Y36BEZ – Bezpečnost přenosu a zpracování dat

Róbert Lórencz

4. přednáška

Teorie informace, teorie složitosti algoritmů
Hašovací funkce

http://service.felk.cvut.cz/courses/Y36BEZ
lorencz@fel.cvut.cz

Obsah přednášky

- Teorie informace
- Teorie složitosti algoritmů
- Hašovací funkce

Teorie informace (1)

- V 1948 a 1949 C. E. Shannon zveřejněním svých prací Shannon_1948, Shannon_1949, položil základy teorie informace a teorie šifrovacích systémů.
- Stanovil teoretickou míru bezpečnosti šifry pomocí neurčitosti OT.
- Mějme ŠT $\Rightarrow$ jestliže se nic nového o OT nedozvíme (například zúžení prostoru možných zpráv), i když přijmeme jakékoliv množství ŠT $\Rightarrow$ šifra dosahuje absolutní bezpečnosti (perfect secrecy), tj. ŠT nenese žádnou informaci o OT.
- Zvyšováním počtu znaků ŠT je u většiny praktických (zvlášť u historických šifer) nabízeno více a více informace o OT.
- Tato informace nemusí být bezprostředně viditelná.
- Vzdálenost jednoznačnosti = # znaků OT, pro který množství informace o OT obažené v ŠT dosáhne takového bodu, že je možný jen jediný OT.

Teorie informace (2)

Příklad:

- Mějme jednoduchou substituci. Se znalosti jediného znaku "Z" ŠT nemáme ještě žádnou informaci o OT.
- Pravděpodobnost, že se pod ŠT skrývá písmeno A (resp. B, C, ..., Z) je stejná jako pravděpodobnost výskytu písmene A (B, ..., Z) v OT, tj. nedozvěděli jsme se o OT žádnou informaci.
- Zvýšením počtu písmen ŠT na 2, například "ZP", máme určitou informaci o OT, tj. OT nemohou být všechny možné bigramy, protože vylučujeme bigramy se stejnými písmeny. Ty by dávaly ŠT typu "ZZ". Při 50 písmenech už v ŠT můžeme lehce rozeznávat samohláskové a souhláskové pozice a můžeme je určovat.
- Při 1000 znacích ŠT je OT plně nebo až na detaily plně určen. Mezi číslem 1 a 1000 je tedy bod, kdy je OT už jen jeden.
- Z praxe víme, že při určování vzdálenosti jednoznačnosti bude také záležet na jazyku OT $\Rightarrow$ pojem entropie zdroje zpráv.

Teorie informace (3)

Definice – Entropie

Entropie je množství informace obsažené ve zprávě.
Teorie informace měří entropii zprávy průměrným počtem bitů, nezbytných k jejímu zakódování při optimálním kódování (minimum bitů).

Entropie zprávy ze zdroje X je

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i,$$

$p_1, \ldots, p_n$ jsou pravděpodobnosti všech zpráv $X_1, \ldots, X_n$ zdroje X a $-p_i \log_2 p_i$ = # bitů nutných k optimálnímu zakódování zprávy X_i.

V případě, že máme n zpráv se stejnou pravděpodobností $p = \frac{1}{n}$ (náhodné řetězce délky $\log_2 n$) $\Rightarrow$ entropie takového zdroje je $H(X) = -n(\frac{1}{n} \log_2(\frac{1}{n})) = \log_2 n$. Znamená to, že k binárnímu zakódování n zpráv se stejnou pravděpodobností nějakého zdroje potřebujeme přibližně tolik bitů, kolik obsahuje informací — entropie (podle definice).

Teorie informace (4)

Poznámka

Lze dokázat, že maximální entropie nabývá zdroj, který produkuje všechny zprávy se stejnou pravděpodobností. Má-li zdroj n zpráv se stejnou pravděpodobností $\frac{1}{n}$, pak dosahuje maximální entropie $\log_2 n$.

