Základní metody z kryptografie

Ukázky aplikací matematiky

Zdeněk Pustějovský

25. září 2019

Obsah

1 Úvod

Jako téma své práce jsem si vybral popis a vysvětlení dvou základních kryptografických metod, které se s jistými úpravami používají dodnes - RSA a Diffie-Hellmanův protokol. Také se pokusím vysvětlit alespoň klasickou část Shorova algoritmu. Tento algoritmus je vymyšlen pro kvantové počítače, já se ale jeho kvantové části pro jeho složitost vyhnu. Na začátek vysvětlím podstatu, rozdíly a nevýhody symetrických a asymetrických šifrovacích metod.

2 Symetrické šifrování

Dnes se každý den na internetu vymění obrovské množství informací a mnoho z nich je citlivých, tedy nechceme, aby se dostaly do nesprávných rukou. Nejjednodušším řešením je zprávu zašifrovat, zašifrovanou ji odeslat a adresát, pokud ví jak, ji dešifruje a může si ji přečíst.

Jak ale poznat, že je naše šifra bezpečná? Toto není jednoduchá otázka a nejlepším způsobem je nechat co nejvíc lidí se pokusit o její prolomení, což ale znamená zveřejnit její detaily. Z tohoto důvodu se dnes používaji tzv. *šifrovací protokoly*, kde protokol je veřejný, ale šifruje se podle nějakého klíče, typicky čísla, které znají jenom korespondenti. Oba používají stejný klíč pro zašifrování i dešifrování zpráv a tomuto způsobu se říká *Symetrické šifrování*.

Jeden z dnes nejpoužívanějších protokolů v symetrickém šifrování je *AES - Advanced Encryption Standard*. Velmi zhruba si ho popíšeme. Je to blokova šifra, tedy zpráva je šifrovaná po blocích pevně dané délky - u *AES* je to 128 bitů a klíč může mít 128, 192 nebo 256 bitů. Pracuje s maticí 4 x 4 bytů, které se říká *stav* a na začátku je to nezašifrovaný 128-bitový blok . Z klíče se nejdříve odvodí podklíče - tzv. *rundovací klíče*. Každý byte *stavu* se zkombinuje s jedním podklíčem. Pak se podle určitých pravidel zaměňují byty ve *stavu*, byty v každém řádku se posouvají o určitý přesah a ještě se kombinují v rámci jednotlivých řádků. Tato sekvence generování klíčů, zaměňování, posouvání a kombinování se provádí několikrát v závislosti na velikosti klíče.

AES je považovaná za velmi bezpečnou, používá ji i Americká vláda nebo velké firmy jako Youtube. Dnes jsou známy pouze nepatrně lepší metody na rozluštění, než je pouhé hádání klíče. S dostatečně dlouhým klíčem není ani na tom nejlepším počítači prakticky žádná šance na nalezení klíče v nějakém přiměřeném čase.

Symetrické šifrování samo o sobě je tedy prakticky neprolomitelné. Má ale jednu slabinu. Korespondenti se musí nejdříve nějak dohodnout na klíči, který budou používat. Mohou se sejít třeba osobně, ale to je v dnešní době nepraktické a ve většině případů nemožné. Je tedy potřeba nějaký jiný způsob a jedním z nich je tzv. *asymetrické šifrování*.

3 Asymetrické šifrování

Každý korespondent má dva klíče - jeden soukromý a jeden veřejný. Oba klíče lze použít k zašifrování zprávy i k dešifrování zprávy zašifrované druhým klíčem z páru. Pokud mají korespondenti *Áčko a Béčko* každý svůj vlastní pár klíčů, mohou si podle níže popsaného postupu bezpečně psát a také se jeden druhému verifikovat.

- Pokud *Áčko* chce poslat zprávu *Béčku*, zašifruje svojí zprávu pomocí veřejného klíče *Béčka* a pošle mu ji. *Béčko* ji jako jediný může dešifrovat, protože pouze on zná svůj soukromý klíč.
- Pokud se *Áčko* chce verifikovat *Béčku*, udělá to následovně: nejdřív zašifruje zprávu svým soukromým klíčem a potom *Béčkovým* veřejným klíčem. Béčko dešifruje nejdřív pomocí svého soukromého klíče a potom veřejným klíčem *Áčka*. Pokud se mu takto podaří zprávu dešifrovat, tak ví, že ji poslalo *Áčko*.

Aby bylo asymetrické šifrování použitelné, potřebujeme, aby vygenerované klíče splňovaly tyto vlastnosti:

1. Aby klíče v páru byly zaměnitelné.
2. Aby generování takových párů klíčů bylo výpočetně rychlé.
3. Aby nalezení jednoho ze znalosti druhého bylo výpočetně pomalé.

