1 ZÁKLADY MODULÁRNEJ ARITMETIKY, TEÓRIA ČÍSEL

1.1 ÚVOD

Moderná kryptografia je založená predovšetkým na využití matematického aparátu, ktorý patrí do **teórie čísel**, t.j. tej časti matematiky, ktorej základným objektom štúdia sú vlastnosti prirodzených a celých čísel. Pod prirodzenými číslami rozumieme prvky množiny čísel

$$\mathbb{N} = \{1, 2, 3, \ldots\} \tag{1.1}$$

a pod celými číslami prvky množiny

$$\mathbb{Z} = \{0, \pm 1, \pm 2, \pm 3, \ldots\} \tag{1.2}$$

Vzhľadom na objekt skúmania, t.j. množiny prirodzených a celých čísel s ktorými sa človek stretával už v najstarších dobách, je možné teóriu čísel zaradiť k jednej z najstarších matematických disciplín. Už v období antiky boli známe pojmy a vedomosti, ktoré sa v súčasnosti vyučujú na základných a stredných školách. Niektoré algoritmy objavené v tomto období patria dokonca stále k efefektívnym výpočtovým algoritmom, ktoré sa využívajú aj v súčasných moderných kryptografických a kódovacích algoritmoch[1]. Prácami L. Eulera (1707-1783) a predovšetkým zavedením pojmu **kongruencie** C.F. Gaussom (1777-1855) boli položené základy modernej teórie čísel a príbuznej matematickej disciplíny – **algebry**, ktorú je možné v rozšírenom zmysle charakterizovať ako tú časť matematiky, ktorá vyšetruje vlastnosti množín a ich prvkov z hľadiska algebraickej manipulácie s nimi[2].

V ďalšej časti budú uvedené základné informácie z oblasti teórie čísel, ktoré sú využívané v oblasti aplikovanej kryptografie. Aj keď matematický aparát vo všeobecnosti patrí do oblasti menej populárnych disciplín, je vhodné si uvedomiť, že úspešné zvládnutie tohoto aparátu umožňuje pochopenie a využívanie prakticky využívaných algoritmov. Typickým (ale zďaleka nie jediným!) príkladom je *generovanie prvočísel*, ktoré sú základným stavebným blokom moderných kryptografických algoritmov a protokolov (ako napr. **digitálne podpisy**, **šifrovanie s verejným kľúčom** a pod.).

[1] Typickým príkladom je Euklidov algoritmus na nájdenie najväčšieho spoločného deliteľa dvoch čísel (prípadne aj polynómov), ktorý sa okrem iného využíva aj v jednej z metód dekódovania Reedových – Solomonových zabezpečovacích blokových kódov.

[2] Typickými predstaviteľmi takýchto manipulácií sú napr. operácie sčítania a odčítania čísel.

1.2 MODULÁRNA ARITMETIKA

Typickým príkladom modulárnej aritmetiky, s ktorou sa stretávame v bežnom živote je „hodinová aritmetika“. Ak povieme, že sa stretneme o 11 hodine, môže to znamenať aj 23 hodinu. V prípade, že používame len 12 hodinový interval, sú údaje 11 a 23 ekvivalentné[3], čo budeme zapisovať v tvare

$$23 \equiv 11 \pmod{12}$$

Obecne platí $a \equiv b \pmod{n}$ vtedy, ak $a = b + kn$ pre nejaké celé číslo k (t.j. rozdiel $a-b$ je deliteľný číslom n). Ak bude a nezáporné číslo a b číslo medzi 0 až $n-1$, potom je možné b chápať ako zvyšok po delení čísla a číslom n. Niekedy číslu b hovoríme **reziduo** a, modulo n a o čísle a hovoríme, že je **kongruentné** (zhodné) s číslom b, modulo n (symbol $\equiv$ označuje kongruenciu, číslo n sa nazýva **modul kongruencie**). V opačnom prípade hovoríme, že čísla a, b sú nekongruentné modulo n.