Doporučení

V případě velkých souborů dat lze získat orientační představu a horní odhad pro entropii těchto souborů použitím kvalitního komprimačního programu $\Rightarrow$ entropie souboru $\leq$ # bitů komprimovaného souboru.

Příklad: Mějme 2 možné zprávy: „panna“ nebo „orel“ a nechť mají stejnou pravděpodobnost $\Rightarrow$ entropie zprávy z tohoto zdroje je $H(X) = -0.5 \cdot (-1) - 0.5 \cdot (-1) = 1$, tedy jeden bit.

Teorie informace (5)

Příklad: Mějme zdroj, který vydává zprávy: „bílý“ s pravděpodobností $\frac{1}{4}$ a „černý“ s pravděpodobností $\frac{3}{4} \Rightarrow$ entropie zprávy z tohoto zdroje je $H(X) = 0.25 \cdot \log_2 4 + 0.75 \cdot \log_2 \frac{4}{3} = 0.25 \cdot 2 + 0.75 \cdot 0.41 = 0.81$, takže zprávy z tohoto zdroje obsahují v průměru necelý bit informace.

Příklad: Mějme zdroj vydávající pouze jednu zprávu s $p = 1 \Rightarrow$ entropie takového zdroje je $H(X) = -(1 \cdot 0) = 0$, (0 bitů). Daný zdroj nemá žádnou neurčitost a zprávy z něj nenesou žádnou informaci.

Neurčitost

Entropie zprávy vyjadřuje také míru její neurčitosti. Tato míra neurčitosti je počet bitů, které potřebujeme získat luštěním ŠT, s cílem určit OT.

Například v případě, že část ŠT „4lů9Fg“ vyjadřuje buď výsledek „panna“ nebo „orel“, pak míra neurčitosti této zprávy je rovná právě 1.

Teorie informace (6)

Definice – Obsažnost jazyka – Průměrná entropie

Pro daný jazyk uvažujme množinu X všech N-znakových zpráv. Obsažnost jazyka pro zprávy délky N znaků definujeme jako výraz

$$R_N = \frac{H(X)}{N},$$

tj. průměrnou entropii na 1 znak (průměrný # bitů informace v 1 znaku).

- Máme-li dlouhou zprávu, její další písmeno bývá v řadě případů určeno už jednoznačně nebo je možný jen malý počet variant.
- Například máme-li 13 znakovou zprávu „Zítra odpoled“ , je pravděpodobné, že pokračuje písmenem „n“. U 14. znaku nepřibude žádná entropie.
- V případě ostatních možných 13-znakových řetězců bude situace podobná, tj. umožní jen 1 nebo několik málo variant.
- Entropie příliš nevzroste $\Rightarrow R_{N+1} < R_N$.

Teorie informace (7)

- U přirozených jazyků výraz R_N pro zvyšující se N klesá.
- Z předchozího plyne: $\lim_{N\to\infty} R_N = r$
- Konstanta r je obsažnost jazyka vzhledem k jednomu písmenu.
- Pro hovorovou angličtinu je $r = 1.3$ až 1.5 bitů/znak.
- Obsažnost jazyka nám tedy říká, kolik bitů informace ve skutečnosti obsahuje průměrně 1 písmeno jazyka.

Definice – Absolutní obsažnost jazyka R

Mějme stejně pravděpodobné zprávy tvořené v jazyce s L stejně pravděpodobnými znaky $\Rightarrow R_N = \frac{\log_2 L^N}{N} = \log_2 L = R$. R nazýváme absolutní obsažnost jazyka. Absolutní obsažnost dosahuje takový jazyk, který poskytuje generátor náhodných znaků. Je to maximální neurčitost, kterou přirozené jazyky nemohou dosáhnout, neboť jednotlivé znaky tvoří slova a věty, které mají odlišné pravděpodobnosti.

