

# MB141 – 12. přednáška

## Aplikace teorie čísel

Martin Čadek  
s využitím přednášek pro předmět MB104

Jarní semestr 2021

- Výpočetní aspekty teorie čísel
- Kryptografie s veřejným klíčem

V mnoha praktických úlohách využívajících výsledky teorie čísel je zapotřebí umět rychle provést jeden či více z následujících výpočtů:

- 1 běžné aritmetické operace (součet, součin, dělení se zbytkem) na celých číslech,
- 2 zbytek mocniny celého čísla  $a$  na přirozené číslo  $n$  po dělení daným  $m$ .
- 3 inverzi celého čísla  $a$  modulo  $m \in \mathbb{N}$ ,
- 4 největší společný dělitel dvou celých čísel (a případně koeficienty do Bezoutovy rovnosti),
- 5 rozhodnout o daném čísle, je-li prvočíslo nebo složené,
- 6 v případě složenosti rozložit dané číslo na součin prvočísel.

Základní aritmetické operace se i na velkých číslech obvykle provádějí obdobně jako jsme se to učili na základní a střední škole, kdy umíme **sčítat** v *lineárním* čase, **násobit a dělit se zbytkem** v *kvadratickém* čase. Pro násobení, které je základem mnoha dalších operací, existují asymptoticky rychlejší algoritmy (typu *rozděl a panuj*) - např. první takový Karatsubův (1960) časové náročnosti  $\Theta(n^{\log_2 3})$  nebo algoritmus Schönhage-Strassenův (1971) časové náročnosti  $\Theta(n \log n \log \log n)$ . Číslo  $n$  zde udává celkový počet cifer vstupujících do výpočtu. Pěkný přehled najdete např. na

[http://en.wikipedia.org/wiki/Computational\\_complexity\\_of\\_mathematical\\_operations](http://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations)

Jak už jsme ukazovali dříve, výpočet výpočet **inverze modulo  $m$** , tj. řešení kongruence  $a \cdot x \equiv 1 \pmod{m}$  s neznámou  $x$  lze snadno (díky Bezoutově větě) převést na výpočet největšího společného dělitele čísel  $a$  a  $m$  a na hledání koeficientů  $k, l$  do Bezoutovy rovnosti  $k \cdot a + l \cdot m = 1$  (nalezené  $k$  je pak onou hledanou inverzí  $a$  modulo  $m$ ).

Podrobná analýza ukazuje, že tento algoritmus je **kvadratické** časové složitosti.

# Modulární umocňování

V kryptografii s veřejným klíčem budeme budeme potřebovat **umocňování modulo  $m$** . To se také využívá při ověřování, zda je dané číslo prvočíslo nebo číslo složené. Jedním z efektivních algoritmů je tzv. **modulární umocňování zprava doleva**:

```
function modular_pow(base, exponent, modul)
    result := 1
    while exponent > 0
        if (exponent mod 2 == 1):
            result := (result * base) mod modul
        exponent := exponent >> 1
        base = (base * base) mod modul
    return result
```

Symbol `>>` ve třetím řádku zdola znamená, že od exponentu zapsaného v dvojkové soustavě odebereme poslední cifru 1.

Algoritmus modulárního umocňování je založen na myšlence, že např. při počítání  $2^{64} \pmod{1000}$

- není třeba nejprve počítat  $2^{64}$  a poté jej vydělit se zbytkem číslem 1000, ale lépe je postupně násobit „dvojky“ a kdykoliv je výsledek větší než 1000, provést redukci modulo 1000,
- ale zejména, že není třeba provádět takové množství násobení (v tomto případě 63 naivních násobení je možné nahradit pouze šesti umocněními na druhou, neboť

$$2^{64} = ((((((2^2)^2)^2)^2)^2)^2)^2.$$

# Ukázka průběhu algoritmu

Vypočtěme  $2^{560} \pmod{561}$ . Protože  $560 = (1000110000)_2$ , dostaneme uvedeným algoritmem

exponent	base	result	exp's last digit
560	2	1	0
280	4	1	0
140	16	1	0
70	256	1	0
35	460	1	1
17	103	460	1
8	511	256	0
4	256	256	0
2	460	256	0
1	103	256	1
0	511	1	0

A tedy  $2^{560} \equiv 1 \pmod{561}$ .