Množina celých čísel od 0 do $n-1$ tvorí to, čo je označované ako **úplná množina reziduí modulo** n. To znamená, že reziduom modulo n akéhokoľvek celého čísla je nejaké číslo medzi 0 a $n-1$.

Operácia $a \bmod n$ určuje reziduo a, ktoré leží v intervale 0 až $n-1$. Táto operácia je známa pod označením **modulárna redukcia**.

Modulárna aritmetika je v mnohých rysoch podobná normálnej aritmetike. Je komutatívna, asociatívna a distributívna. Taktiež modulárnou redukciou každého medzivýsledku modulom n dospejeme k rovnakému výsledku, ku ktorému by sme sa dostali pokiaľ by sme vykonali výpočet v normálnej aritmetike a až na konci realizovali modulárnu redukciu modulo n, t.j. platí:

$$(a+b) \bmod n = \big((a \bmod n) + (b \bmod n)\big) \bmod n \tag{1.3}$$

$$(a-b) \bmod n = \big((a \bmod n) - (b \bmod n)\big) \bmod n \tag{1.4}$$

$$(a*b) \bmod n = \big((a \bmod n) * (b \bmod n)\big) \bmod n \tag{1.5}$$

$$\big(a*(b+c)\big) \bmod n = \Big(\big((a*b) \bmod n\big) + \big((a*c) \bmod n\big)\Big) \bmod n \tag{1.6}$$

Príklad 1

Overte platnosť vzťahov (1.3) - (1.6) pre vybrané číselné hodnoty.

Pravidlá pre počítanie s kongruenciami v mnohom pripomínajú známe pravidlá pre počítanie s celými číslami. Platia napr. nasledujúce tvrdenia:

[3] Samozrejme v reálnom živote je medzi nimi zvyčajne veľký rozdiel.

T1: $a \equiv a \bmod n$ pre každé $a \in \mathbb{Z}$
T2: ak $a \equiv b \bmod n$, tak $b \equiv a \bmod n$
T3: ak $a \equiv b \bmod n$ a $b \equiv c \bmod n$, tak $a \equiv c \bmod n$
T4: ak $a \equiv b \bmod n$ a súčasne $c \equiv d \bmod n$, tak aj $a + c \equiv b + d \bmod n$
T5: ak $a \equiv b \bmod n$ a súčasne $c \equiv d \bmod n$, tak aj $a * c \equiv b * d \bmod n$
T6: ak $a * c \equiv b * c \bmod n$ a čísla c, n sú nesúdeliteľné, tak $a \equiv b \bmod n$

Príklad 2
Overte platnosť tvrdení **T1** *až* **T6** *pre vybrané číselné hodnoty.*

Modulárna aritmetika ma v súčasnej kryptografii široké využitie. Jadrom veľkej triedy najmodernejších kryptografických algoritmov a protokolov je algoritmus výpočtu mocniny nejakého čísla modulo n

$$a^x \bmod n \tag{1.7}$$

ktorý je postupnosťou násobení a modulárneho delenia. V prípade priameho výpočtu napr. pre $a^8 \bmod n$ (t.j. ak je exponent x mocninou 2) je potrebné realizovať 8 násobení a jednu $\bmod n$ operáciu v tvare

$$(a * a * a * a * a * a * a * a) \bmod n \tag{1.8}$$

Pomocou modulárnej aritmetiky je možné túto operáciu výrazne zrýchliť[4] pomocou výpočtu v tvare

$$\left(\left(a^2 \bmod n\right)^2 \bmod n\right)^2 \bmod n \tag{1.9}$$

V prípade, že exponent x nie je mocninou 2, je možné určiť výsledok pomocou postupu, ktorý nie je výrazne zložitejší, ako predchádzajúci výpočet. Postup výpočtu je dokumentovaný na nasledujúcom príklade.

Príklad 3
Určite hodnotu $5^{21} \bmod 7$.