Teorie informace (8)

Definice – Nadbytečnost jazyka vzhledem k jednomu písmenu D

Nadbytečnost (redundance) jazyka vzhledem k jednomu písmenu nám vyjádřuje kolik bitů je v jednom znaku daného jazyka nadbytečných a je dána výrazem

$$D = R - r \qquad \text{a číslo} \qquad \frac{100D}{R}$$

pak udává, kolik bitů jazyka je nadbytečných procentuálně.

Příklad: Pro angličtinu máme $L = 26$, $R = \log_2 26 = 4.7$ bitů na písmeno, $r = 1.5$ bitů/písmeno, $D = R - r = 4.7 - 1.5 = 3.2$ nadbytečných bitů/písmeno a procentuálně to je

$$\frac{100D}{R} = \frac{100 \cdot 3.2}{4.7} = 68\ \%$$ nadbytečných bitů/písmeno.

Příklad: Anglický text v kódu ASCII má v každém bytu (7b informací a 1b parity) 1.5b informace anglického znaku. 8b informace může nést také 8b informace $\Rightarrow D = R - r = 8 - 1.5 = 6.5$b redundance.

Teorie informace (9)

Výpočet vzdálenosti jednoznačnosti (1)

Uveďme si pro ilustraci následující úvahu (neplatí pro všechny šifry).

- Mějme množinu zpráv M, množinu ŠTů C a množinu $2^{H(K)}$ stejně pravděpodobných klíčů K.
- Předpokládejme, že máme ŠT c délky N znaků a že pro klíče $k \in K$ jsou odpovídající OT $D_k(c)$ vybírány z množiny všech zpráv M nezávisle a náhodně.
- V množině M je celkem 2^{RN} zpráv a z toho je 2^{rN} smysluplných zpráv a $U = 2^{RN} - 2^{rN}$ zpráv nesmysluplných.
- Pokud provedeme dešifrování ŠT c všemi možnými $2^{H(K)}$ klíči, dostáváme $2^{H(K)}$ zpráv. Z nich je smysluplných průměrně pouze $S = 2^{H(K)} \frac{2^{rN}}{2^{RN}} = \frac{2^{H(K)}}{2^{DN}} = 2^{H(K)-DN}$.
- Abychom dostali pouze jednu zprávu - tu, která byla skutečně zašifrována - musí být $S = 1$.

Teorie informace (10)

Výpočet vzdálenosti jednoznačnosti (2)

- Z předchozího plyne: $H(K) = DN$ a $N = \frac{H(K)}{D} = \delta_U$.

Definice — Vzdálenost jednoznačnosti

Vzdálenost jednoznačnosti je definována jako

$$\delta_U = \frac{H(K)}{D},$$

kde $H(K)$ je neurčitost klíče a D je redundance jazyka otevřené zprávy.

Příklad: Mějme obecnou jednoduchou substituci nad anglickou abecedou. Její vzdálenost jednoznačnosti je $\delta_U = \frac{H(K)}{D} = \frac{\log_2(26!)}{3.2} = \frac{88.3}{3.2} = 27.6$. V ŠT o 28 znacích je tedy dostatečné množství informace na to, aby zbýval v průměru jediný možný OT. K rozluštění jednoduché substituce v angličtině postačí tedy v průměru 28 písmen ŠT.

Teorie informace (10)

Výpočet vzdálenosti jednoznačnosti (3)

Příklad: Mějme klíč Vigenerovy šifry o délce V náhodných znaků a uvažujme otevřený text z anglické abecedy. Vzdálenost jednoznačnosti je $\delta_U = \frac{H(K)}{D} = \frac{\log_2(26^V)}{3.2} = \frac{V \log_2(26)}{3.2} = \frac{4.7V}{3.2} = 1.5V$.

To je na první pohled optimistický výsledek, ale ...

