prístup k verejnému kľúču Alice a zašifruje správu ako:

$$c = 15^{43}\ mod\ (377)$$

$$c = 362$$

Zašifrovanú správu pošle Alici a tá ju rozlúšti pomocou svojho súkromného kľúča:

$$m = 362^{211}\ mod\ (377)$$

$$m = 15$$

Ako som spomínal v úvode, keby si chcela byť Alica istá, že správa je od Boba, Bob mohol zašifrovanú správu *c* zašifrovať ešte raz svojím súkromným kľúčom a dostal by správu *z*:

$$z = 362^{379}\ mod\ 527$$

$$z = 385$$

Potom by správu *z* poslal Alice, ktorá by ju ako prvé šifrovala Bobovym súkromným kľúčom:

$$385^{19}\ mod\ 527 = 362$$

A následne by ju dešifrovala svojím súkromným (bolo vypočítané vyššie) a mala by originálnu správu *m* = 15.

6 Záver

6.1 Zhrnutie a názor

Povedali sme si čo je to asymetrická šifra a ukázali sme si pár príkladov jej využitia. Následne sme si popísali činnosť algoritmu a podrobnejšie vysvetlili v čom spočíva – teda v probléme faktorizácie, čo je problém na ktorý nepoznáme efektívne riešenie. Viac krát som zdôraznil, že je veľmi dôležité vybrať veľké prvočísla, ktoré nám zaručia bezpečnosť. Na záver sme si dokázali funkčnosť tohto algoritmu a ukázali si ho na praktickom príklade.

Túto prácu som si vybral po debate s mojim spolubývajúcim programátorom kde sme riešili bezpečnosť a súkromie na internete, čo si myslím že v dnešnej dobe, kedy sa technológia dostáva do nášho každodenného života bude stále aktuálnejšou otázkou. Čo sa mi páči na RSA šifre je, že je tu už od roku 1977, jej algoritmus je veľmi jednoduchý a aj tak, keď je správne použitá, je nerozlúštiteľná. Verím, že každý s dostatkom času a chute môže pochopiť matematiku za ňou. Bol som prekvapený, že moje najnavštevovanejšie stránky ako Gmail a Facebook využívajú SSL protokol, ktorý funguje na RSA algoritme, čo som predtým nevedel. Ak by som v budúcnosti mal internetovú stránku, na ktorej by moji zákazníci museli zverejniť svoje citlivé informácie, jednoznačne by som ju zabezpečil certifikátom využívajúcim RSA algoritmus. Na druhú stranu aj keď ma RSA šifra utvrdila vo svojej bezpečnosti, keby som ju potreboval využiť pri komunikácií a zašifrovaní dlhého textu, vtedy by som ju nevyužil kvôli časovej náročnosti, no skôr by som to riešil tak, že by som ňou zašifroval napríklad kód symetrickej šifry. Tým by som zabezpečil bezpečný prenos symetrickej šifry druhej strane a so symetrickou by sa rozlúštil text časovo rýchlejšie.

7 Zdroje

7.1 Zdroje

[1] *Youtube, Public Key Cryptography – Computerphile*,
https://www.youtube.com/watch?v=GSIDS_lvRv4

[2] *Ssl-thawte*, http://www.ssl-thawte.sk/inpage/co-je-to-ssl/
[3] *Webopedia, Pretty Good Privacy*,
http://www.webopedia.com/TERM/P/Pretty_Good_Privacy.html
[4] *Slovenská sporiteľňa, bezpečnosť mailbankingu*,
https://www.slsp.sk/sk/ludia/non-product/bezpecnost-mailbankingu
[5] *Doctrina, Barry Steyn*, http://doctrina.org/How-RSA-Works-With-Examples.html
[6] *Wikipedia,Eulerova funkcia*, https://cs.wikipedia.org/wiki/Eulerova_funkce
[7] *Math.stackexchange, obrazok*,
http://math.stackexchange.com/questions/263243/lines-in-the-euler-phi-graph
[8] *Algoritmy, Euklidov algoritmus*, http://www.algoritmy.net/article/44/Eukliduv-algoritmus
[9] *Algoritmy, Eulerova veta*, http://www.algoritmy.net/article/57/Eulerova-veta
[10] *Algoritmy, Mala Fermatova veta*,
http://www.algoritmy.net/article/59/Mala-Fermatova-veta
[11] *Doctrina, Barrz Steyn*,
http://doctrina.org/Why-RSA-Works-Three-Fundamental-Questions-Answered.html
[12] *Cleverandsmart, generovanie kľúču,*
http://www.cleverandsmart.cz/zaklady-kryptografie-pro-manazery-rsa/
[13] *Kryptografia, bezpečnosť RSA*, http://www.kryptografie.wz.cz/data/RSA.htm
[14] *Wikipedia, RSA factoring challenge*,
https://en.wikipedia.org/wiki/RSA_Factoring_Challenge
[15] *Stackexchange, obrazok*, http://crypto.stackexchange.com/questions/1978/how-big-an-rsa-key-is-considered-secure-today
[16] *Martin Stanek, RSA*, http://mayor.fri.uniza.sk/krypto/09/rsa.pdf
[17] *Javascripter, kalkulačka*,
http://www.javascripter.net/math/calculators/100digitbigintcalculator.htm