Riešenie
Vyjadríme exponent 21 v dvojkovej sústave

$$21 = (10101)_2$$

t.j. platí $21 = 16 + 4 + 1$ a teda $5^{21} = 5^{16} * 5^4 * 5^1$.
Postupným umocňovaním dostávame

[4] V uvedenom príklade je zrýchlenie výpočtu nepodstatné, v prípade počítania napr. s 200 cifernými číslami, ktoré je v kryptografii bežné, je zrýchlenie veľmi výrazné.

$$5 \equiv 5 \bmod 7$$
$$5^2 \equiv 4 \bmod 7$$
$$5^4 \equiv 4^2 \equiv 2 \bmod 7$$
$$5^8 \equiv 2^2 \equiv 4 \bmod 7$$
$$5^{16} \equiv 4^2 \equiv 2 \bmod 7$$

a teda platí $5^{21} \equiv 2 * 2 * 5 \equiv 20 \equiv 6 \bmod 7$.

1.3 PRVOČÍSLA A KANONICKÝ TVAR ČÍSEL

Prvočíslo je celé číslo väčšie ako 1, ktoré je deliteľné iba sebou samým a jednotkou. Žiadne ďalšie číslo ho nedelí. Prvočísel je nekonečne veľa a sú využívané predovšetkým v algoritmoch šifrovania s verejným kľúčom.

V teórii čísel majú prvočísla významnú úlohu a predstavujú základ multiplikatívnej štruktúry prirodzených čísel, čo vyjadruje aj nasledujúca veta, známa aj ako **základná veta aritmetiky**:

Každé prirodzené číslo $m > 1$ sa dá jednoznačne (až na poradie) napísať v tvare

$$m = p_1^{\alpha_1} * p_2^{\alpha_2} * \ldots * p_k^{\alpha_k} \tag{1.10}$$

kde $p_1, p_2, \ldots, p_k$ sú navzájom rôzne prvočísla a $\alpha_1, \alpha_2, \ldots, \alpha_k$ sú prirodzené čísla.

O číslach zapísaných v tvare (1.10) hovoríme, že sú zapísané v **kanonickom tvare**. Úloha zistiť, či dané číslo je prvočíslo, alebo je zloženým číslom je netriviálna. Na zložitosti tejto operácie je napr. založený najznámejší algoritmus šifrovania s verejným kľúčom, ktorý bude analyzovaný v jednom z nasledujúcich cvičení.

1.4 NAJVÄČŠÍ SPOLOČNÝ DELITEĽ (GREATEST COMMON DIVISOR)

Dve čísla sú **nesúdeliteľné** (relative prime), ak nemajú žiadnych spoločných deliteľov okrem 1. Inak povedané čísla a, b sú nesúdeliteľné, ak je ich **najväčší spoločný deliteľ** rovný číslu 1. Túto skutočnosť budeme zapisovať v tvare

$$GCD(a,b) = 1 \tag{1.11}$$

Jeden zo spôsobov určenia najväčšieho spoločného deliteľa dvoch čísel je **Euklidov algoritmus**, ktorý Euclidos popísal vo svojej knihe Elementy okolo roku 300 pre našim letopočtom. Nie je však jeho autorom a predpokladá sa, že je ešte o 200 rokov starší. Euklidov algoritmus je najstarším netriviálnym algoritmom, ktorý je aj v súčasnosti stále výkonným algoritmom.

1.5 EUKLIDOV ALGORITMUS

Tento algoritmus vychádza zo skutočnosti, že ak x delí a a b, potom x delí tiež $a - (k * b)$ pre každé k. Aby sme pochopili, prečo toto tvrdenie platí, predpokladajme, že x delí a a b, t.j. platí $a = x * a_1$ resp. $b = x * b_1$. Platí teda

$$a-(k*b)=x*a_1-(x*k*b_1)=x*(a_1-k*b_1)=x*d \qquad (1.12)$$

takže x delí $a-(k*b)$.