- Mějme ŠT v délce 1.5 násobku hesla, tj. oněch $1.5V$ znaků. 1. a 3. třetina textu používá stejné heslo, které lze eliminovat odečtením a poté s určitou pravděpodobností vyluštit 1. a 3. třetinu OT.
- Zůstává neznámá 2. třetina OT, kde je heslo zcela náhodné a neznámé, nemáme tedy z něj žádnou informaci. Zbývá redundance OT, z něhož známe 1. a 3. třetinu.
- δ_U je střední hodnota vzdálenosti jednoznačnosti a nezohledňuje právě takové rozložení informace z OT, které je k dispozici.
- Z hlediska množství informace pokud máme dvě třetiny OT, zbytek bychom měli být schopni u anglického jazyka doplnit.

Teorie informace (10)

Výpočet vzdálenosti jednoznačnosti (4)

- Problém: na krátkých úsecích nedosahuje D hodnoty 3.2 bitu, ale méně, a navíc rozložení informace o OT je v daném případě atypické (známe 1. a 3. třetinu textu) $\Rightarrow$ nelze na něj vztahovat výsledky, týkající se průměru a průměrných — obvyklých textů.
- Na druhou stranu, pokud bychom měli k dispozici $2V$ znaků, budeme už schopni heslo eliminovat a luštit metodou knižní šifry. Kdybychom měli o pár znaků méně, byli bychom ještě schopni tyto znaky doplnit právě z důvodu nadbytečnosti zprávy.
- Skutečnou vzdálenost jednoznačnosti lze proto v závislosti na konkrétním problému a daném V očekávat v rozmezí $1.5V$ až $2V$.

Upozornění

Vzdálenost jednoznačnosti je odhad množství informace, nutného k vyluštění dané úlohy. Neříká však nic o složitosti takové úlohy.

Teorie informace (11)

Konfúze a difúze – podle Shannona základní techniky k potlačení redundance D ve zprávě. Konfúze – maří vztahy mezi ŠT a OT.

- Ztěžuje studium redundancí a statistických struktur OT.
- Nejjednoduší — substituce (Caesarova šifra).
- Proudové šifry — konfúze, pokud také zpětné vazby $\Rightarrow$ i difúze.
- Moderní substituční šifry jsou mnohem složitější, nahrazují se dlouhé bloky OT blokem ŠT a mechanizmus substituce se mění s bity klíče nebo OT.

Difúze – rozprostírá redundanci OT.

- Vyhledávaní rozprostřených redundancí ztěžuje kryptoanalýzu.
- Nejjednodušší — transpozice.
- V současnosti jsou tyto způsoby kombinovány k dokonalejšímu rozptylu redundance.
- Obecně — difúzi lze snadno luštit, ale už ne například po dvojnásobné transpozice.

Teorie složitosti (1)

Teorie složitosti

- Metodologický základ pro analýzu výpočetní složitosti kryptografických technik a algoritmů.
- Porovnává kryptografické techniky a algoritmy a určuje jejich bezpečnost.
- Teorie informace $\rightarrow$ za jakých podmínek znalosti (jazykového) kryptografického prostředí lze kryptografický algoritmus rozluštit.
- Teorie složitosti $\rightarrow$ s jakou složitostí bude možné kryptografický algoritmus rozluštit.

Výpočetní složitost algoritmu je dána výpočetním výkonem potřebným pro jeho realizaci a zahrnuje:

- $T(n)$ – časovou náročnost.
- $S(n)$ – prostorovou náročnost (paměťové požadavky).

Proměnná n je rozsah vstupu.

Teorie složitosti (2)

Výpočetní složitost se vyjadřuje řádem O (Order) hodnoty výpočetní složitosti.