V průběhu algoritmu se pro každou binární číslici exponentu provede umocnění základu na druhou modulo  $m$  (což je operace proveditelná v nejhůře kvadratickém čase), a pro každou „jedničku“ v binárním zápisu navíc provede jedno násobení. Celkově jsme tedy schopni provést modulární umocňování nejhůře v **kubickém** čase.

Další úlohy výpočetní teorie čísel jsou **testování prvočíselnosti** a **rozklad složených čísel na prvočísla**. Tato témata jsou na samostatnou přednášku, nebudeme se jimi zde zabývat. Více lze najít v učebnici Drsná matematika v odstavcích 10.38-47.

Dva hlavní úkoly pro „public-key cryptography“ jsou zajistit

- šifrování, kdy zprávu **zašifrovanou veřejným klíčem** není schopen rozšifrovat nikdo kromě držitele soukromého klíče,
- podepisování, kdy integrita zprávy **podepsané soukromým klíčem** odesílatele může být ověřena kýmkoliv s přístupem k veřejnému klíči odesílatele.

Nejčastěji používané systémy:

- RSA (šifrování) a odvozený systém pro podepisování zpráv
- Digital signature algorithm (DSA) a varianta založená na eliptických křivkách (ECDSA)
- Rabinův kryptosystém (a podepisování)
- Diffie-Hellmanův protokol na výměnu klíčů (DH)
- ElGamal kryptosystém (a podepisování)
- Kryptografie eliptických křivek (ECC)

*Rivest, Shamir, Adleman (1977); Cocks(1973)*

- Jsou dva typy klíčů – veřejný a soukromý.
- Generování klíčů: zvolí se dvě velká prvočísla  $p, q$ , vypočte se  $n = pq$ ,  $\varphi(n) = (p - 1)(q - 1)$ . Přitom pouze  $n$  je veřejné,  $\varphi(n)$  nelze snadno spočítat.
- **Veřejný klíč** je číslo  $e$  s vlastností  $(e, \varphi(n)) = 1$
- **Soukromý klíč** je číslo  $d$  s vlastností  $e \cdot d \equiv 1 \pmod{\varphi(n)}$ . Ze znalosti  $\varphi(n)$  ho spočítáme např. pomocí Eukleidova algoritmu.
- Zpráva je číslo  $M \pmod{n}$ . Zašifrujeme ji jako  $C \equiv M^e \pmod{n}$ .
- Dešifrování šifry  $C$  spočívá ve výpočtu:  $C^d \pmod{n}$ .
- Na základě Eulerovy věty totiž  $\pmod{n}$  dostaneme  $C^d \equiv (M^e)^d \equiv M^{ed} \equiv M^{k\varphi(n)+1} \equiv (M^{\varphi(n)})^k \cdot M \equiv M$ .

Dalším veřejným kryptosystémem je **Rabinův kryptosystém**, který si uvedeme ve zjednodušené verzi:

- Každý účastník  $A$  potřebuje dvojici klíčů – veřejný  $V_A$  a soukromý  $S_A$ .
- Generování klíčů:  $A$  zvolí dvě podobně velká prvočísla  $p, q \equiv 3 \pmod{4}$ , vypočte  $n = pq$ .
- $V_A = n, S_A = (p, q)$  - dvojice prvočísel, nikoliv jejich největší společný dělitel, který je 1.
- Zašifrování numerického kódu zprávy  $M$ :  
 $C \equiv M^2 \pmod{n}$ .
- Dešifrování šifry  $C$ : vypočtou se (čtyři) odmocniny z  $C$  modulo  $n$  a snadno se otestuje, která z nich byla původní zprávou.

Výpočet druhé odmocniny z  $C$  modulo  $n = pq$ ,  
kde  $p \equiv q \equiv 3 \pmod{4}$  probíhá takto:

- vypočti  $r = C^{(p+1)/4} \pmod{p}$  a  $s = C^{(q+1)/4} \pmod{q}$
- vypočti  $a, b$  tak, že  $ap + bq = 1$
- polož  $x = (aps + bqr) \pmod{n}$ ,  $y = (aps - bqr) \pmod{n}$
- druhými odmocninami z  $C$  modulo  $n$  jsou  $\pm x, \pm y$ .