Tento výsledok vedie k jednoduchému algoritmu pre výpočet $GCD(a,b)$. Predpokladajme, že chceme určiť $x=GCD(a,b)$, pričom $a>b$. Vyjadríme a ako

$$a=m*b+r \qquad (1.13)$$

pričom $0\le r<b$ (inak povedané počítame $m=a/b$ so zvyškom r). Ak $x=GCD(a,b)$, potom x delí a, x delí b a x delí r. Ale $GCD(a,b)=GCD(b,r)$ a $a>b>r\ge 0$. Na základe týchto skutočností je možné hľadanie GCD zjednodušiť tým, že prácu s a a b nahradíme prácou s b a r. Tento prístup vedie k jednoduchému iteratívnemu algoritmu, ktorý končí vtedy, ak zvyšok po delení dosiahne hodnotu 0. Euklidov algoritmus je možné formálne opísať takto

ALGORITMUS: Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa dvoch celých čísel
VSTUP: dve nezáporné celé čísla a, b pričom a $a\ge b$
VÝSTUP: najväčší spoločný deliteľ a a b

1. Pokiaľ $b\ne 0$ vykonaj
   a. Nastav $r\leftarrow a \bmod b,\quad a\leftarrow b,\quad b\leftarrow r$
2. Vráť(a)

pričom jeho použitie je ilustrované v nasledujúcom príklade.

Príklad 4
Určite hodnotu $GCD(3615807,2763323)$

Riešenie

$$3615807=(1)*2763323+852484$$
$$2763323=(3)*852484+205871$$
$$852484=(4)*205871+29000$$
$$205871=(7)*29000+2871$$
$$29000=(10)*2871+290$$
$$2871=(9)*290+261$$
$$290=(1)*261+29$$
$$261=(9)*29+0$$

a teda $GCD(3615807,2763323)=29$.

Je možné ukázať, že počet delení potrebných na vypočítanie najväčšieho spoločného deliteľa dvoch prirodzených čísel pomocou Euklidovho algoritmu nepresiahne päťnásobok počtu cifier (v dekadickom zápise) menšieho z čísel, čo dokumentuje jeho efektívnosť.

Euklidov algoritmus je možné modifikovať (rozšíriť) tak, že okrem hodnoty $d=GCD(a,b)$, sa určia aj celé čísla x a y splňujúce podmienku $ax+by=d$, pričom algoritmus je možné formálne opísať takto

ALGORITMUS: Rozšírený Euklidov algoritmus
VSTUP: dve nezáporné celé čísla a, b pričom $a \geq b$
VÝSTUP: $d = GCD(a,b)$ a celé čísla x, y splňujúce podmienku $ax + by = d$

1. Ak $b = 0$ potom nastav $d \leftarrow a,\quad x \leftarrow 1,\quad y \leftarrow 0,$ a vráť(d, x, y)
2. Nastav $x_2 \leftarrow 1$, $x_1 \leftarrow 0$, $y_2 = 0$, $y_1 \leftarrow 1$
3. Pokiaľ $b > 0$ vykonávaj[5]
   a. $q = \lfloor a/b \rfloor,\quad r \leftarrow a - qb,\quad x \leftarrow x_2 - qx_1,\quad y \leftarrow y_2 - qy_1$
   b. $a \leftarrow b,\quad b \leftarrow r,\quad x_2 \leftarrow x_1,\quad x_1 \leftarrow x,\quad y_2 \leftarrow y_1,\quad y_1 \leftarrow y$
4. Nastav $d \leftarrow a,\quad x \leftarrow x_2,\quad y \leftarrow y_2$ a vráť(d, x, y)

Rozšírený Euklidov algoritmus je s výhodou možné využiť predovšetkým na výpočet modulárnej inverzie.

1.6 MODULÁRNA INVERZIA (INVERZIA MOD N)

Nech $\odot$ vyjadruje nejakú číselnú operáciu (napr. operáciu $+, *$). Číslo i bude **identickým prvkom** pre $\odot$, pokiaľ bude platiť $x \odot i = x$ a $i \odot x = x$ pre každé číslo x. Napríklad 0 číslo je identickým prvkom pre $+$, pretože $0 + x = x$ a $x + 0 = x$. Podobne 1 je identickým prvkom pre $*$.

Nech i je identickým prvkom pre $\odot$. Číslo b bude **inverzným (opačným) prvkom** čísla a pre operáciu $\odot$ vtedy, ak $a \odot b = i$.