- Řád složitosti O roste nejrychleji v závislosti na n.
- Všechny prvky nižšího řádu se zanedbávají.
- Pro výpočetní náročnost algoritmu $7n^4 + 5n^2 + 6$ je $O(n^4)$.
- Takové vyjadřování složitosti algoritmu je systémově nezávislé.
- Umožňuje vyhodnotit vliv velikosti vstupu na časové a prostorové požadavky.
- Když $T(n) = O(n) \Rightarrow$ pro dvakrát větší vstup je dvakrát větší časová náročnost.
- V případě $T(n) = O(2^n)$ zvětšením vstupu o jednu se prodlouží doba výpočtu dvojnásobně.
- Časová složitost $T(n) = O(1)$ je nezávislá na n (na vstupech).

Teorie složitosti (3)

Klasifikace algoritmů podle časové nebo prostorové náročnosti

- Konstatní složitost, nezávislá na $n \Rightarrow O(1)$.
- Lineární složitost $\Rightarrow O(n)$.
- Polynomiální složitost $\Rightarrow O(n^m)$, kde m je konstantní. Například kvadratická složitost má $O(n^2)$, kubická složitost má $O(n^3)$, atd.
- Exponenciální složitost $\Rightarrow O(t^{f(n)})$, kde $t > 1$ je konstanta a $f(n)$ je nějaká polynomiální funkce (také lineární).
- Podmnožinou exponenciálních složitostí je superpolynomiální složitost, pro kterou je řád složitosti $O(t^{f(n)})$, kde t je konstanta a $f(n)$ je více něz konstanta, ale méně než lineární funkce.

Poznámka

Známé luštitelské algoritmy jsou časově superpolynomiálně složité. Nedá se ale zatím dokázat, že nebude nikdy možné objevit nějaký časově polynomiální luštitelský algoritmus.

Teorie složitosti (4)

Tabulka ukazuje doby zpracování algoritmů s různou časovou složitostí

$T(n)$	# oper. ($n = 10^6$)	Čas výpočtu (1 oper. = 0.1μs)
$O(1)$	1	0.1μs
$O(n)$	10^6	100ms
$O(n^2)$	10^{12}	1.2 D
$O(n^3)$	10^{18}	3200 Y
$O(2^n)$	10^{301030}	$10^{301005}\times$ věk vesmíru

- S růstem n může časová složitost výrazně ovlivnit praktickou použitelnost algoritmu.
- Při útoku hrubou silou je časová náročnost luštění úměrná množství klíčů, které je exponenciální funkcí délky klíče.
- Pro n-bitový klíč je časová složitost luštění hrubou silou $O(2^n)$.
- Nechť klíč je 56 bitový $\Rightarrow O(2^{56}) = 7.2 \cdot 10^{16}$ a čas výpočtu je $\approx$ 81000 dnů (1 oper. = 0.1μs). Uvažujme 1000 $\times$ μPC s tímto výkonem $\Rightarrow$ nám stačí pro luštění čas kratší než 3 měsíce.

Teorie složitosti (5)

Složitost problémů

- Teorie složitosti klasifikuje kromě složitosti jednotlivých algoritmů použitých k řešení problému také jeho vnitřní složitost.
- Turingův stroj (TS) – konečný automat s nekonečnou čtecí – zapisovací páskovou pamětí.
- TS realistický modeluje výpočetní operace.

Klasifikace problémů podle složitosti

- Snadno řešitelné problémy – dají se řešit časově polynomiálními algoritmy za „přijatelně“ dlouhou dobu pro „rozumné“ velkosti n.
- Obtížne problémý – nelze řešit v polynomiálním čase za „přijatelně“ dlouhou dobu. Také jsou označováný jako těžké problémy. Problémy rešitelné pouze supernomiálnimi algoritmy jsou obtéžně řešitelné i při malých hodnotach n.
- Nerozhodnutelné problémy – pro jejích řešení nelze navrhnout žádný algoritmus, i vpřípadě, že neuvažujeme časovou složitost.