Jeden z prvních protokolů splňujících tyto vlastnosti si teď popíšeme a dokážeme jeho správnost. Jmenuje se *RSA* - Rivest-Shamir-Adleman, podle svých autorů. Byl představen roku 1977 a jeho obměněné, vylepšené verze se používají dodnes.

4 RSA

4.1 Potřebné věty bez důkazu

Na začátek uvedu tři věty bez důkazu, které využijeme později v důkazu toho, že protokol skutečně funguje, jak má. Věta o *Bézoutově rovnosti* a *Čínská věta o zbytcích* jsou převzaty ze skript *Základy algebry* od D. Stanovského, str. 17 respektive str. 21. Znění *Multinomické věty* je [*zde na str. 2*].

Bézoutova rovnost: pro každou dvojici $a, b \in \mathbb{N}$ existují $u, v \in \mathbb{Z}$ takové, že $NSD(a, b) = ua + vb$.
Multinomická věta: nechť $m, n \in \mathbb{N}, x_1, ..., x_m \in \mathbb{R}$, potom

$$(x_1 + ... + x_m)^n = \sum_{k_1+...+k_m=n} \frac{n!}{k_1!...k_m!} \prod_{t=1}^{m} x_t^{l_t}$$

Čínská věta o zbytcích: nechť $m_1, ..., m_n \in \mathbb{N}$ jsou vzájemně nesoudělná, $M = m_1 \cdot ... \cdot m_n$, potom pro libovolná $u_1, ..., u_n \in \mathbb{Z}$ existuje právě jedno $x \in \{0, ..., M-1\}$, které řeší soustavu kongruencí

$$x \equiv u_1 \pmod{m_1}$$

$$...$$

$$x \equiv u_n \pmod{m_n}$$

4.2 Generace klíčů

Začneme tím, že si vybereme 2 prvočísla p, q a položíme $n = pq$. Potom vybereme číslo $e \in \mathbb{N}$ tak, aby platilo $NSD(e, (p-1)(q-1)) = 1$. K němu najdeme $d, k \in \mathbb{Z}$ taková, že $ed - k(p-1)(q-1) = 1$. To můžeme udělat protože e a $(p-1)(q-1)$ jsou nesoudělná a z *Bézoutovy rovnosti* plyne, že d, k existují. Najít je lze efektivně pomocí rozšířeného Eukleidova algoritmu.

Jako veřejný klíč použiju pár $V = (e, n)$ a jako soukromý budu mít pár $S = (d, n)$. V tajnosti musím držet taky p, q, k. Pro jednoduchost předpokládejme, že zprávu zašifruju podle nějakého protokolu jako číslo X, kde $2 \leq X \leq n$. Pokud chci zprávu poslat, použiju veřejný klíč adresáta a pošlu mu číslo $Y = X^e \pmod{n}$. Adresát po přijmutí zprávy použije svůj soukromý klíč a spočítá $Y^d = X^{ed} \pmod{n}$. No a protože platí $X^{ed} = X \pmod{n}$ (to si za chvíli dokážeme), získá adresát původní X a z něj díky znalosti protokolu i původní zprávu.

4.3 Správnost RSA

Než si dokážeme výše uvedený vztah, potřebujeme dokázat ještě jednu větu. Říká se jí *Malá Fermatova věta*, formuloval ji roku 1640 Pierre de Fermat a dokázal ji Euler roku 1736. Věta i důsledek jsou dokázány [*zde na str. 2*].
Věta: nechť p je prvočíslo, $a \in \mathbb{N}$, potom

$$a^p \equiv a \pmod{p}$$

Důkaz: Věta se dá dokázat mnoha způsoby, například indukcí podle a. Tento důkaz používá multinomickou větu. Platí:

$$a^p = (1 + ... + 1)^p = \sum_{n_1+...+n_a=p} \frac{p!}{n_1!...n_a!} 1^{n_1}...1^{n_a} \equiv 1^p + ... + 1^p \equiv a \pmod{p}$$

Protože $\frac{p!}{n_1!...n_a!} \equiv 0 \pmod{p}$ platí vždy až na případ, kdy $n_k = p$ pro $1 \leq k \leq a$. Zároveň platí $n_1 + ... + n_a = p$, proto v takovém případě jsou ostatní členy nulové a tím pádem je celý zlomek roven 1 a celý sčítanec také. Takových sčítanců je a, ostatní jsou rovny $0 \pmod{p}$ a odtud plyne tvrzení.
□