V modulárnej aritmetike je hľadanie inverzných prvkov

$$1 = (a * x) \bmod n \tag{1.14}$$

čo tiež zapisujeme v tvare

$$a^{-1} \equiv x \pmod n \tag{1.15}$$

podstatne komplikovanejšie ako v klasickej aritmetike. Niekedy jeho riešenie existuje a niekedy neexistuje. Obecne platí, že $a^{-1} \equiv x \pmod n$ má jediné riešenie vtedy, ak čísla a a n nemajú spoločného deliteľa. Ak a a n majú spoločného deliteľa, potom $a^{-1} \equiv x \pmod n$ nemá žiadne riešenie. Ak je teda n prvočíslo, potom každé prirodzené číslo v intervale 1 až $n-1$ bude mať v tomto intervale presne jednu inverziu modulo n.

1.7 FERMATOVA VETA

Fermatova veta hovorí, že pre každé prvočíslo m a každý prvok $a < m$ platí

$$a^m \bmod m = a \tag{1.16}$$

alebo

$$a^{m-1} \bmod m = 1 \tag{1.17}$$

[5] Funkcia $f(x) = \lfloor x \rfloor$ je v anglickej terminológii nazývaná „floor function”. Jej hodnotou je najväčšie celé číslo, ktoré je menšie alebo rovné x. Podobne funkcia $g(x) = \lceil x \rceil$ nazývaná „ceiling function“, vracia najmenšie celé číslo väčšie alebo rovné x.

Tento výsledok je návodom k získaniu hľadaných inverzií. Ak je m prvočíslo a prvok $a < m$, potom inverziou a je taký prvok x, pre ktorý platí

$$ax \bmod m = 1 \tag{1.18}$$

a teda pre výpočet modulárnej inverzie je možné využiť vzťah

$$a^{-1} = x = a^{m-2} \bmod m \tag{1.19}$$

Ak podmienky platnosti Fermatovej vety $a < m$, kde m je prvočíslo, nahradíme výrokom, že a nesmie byť násobkom m, potom výraz

$$a^{m-1} \equiv 1 \pmod{m} \tag{1.20}$$

vyjadruje **malú Fermatovu vetu**.

Malá Fermatova veta sa v tomto tvare dá napríklad použiť na zisťovanie toho, či dané prirodzené číslo je zložené. Ak k číslu m existuje (aspoň jedno) $a \in \{1, 2, \ldots, m-1\}$ také, že vzťah (1.20) nie je splnený, **tak číslo** m **je zložené číslo**. Táto, resp. podobné metódy sa využívajú pri generovaní náhodných prvočísel.

Príklad 5
Overte platnosť malej Fermatovej vety pre $a = 8$, $m = 7$.

1.8 EULEROVA FUNKCIA

Eulerova funkcia nazývaná tiež niekedy Eulerova $\phi(n)$ funkcia udáva počet celých kladných čísel menších ako n, pričom žiadne z týchto čísel nemá s n spoločného deliteľa, t.j. platí

$$\phi(n) = \sum_{\substack{a \le n \\ GCD(a,n)=1}} 1 \tag{1.21}$$

pričom hodnoty $\phi(n)$ pre $n = 1, 2, \ldots, 10$ sú uvedené v nasledujúcej tabuľke

n	1	2	3	4	5	6	7	8	9	10
$\phi(n)$	1	1	2	2	4	2	6	4	6	4

Ak je n prvočíslo, potom

$$\phi(n) = n - 1 \tag{1.22}$$

ak $n = p * q$, pričom p a q sú **rôzne** prvočísla, potom

$$\phi(n) = \phi(p * q) = (p-1) * (q-1) \tag{1.23}$$

čo je napr. jeden z dôvodov, prečo sa tieto čísla objavujú v známom šifrovacom algoritme s verejným kľúčom – RSA.