**Zdůvodnění:** Z Čínské zbytkové věty vyplývá, že  $z^2 \equiv C \pmod{pq}$ , právě když současně platí  $z^2 \equiv C \pmod{p}$  a  $z^2 \equiv C \pmod{q}$ . Lze ukázat, že pro každé liché prvočíslo platí, že kvadratická kongruence  $z^2 \equiv C \pmod{p}$  má řešení právě když  $C^{\frac{p-1}{2}} \equiv 1 \pmod{p}$ . Tedy při počítání  $\pmod{p}$  dostáváme  $x^2 \equiv y^2 \equiv (bq)^2 r^2 \equiv 1^2 \cdot C^{\frac{p+1}{2}} \equiv C^{\frac{p-1}{2}} \cdot C \equiv 1 \cdot C \equiv C \pmod{p}$ . Analogický výpočet lze provést  $\pmod{q}$ .

## Příklad

V Rabinově kryptosystému Alice zvolila za svůj soukromý klíč  $p = 23$ ,  $q = 31$ , veřejným klíčem je pak  $n = pq = 713$ .  
Zašifrujte zprávu  $M = 327$  pro Alici a ukažte, jak bude Alice tuto zprávu dešifrovat.

$C \equiv 327^2 \equiv 692$ . Při dešifrování spočítáme  $C^6 \equiv (2)^6 \equiv 18 \pmod{23}$  a  $C^8 \equiv 10^8 \equiv 14 \pmod{31}$ . Dále  $-4 \cdot 23 + 3 \cdot 31 = 1$ . Proto máme 4 kandidáty na zprávu, a to  $\pm 4 \cdot 23 \cdot 14 \pm 3 \cdot 31 \cdot 18 \pmod{713}$ . To jsou čísla  $\pm 38$  a  $\pm 327 \pmod{713}$ .

## Podepisování

- 1 Vygeneruje se otisk (hash)  $H_M$  zprávy pevně stanovené délky (např. 160 nebo 256 bitů).
- 2 Podpis zprávy  $S_A(H_M)$  je vytvořen (pomocí dešifrování) z tohoto hashe s nutností znalosti soukromého klíče  $S_A$  podepisujícího.
- 3 Zpráva  $M$  (případně zašifrovaná veřejným klíčem příjemce) je spolu s podpisem odeslána.

## Ověření podpisu

- 1 K přijaté zprávě  $M$  se (po jejím případném dešifrování) vygeneruje otisk  $H'_M$
- 2 S pomocí veřejného klíče  $V_A$  (deklarovaného) odesílatele zprávy se rekonstruuje původní otisk zprávy  $V_A(S_A(H_M)) = H_M$ .
- 3 Oba otisky se porovnají  $H_M = H'_M$ ?

# Výměna klíčů podle Diffie-Hellmana

*Diffie, Hellman (1976), Williamson (1974)*

Výměna klíčů pro symetrickou kryptografii bez předchozího kontaktu (tj. náhrada jednorázových klíčů, kurýrů s kufříky, ...).

- Dohoda stran na **prvočísle**  $p$  a **primitivním kořenu**  $g$  modulo  $p$  (veřejné). (Zopakujme, že  $g$  je primitivní kořen modulo prvočíslo  $p$ , jestliže  $g^n \equiv 1 \pmod{p}$  pouze pro násobky  $p - 1$ .)
- Alice vybere náhodné číslo  $a$  a pošle Bobovi  $g^a \pmod{p}$
- Bob vybere náhodné  $b$  a pošle Alici  $g^b \pmod{p}$
- Společným klíčem pro komunikaci je  $g^{ab} \pmod{p}$ .



Z protokolu Diffie–Hellman na výměnu klíčů je odvozen šifrovací algoritmus ElGamal:

- Alice zvolí prvočíslo  $p$  spolu s primitivním kořenem  $g$ .
- Alice zvolí **tajný klíč**  $a$ , spočítá  $h = g^a \pmod{p}$  a zveřejní **veřejný klíč**  $(p, g, h)$ .
- Šifrování zprávy  $M$ : Bob zvolí náhodné  $b$  a vypočte  $C_1 = g^b \pmod{p}$  a  $C_2 = M \cdot h^b \pmod{p}$  a pošle Alici  $(C_1, C_2)$ .
- Dešifrování zprávy provede Alice tak, že spočítá inverzi  $l$  k  $C_1^a = (g^b)^a = g^{ab} \pmod{p}$  a vynásobí  $C_2 \cdot l \equiv M \cdot g^{ab} \cdot l \equiv M \pmod{p}$ .

Příklad najdete ve cvičení. Analogicky jako pro RSA lze odvodit podepisování.