Důsledek: nechť p je prvočíslo, $a \in \mathbb{N}$ a $NSD(a, p) = 1$, potom

$$a^{p-1} \equiv 1 \pmod{p}$$

Důkaz: Z Bézoutovy rovnosti plyne, že existují $x, y \in \mathbb{Z}$ taková, že $ax + py = 1$. Jinými slovy $ax \equiv 1 \pmod{p}$. Použitím *Malé Fermatovy věty* dostaneme $a^{p-1} \equiv xa^p \equiv xa \equiv 1 \pmod{p}$.
□

Konečně můžeme přistoupit k důkazu, že adresát po umocnění dostane v mod p aritmetice moje původní X. Toto tvrzení i s důkazem je [*odsud, str. 3*].

Tvrzení: nechť e, d jsou čísla popsaná výše, $2 \leq X \leq n$, $X \in \mathbb{N}$, potom

$$X^{ed} \equiv X \pmod{n}$$

Důkaz: Výše jsme e, k volili tak, aby platilo

$$ed = 1 + k(p-1)(q-1)$$

Pokud $X \equiv 0 \pmod{p}$, potom určitě $X^{ed} \equiv X \pmod{p}$. Pokud toto neplatí, potom $NSD(X, p) = 1$, protože p je prvočíslo a tedy z **Důsledku** plyne $X^{p-1} \equiv 1 \pmod{p}$ a tím pádem

$$X^{ed} \equiv X^{1+k(p-1)(q-1)} \equiv X(X^{p-1})^{k(q-1)} \equiv X1^{k(q-1)} \equiv X \pmod{p}$$

Tím jsme dokázali $X^{ed} \equiv X \pmod{p}$. Analogicky lze dokázat $X^{ed} \equiv X \pmod{q}$. Z těchto dvou kongruencí a *Čínské věty o zbytcích* vyplývá $X^{ed} \equiv X \pmod{n}$.

□

4.4 Bezpečnost RSA

Bezpečnost RSA stojí na dvou dnes těžko vyřešitelných problémech.

1. Je velmi jednoduché vzít dvě velká prvočísla a vynásobit je, ale pokud jsou zvolena nezávisle(ideálně náhodně a aby se jejich délka lišila o několik číslic), je docela těžké ze znalosti jejich součinu tato dvě čísla zjistit. Ani dnešní nejlepší algoritmy nedokáží dostatečně rychle najít p a q. Pokud by nějaký způsob existoval, pak by šlo snadno ze znalosti e zjistit d a tím pádem RSA prolomit.

2. Druhý problém je, že neexistuje efektivní metoda pro nalezení neznámé x v rovnici $y \equiv x^e \pmod{n}$, kde e, n je veřejný klíč. Po nalezení x by už útočníkovi stačilo jen dešifrovat x pomocí protokolu, který je veřejný. Tomuto problému se říká *RSA problém.*

4.5 Praktické využití

I když je efektivní nalezení prvočíselného rozkladu čísla těžký problém, dnešní algoritmy jsou mnohem lepší, než metody pro prolomení symetrických šifer. Asymetrické šifrování je pořád principiálně bezpečné, ale jsou potřeba mnohem větší klíče, než u symetrického a tím pádem je mnohem pomalejší. Z tohoto důvodu se v praxi používá kombinace obou - pomocí asymetrického se korespondenti domluví na klíči, který pak používají pro symetrické šifrování.

5 Shorův algoritmus

V této sekci se pokusím zjednodušeně popsat algoritmus, který teoreticky umí najít prvočíselný rozklad v polynomiálním čase (vzhledem k počtu číslic v součinu). S jeho použitím bychom mohli dostatečně rychle najít p a q, prolomit asymetrické šifrování a díky tomu i symetrické. Problém je v tom, že algoritmus je navržen pro kvantové počítače a ty dnes ještě zdaleka nejsou na takové úrovni, aby se dal použít. První rozklad na kvantovém počítači se podařil IBM v roce 2001 a to čísla 15. Dnes umíme rozložit čísla v řádu destitisíců. Kvůli složitosti se vysvětlení kvantové části algoritmu vyhnu. Nejprve si uvedeme dvě potřebné definice a jednu větu bez důkazu. Definice *Eulerovy funkce* a *Eulerovy věty* jsou ze skript *Základy algebry* od D. Stanovského, str. 19, respektive str. 20. Definice *periody* vyplývá [*odsud, str. 4*].