V prípade čísla m vyjadreného v kanonickom tvare (1.10) je možné určiť Eulerovu funkciu v tvare

$$\phi(m) = \left(p_1^{\alpha_1} - p_1^{\alpha_1 - 1}\right)\left(p_2^{\alpha_2} - p_2^{\alpha_2 - 1}\right)\ldots\left(p_3^{\alpha_k} - p_3^{\alpha_k - 1}\right) \tag{1.24}$$

S využitím funkcie $\phi(n)$ je možné sformulovať **Eulerovské zovšeobecnenie malej Fermatovej vety** v tvare:

Ak $GCD(a,n) = 1$, potom platí

$$a^{\phi(n)} \bmod n = 1 \tag{1.25}$$

Vzťah (1.25) umožňuje určiť modulárnu inverziu $a^{-1} \bmod n$ v tvare

$$a^{-1} = a^{\phi(n)-1} \bmod n \tag{1.26}$$

Príklad 6

Overte platnosť Eulerovského zovšeobecnenia malej Fermatovej vety pre konkrétne číselné hodnoty.

1.9 ČÍNSKA VETA O ZVYŠKOCH

Čínska veta o zvyškoch veľmi úzko súvisí s pojmom **zvyšková číselná sústava** (ZČS), ktorá je typickou **nepolyadickou (nepozičnou) číselnou sústavou**.

Číselná sústava, v ktorej význam číslic resp. symbolov nie je určený pozíciou, ale konfiguráciou týchto číslic resp. symbolov sa nazýva nepolyadická číselná sústava. Typickým príkladom je rímska číselná sústava a ŽČS.

ZČS je definovaná pomocou nesúdeliteľnej množiny L modulov

$$P = \{p_1, p_2, \ldots, p_L\} \tag{1.27}$$

pre ktoré platí

$$GCD\left(p_i, p_j\right) = 1 \quad pre\ i \neq j \tag{1.28}$$

Každé číslo X v ZČS je možné vyjadriť v tvare

$$X_{ZCS} \to \left(x_1, x_2, \ldots, x_L\right) \quad kde \quad x_i = X \bmod p_i \tag{1.29}$$

Číslo $M = p_1 * p_2 * \ldots * p_L$ je rozsah ZČS, pričom čísla 0 až $M-1$ majú v ZČS jednoznačné vyjadrenie. Prevod zo ZČS do dekadickej sústavy umožňuje **čínska veta o zvyškoch** v tvare

$$X_{dek} = \left(\sum_{i=1}^{L} m_i * \left(x_i * m_i^{-1}\right) \bmod p_i\right) \bmod M \tag{1.30}$$

pričom

$$m_i = \frac{M}{p_i} \tag{1.31}$$

a

$$\left(m_i^{-1} * m_i\right) \bmod p_i = 1 \tag{1.32}$$

pričom vzhľadom na fakt, že m_i a p_i sú navzájom nesúdeliteľné čísla (t.j. $GCD(p_i, m_i) = 1$), čo umožňuje využiť vzťahu (1.26) na výpočet m_i^{-1} v tvare

$$m_i^{-1} = \left(m_i^{\phi(p_i)-1}\right) \bmod p_i \tag{1.33}$$

Príklad 7
Vyjadrite číslo $X_{ZCS} = (1,0,4)$ v ZČS $P = (3,4,5)$ v dekadickom tvare.

ZČS umožňuje pri výpočte operácií $\otimes = \{+,-,*\}$ (nie delenia!) využiť prirodzený paralelizmus ZČS v tvare

$$Z_{dek} = X_{dek} \otimes Y_{dek} \xrightarrow{ZCS} z_i = x_i \otimes y_i \quad pre \quad i = 1,2,\ldots,L \tag{1.34}$$

Na základe vzťahu (1.34) je možné po konverzii do ZČS (ktorá samozrejme vyžaduje určitý počet matematických operácií) realizovať paralelne operácie s L číslami, ktorých dynamický rozsah je podstatne nižší ako dynamický rozsah vstupných čísel. To umožňuje využiť paralelizmus VLSI technológie a efektívnu realizáciu napr. pomocou zákazníckych obvodov.