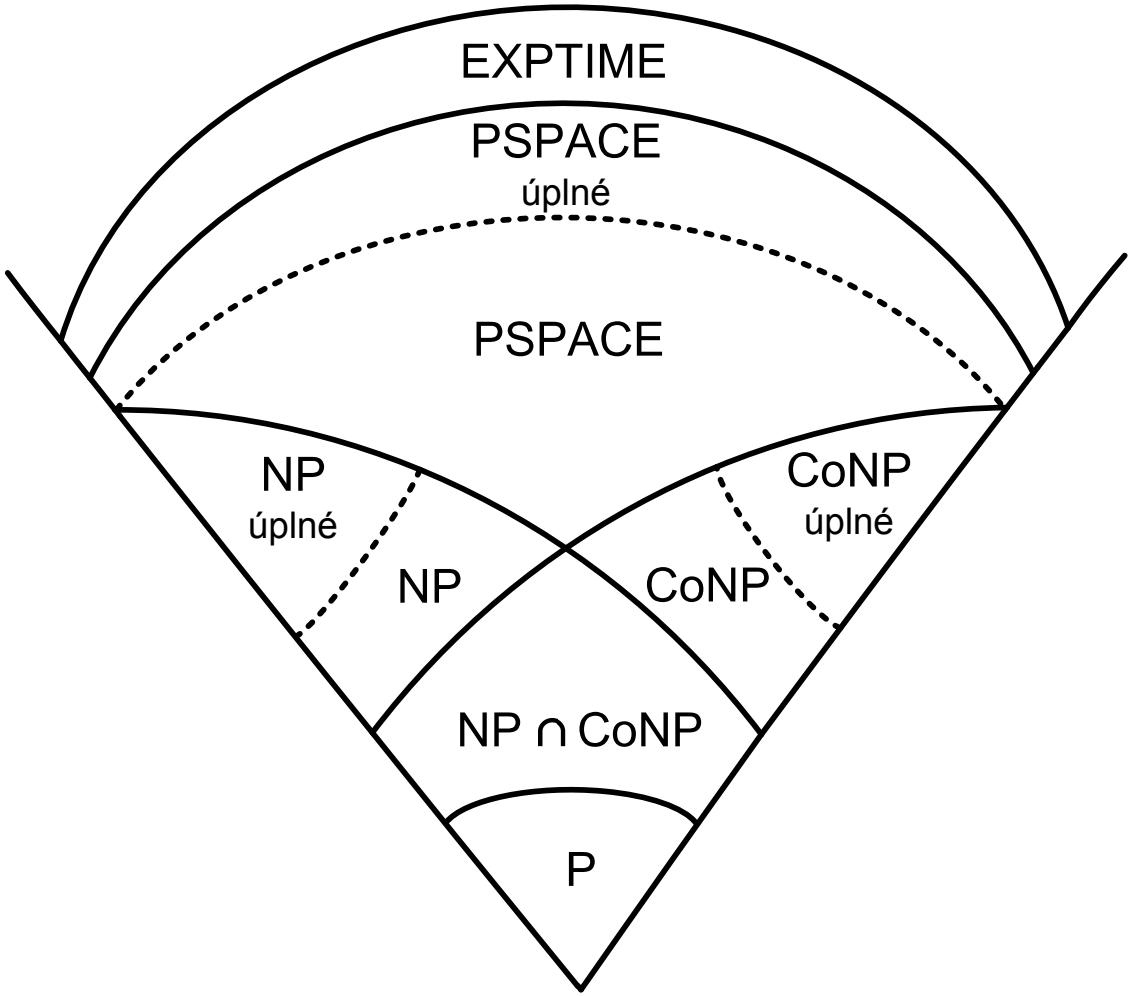
Teorie složitosti (6)

Dělení problémů podle tříd složitosti

- Třída P — Problémy mohou být řešeny v polynomiálním čase.
- Třída NP — Problémy mohou být řešeny v polynomiálním čase, ale pouze nedeterministickým TS, tj. může provádět pouze odhad buď správného řešení, nebo paralelně provede všechny odhady (výsledky prověřuje v polynomiálním čase).
- Třída NP-úplný — Problém, na který lze převést polynomiálním převodem každý problém ve třídě NP (NP těžký) a náleží do třídy NP.
- Třída PSPACE — Problémy mohou být řešeny v polynomiálním prostoru, ale nikoliv nezbytně v polynomiálním čase.
- Třída EXPTIME — Problémy, které jsou řešitelné v exponenciálním čase.
- Třída CoNP — Zahrnuje problémy, které jsou doplňkem některých problémů NP.