Definice: *Eulerova funkce* $\varphi(n)$ pro $n \in \mathbb{N}, n > 1$ udává počet čísel v intervalu $1, ..., n-1$ nesoudělných s číslem n.
Definice: řekneme, že $r \in \mathbb{N}$ je *periodou čísla* $x \in \mathbb{N}$ *v aritmetice* mod n, pokud r je nejmenší číslo splňující

$$x^r \equiv 1 \pmod{n}$$

Eulerova věta: nechť $a, n \in \mathbb{N}$ jsou nesoudělná, potom

$$a^{\varphi(n)} \equiv 1 \pmod{n}$$

Pro jednoduchost předpokládejme, že n je součin dvou prvočísel p, q, tedy $n = pq$, a my tato prvočísla chceme najít. Nejprvé náhodně zvolíme $a \in \mathbb{N}, 1 < a < n$ a spočítáme $NSD(a, n)$ pomocí Eukleidova algoritmu. Pokud

$NSD(a,n) \neq 1$, našli jsme p, tedy i q a jsme hotovi. Pokud $NSD(n,a) = 1$, najdeme *periodu čísla a v aritmetice* mod n. Právě k provedení tohoto kroku se využívá možností kvantového počítače. Eulerova věta nám říká, že perioda čísla a dělí $\varphi(n)$. Pokud je r liché nebo $a^{\frac{r}{2}} + 1 \equiv 0 \pmod{n}$, musíme vybrat jiné a a začít od začátku. Víme, že

$$a^r \equiv 1 \pmod{n}$$
$$a^r - 1 \equiv 0 \pmod{n}$$

Tedy $a^r - 1$ dělí n. Proto musí existovat nějaké $k \in \mathbb{N}$ takové, že

$$a^r - 1 = kn$$

a protože r je sudé, můžeme psát

$$(a^{\frac{r}{2}} + 1)(a^{\frac{r}{2}} - 1) = kn = kpq$$

Z toho plyne, že p i q dělí jeden z činitelů na levé straně. Zároveň ale nemohou dělit stejný činitel, protože ten by pak byl dělitelný n a to nelze, protože jsme předpokládali $a^{\frac{r}{2}} + 1 \not\equiv 0 \pmod{n}$ a z definice periody plyne $a^{\frac{r}{2}} - 1 \not\equiv 0 \pmod{n}$. Tedy p a q dělí aždé jiný činitel na levé straně. Bez újmy na obecnosti můžeme předpokládat, že $p | a^{\frac{r}{2}} + 1$ a $q | a^{\frac{r}{2}} - 1$ a z toho plyne, že $p = NSD(n, a^{\frac{r}{2}} + 1)$ a $q = NSD(n, a^{\frac{r}{2}} - 1)$. Tímto jsme našli prvočíselný rozklad n.

Pokud bychom chtěli popsat algoritmus krokově, vypadal by takto:

1. náhodně zvolím $1 < a < n$
2. spočítám $NSD(a,n)$
3. pokud $NSD(a,n) \neq 1$, našel jsem p a jsem hotov
4. jinak najdu *periodu $r \in \mathbb{N}$ čísla a v aritmetice* mod n
5. pokud r je liché, musím zpět ke kroku 1
6. pokud $a^{\frac{r}{2}} + 1 \equiv 0 \pmod{n}$, musím zpět ke kroku 1
7. p a q jsou rovny $NSD(n, a^{\frac{r}{2}} + 1)$ a $NSD(n, a^{\frac{r}{2}} - 1)$

6 Diffie-Hellmanův protokol

Mimo násobení existují i jiné operace, které jdou udělat jedním směrem relativně jednoduše (jako vynásobit dvě vhodně zvolená čísla) a druhým je to o dost těžší (jako najít rozklad). Jednu takovou využívá *Diffie-Hellmanův protokol*, který roku 1976 publikovali Whitfield Diffie a Martin Hellman. Tento protokol, stejně jako RSA, popisuje, jak se dvě strany mohou domluvit na klíči přes veřejný kanál, který je odposloucháván, aniž by nikdo klíč odhalil. Nejprve opět potřebné definice. Definice *grupy* je zjednodušená verze definice ze skript *Základy algebry* od D. Stanovského, str. 78. Zbylé definice jsou z těch stejných skript, po řadě str. 83, 80 a 84.
Definice: *grupou* **G** nazýváme neprázdnou množinu G společně s binární operací $\cdot$ splňující

1. $\forall a, b \in G : a \cdot b \in G$
2. $\forall a, b, c \in G : (a \cdot b) \cdot c = a \cdot (b \cdot c)$
3. $\exists e \in G \; \forall a \in G : e \cdot a = a \cdot e = a$
4. $\forall a \in G \; \exists a^{-1} \in G : a \cdot a^{-1} = a^{-1} \cdot a = e$