1.10 KVADRATICKÉ RESIDUA A NERESIDUA

Ak je p prvočíslo a a je číslo v intervale $0 < a < p$, potom číslo a bude **kvadratickým reziduom** vtedy , ak

$$x^2 \equiv a \pmod{p} \tag{1.35}$$

pre nejaké x. Čísla a, ktoré podmienku (1.35) nesplňujú sa nazývajú **kvadratické nerezidua**. Je možné ukázať, že pre nepárne p existuje $(p-1)/2$ kvadratických reziduí mod p a rovnaký počet nereziduí mod p. Ak bude a kvadratickým reziduom mod p, potom pre a budú existovať presne dva korene odmocniny výrazu $x \equiv a^2 \pmod{p}$, jeden medzi 0 a $(p-1)/2-1$ a druhý medzi $(p-1)/2$ a $(p-1)$.

1.11 LEGENDEROVA FUNKCIA (LEGENDEROV SYMBOL)

Legenderova funkcia $L(a,p)$ definovaná pre akékoľvek celé číslo a prvočíslo $p > 2$ má nasledujúce hodnoty:

$$L(a,p) = 0 \quad \text{ak je } a \text{ deliteľné } p \tag{1.36}$$

$$L(a, p) = 1 \quad \text{ak } a \text{ je kvadratickým reziduom mod } p$$

$$L(a, p) = -1 \quad \text{ak } a \text{ je kvadratickým nonreziduom mod } p$$

Jeden z možných spôsobov určenia hodnoty $L(a, p)$ využíva vzťah

$$L(a, p) = a^{(p-1)/2} \bmod p \tag{1.37}$$

Iný spôsob určenia Legenderovej funkcie využíva nasledujúci algoritmus

$$\text{ak } a = 1, \text{ potom } L(a, p) = 1 \tag{1.38}$$

$$\text{ak je } a \text{ párne, potom } L(a, p) = L(a/2, p) * (-1)^{(p^2-1)/8}$$

$$\text{ak je } a \text{ nepárne (a tiež } \neq 1\text{), potom } L(a, p) = L(p \bmod a, a) * (-1)^{(a-1)*(p-1)/4}$$

1.12 *JAKOBIHO FUNKCIA (JAKOBIHO SYMBOL)*

Jakobiho funkcia $J(a,n)$ je zovšeobecnením Legenderovej funkcie pre zložené moduly a je definovaná pre akékoľvek celé číslo a a akékoľvek nepárne celé číslo n. Nech $n \geq 3$ je nepárne číslo vyjadrené v kanonickom tvare (1.10). Potom pre Jakobiho funkciu platí

$$J(a,n) = L(a, p_1)^{\alpha_1} L(a, p_2)^{\alpha_2} \dots L(a, p_k)^{\alpha_k} \tag{1.39}$$

a teda pre prvočíslo n platí

$$J(a,n) = L(a,n) \tag{1.40}$$

Jakobiho funkcia sa využíva napr. v niektorých **testoch pre vyhľadávanie prvočísel**.

1.13 *BLUMOVE (CELÉ) ČÍSLA*

Ak p a q sú dve prvočísla, ktoré sú kongruentné s 3 modulo 4 (t.j. platí $p \equiv 3 \bmod 4$, $q \equiv 3 \bmod 4$) , potom je číslo $n = pq$ **Blumovým (celým) číslom**. Blumové čísla sa využívajú v špeciálnych generátoroch pseudonáhodných čísel.

1.14 *GENERÁTORY*

Ak je p a $g < p$, pričom pre každé číslo b v intervale 1 až $(p-1)$ bude existovať nejaké číslo a také, že

$$g^a \equiv b \pmod{p} \tag{1.41}$$

potom číslo g bude **generátor** $\bmod p$. To isté je možné vyjadriť takto: číslo g je vzhľadom k číslu p **primitívne**.