Teorie složitosti (7)

Třídy složitosti a jejich předpokládané vztahy



Teorie složitosti (8)

Vztah kryptologie a tříd složitosti

- Mnohé symetrické algoritmy a všechny algoritmy veřejného klíče se dají vyluštit v nedeterministickém polynomiálním čase.
- Při zadaném ŠT kryptoanalytik odhaduje OT a klíč a v polynomiálním čase nechá zpracovávat šifrovacím algoritmem tento odhad a prověřuje případnou shodu se ŠT.
- Tento postup je důležitý z teoretického hlediska, protože vymezuje horní mez složitosti kryptoanalýzy algoritmu.
- V praxi kryptoanalytik hledá deterministický časově polynomiální algoritmus.
- Pokud by se dokázalo, že P = NP pak ???
- Hodně šifer lze triviálně luštit v nedeterministikém polynomiálním čase, pokud P = NP, pak tyto šifry budou luštitelné realizovatelnými deterministickými algoritmy.

Hašovací funkce (1)

- Silný nástroj moderní kryptologie.
- Jedna z klíčových kryptologických myšlenek počítačové revoluce. Přinesly řadu nových použití.
- Základní pojmy: jednosměrnost a bezkoliznost.

Definice – Jednosměrné funkce

Funkce $f : X \rightarrow Y$, pro něž je snadné z jakékoli hodnoty $x \in X$ vypočítat $y = f(x)$, ale pro nějaký náhodně vybraný obraz $y \in f(X)$ nelze (je to pro nás výpočetně nemožné) najít její vzor $x \in X$ tak, aby $y = f(x)$. Přitom víme, že minimálně jeden takový vzor existuje.

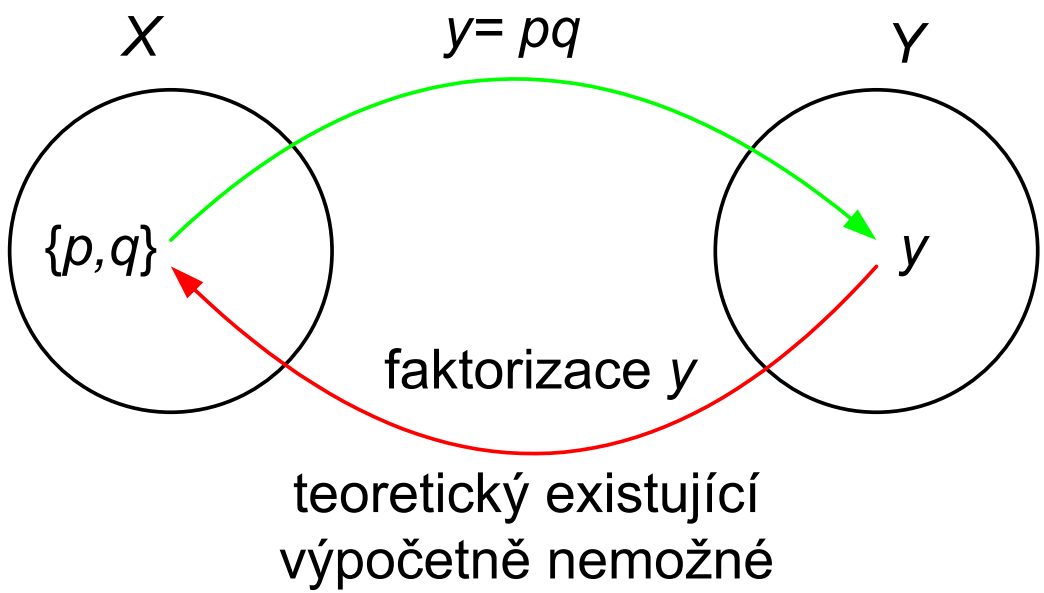
Jednosměrné funkce delíme na:

1. Jednosněrné, pro které je výpořetně nemožné, ale teoretický existující, najít vzor z obrazu.
2. Jenosměrné funkce s padacími vrátky, u kterých lze najít vzor z obrazu, ale jen za předpokladu znalosti „padacích vrátek“ – klíče.

Hašovací funkce (2)

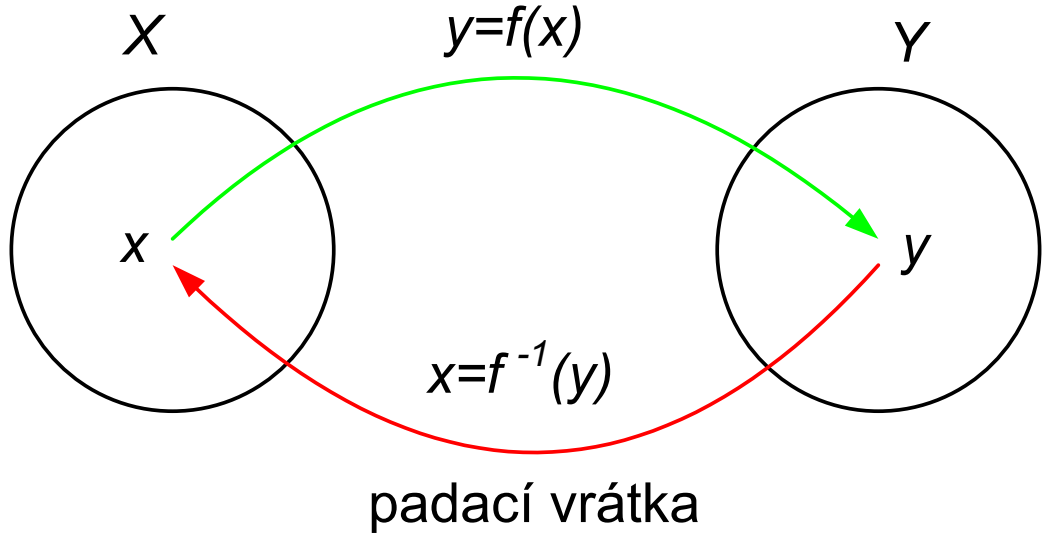
Jednosměrné funkce 1. typu

Jednosměrnost je dána násobením versus faktorizace dvou velkých prvočísel p a q.



Jednosměrné funkce 2. typu

Jednosměrnost je dána znalostí „padacích vrátek“, například u asymetrické kryptografie je to klíč.



Hašovací funkce (3)

Kryptografická hašovací funkce

- Původní význam hašovací funkce byl proto označením funkce, která libovolně velkému vstupu přiřazovala krátký hašovací kód o pevně definované délce.
- V současnosti se pojem hašovací funkce používá v kryptografii pro kryptografickou hašovací funkci, která má oproti původní definici ještě navíc vlastnost jednosměrnosti a bezkoliznosti.

Definice – Hašovací funkce

Mějme přirozená čísla d a množinu X všech binárních řetězců délky 0 až d (prázdný řetězec je platným vstupem a má délku 0).
Funkci $h : X \rightarrow \{0,1\}^n$ nazveme hašovací, jestliže je jednosměrná 1. typu a bezkolizní.
Říkáme, že každému binárnímu řetězci z množiny X přiřadí binární hašovací kód (haš, hash) délky n bitů.