Definice: *řádem grupy* **G** rozumíme velikost množiny G

Definice:$(g_1, ..., g_n) \in G$ nazveme *generátory grupy* **G**, pokud lze kombinací dvojic $g_i \cdot g_j, i, j \in \{1, ..., n\}$ získat celou G

Definice: grupa **G** je *cyklická*, pokud má alespoň jeden generátor

6.1 Příklad

Grupu tvoří například množina všech nesoudělných čísel menších než nějaké $n \in \mathbb{N}$ společně s násobením v mod n aritmetice. Například pro $n = 5$ je $G = (1, 2, 3, 4)$ a tato grupa je navíc cyklická, neboť má generátor například

2 (ale i 3).

$$2^1 \equiv 2 \pmod 5$$
$$2^2 \equiv 4 \pmod 5$$
$$2^3 \equiv 3 \pmod 5$$
$$2^4 \equiv 1 \pmod 5$$

Dá se dokázat, že výše popsané grupy jsou cyklické, pokud $n = 1, 2, 4, p^k$ nebo $2p^k$, kde p je prvočíslo a $k \in \mathbb{N}$.

6.2 Protokol

Protokol sám o sobě není zase tak komplikovaný. Korespondenti se shodnou přes veřejný kanál na nějaké cyklické grupě a jejím generátoru. Pokud použijeme **Příklad**, domluví se na nějakém n takovém, aby grupa byla cyklická a vyberou nějaký její generátor g. Tyto informace mohou být veřejné. Potom si jeden náhodně zvolí číslo $a \in \mathbb{N}$, druhý $a \in \mathbb{N}$, obě menší než n. Potom si oba spočítají $g^a \pmod n$, respektive $g^b \pmod n$ a tato čísla si vymění. Potom obdržená čísla opět umocní na svoje náhodně zvolené číslo a protože $(g^a)^b \pmod n = (g^b)^a \pmod n$, dostanou oba stejné číslo a to mohou použít jako klíč k symetrickému šifrování. V principu může být samozřejmě použita jakákoliv cyklická grupa, v praxi se používají například grupy konstruované pomocí elitpických křivek .

6.3 Bezpečnost

Aby útočník zjistil klíč, musel by nějak zjistit a a b z čísel $g^a \pmod n$ a $g^b \pmod n$. Pokud by například bylo $n = 11, g = 7$ a korespondenti by si mezi sebou poslali 5 a 3, hledal by neznámé ve vztazích

$$7^a \pmod{11} \equiv 5 \pmod{11}$$
$$7^b \pmod{11} \equiv 3 \pmod{11}$$

Pro takto malá čísla snadno zjistí, že $a = 2$ a $b = 4$, ale pro velká n to už tak jednoduché není. Stále neexistují o moc lepší metody, než si vypisovat mocniny 7 v mod n a kontrolovat, kdy se budou rovnat 5 nebo 3. A to je ta operace, kterou jsem zmiňoval na začátku této sekce - umocňování v mod n. I pro velká čísla to počítače umí velmi rychle, ale hledání exponentu jim

jde mnohem pomaleji. Tento problém je znám jako *Problém diskrétního logaritmu*. V současnosti není znám žádný algoritmus fungující v polynomiálním čase (vzhledem k počtu cifer n), který by logaritmus spočítal.

7 Zdroje

1. D. Stanovský -*Základy algebry*, MatfyzPress 2010

2. RSA (cryptosystem) - Wikipedia. [online]. Dostupné z: https://en.wikipedia.org/wiki/RSA_(cryptosystem)

3. TAMU Computer Science People Pages [online]. Copyright ©, [cit. 09.09.2019]. Dostupné z: http://faculty.cs.tamu.edu/klappi/alg/rsa.pdf

4. Shorův algoritmus: https://arxiv.org/pdf/quant-ph/0010034.pdf

5. Shor's algorithm - Wikipedia. [online]. Dostupné z: https://en.wikipedia.org/wiki/Shor%27s_algorithm

6. (Almost) Unbreakable Crypto — Infinite Series - YouTube. YouTube [online]. Dostupné z: https://www.youtube.com/watch?v=NOs34_-eREk&list=PLa6IE8XPP_gmVt-Q4ldHi56mYsBuOg2Qw&index=1

7. Diffie–Hellman key exchange - Wikipedia. [online]. Dostupné z: https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

8. cryptography - "Diffie-Hellman Key Exchange"in plain English - Information Security Stack Exchange. Information Security Stack Exchange [online]. Dostupné z: https://security.stackexchange.com/questions/45963/diffie-hellman-key-exchange-in-plain-english

9. AES: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard#High-level_description_of_the_algorithm