Overovanie toho, či je dané číslo generátorom, nie je vo všeobecnom prípade jednoduchým problémom. Pokiaľ však poznáme rozklad $(p-1)$ na prvočiniteľe, je testovanie jednoduché. Predpokladajme, že jednotlivými prvočíselnými súčiniteľmi rozkladu $(p-1)$ sú čísla $q_1, q_2, \ldots, q_k$, t.j. pre kanonický rozklad čísla $(p-1)$ platí

$$(p-1) = q_1^{\alpha_1} * q_2^{\alpha_2} * \ldots * q_k^{\alpha_k}$$

Na overenie toho, či číslo g je generátorom $\bmod p$ vypočítame

$$g^{(p-1)/q} \bmod p \tag{1.42}$$

pre všetky $q = q_1, q_2, \ldots, q_k$. Pokiaľ pre niektorú hodnotu q bude výraz (1.42) **rovný** 1, potom g **nebude** generátorom.

Pokiaľ potrebujeme nájsť generátor $\bmod p$, stačí náhodne zvoliť nejaké číslo medzi 1 a $(p-1)$ a otestovať ho. Vzhľadom na to, že medzi týmito číslami je veľké množstvo generátorov, nejaký generátor nájdeme pomerne rýchlo.

Príklad 8

Overte že číslo 2 *je generátorom* $\bmod 11$.

1.15 DISKRÉTNE LOGARITMY

Modulárne umocňovanie v tvare

$$y = a^x \bmod n \tag{1.43}$$

patrí medzi tzv. **jednosmerné funkcie**, ktoré sú v modernej kryptológii často využívané. Vyčíslenie výrazu (1.43) je jednoduché. Inverzný (opačný) problém k modulárnemu umocňovaniu je hľadanie **diskrétneho logaritmu čísla**, t.j. nájdenie čísla x vo výraze

$$a^x \equiv y \pmod{n} \tag{1.44}$$

Toto je zložitý problém, a nie všetky diskrétne logaritmy majú riešenie.

1.16 GENEROVANIE PRVOČÍSEL

Mnohé moderné kryptografické algoritmy a protokoly vyžadujú prvočísla. Aby sme si vytvorili predstavu o ich počte uveďme niektoré fakty. Odhaduje sa, že vo vesmíre je približne 10^{77} atómov. Existuje približne 10^{151} prvočísel dĺžky 512 bitov (čo je v praktických kryptografických aplikáciách bežne využívaná veľkosť[6]) alebo kratších.

[6] Tieto čísla jasne dokumentujú vysokú abstrakciu matematiky. Počet atómov vo vesmíre je pre mnohých ľudí ťažko predstaviteľná hodnota. Na druhej strane počítanie s 512 bitovými (a aj s podstatne

Pravdepodobnosť, že náhodne zvolené číslo v blízkosti čísla n bude prvočíslo je približne 1 ku $\ln n$. Takže celkový počet prvočísel menších než n bude približne

$$\frac{n}{\ln n} \tag{1.45}$$

V moderných algoritmoch sa často vyskytujú dva typy otázok. Prvá: „*Je číslo n prvočíslo*?“ je **neporovnateľne jednoduchšia** ako druhá: „*Aké prvočísla sú rozkladom čísla n*?“ Práve tento výrazný rozdiel v zložitosti odpovedí na tieto otázky je základom pre významné kryptografické algoritmy.

Pri generovaní prvočísel sa vychádza z (relatívnej) jednoduchosti odpovede na prvú otázku. Náhodne sa vyberie číslo n a niektorým zo známych testov prvočíselnosti sa zistí sa, či je prvočíslom. Pokiaľ sa zistí, že číslo n nie je prvočíslo, je možné zvoliť ďalšie náhodné číslo a test opakovať. Iná stratégia výberu napr. využíva voľbu najbližšieho[7] nepárneho čísla $n+2$ a opakovanie testu prvočíselnosti.

Na testovanie prvočíselnosti je možné využiť niektorý zo známych testov (napr. Solovay-Strassenov, Lehmannov, Rabin-Millerov a pod.).

Lehmannov test

Tento jednoduchší test prvočíselnosti čísla p je realizovaný pomocou nasledujúcich krokov:

1) vyberieme náhodne číslo a menšie než p
2) vypočítame $a^{(p-1)/2} \bmod p$
3) ak je $a^{(p-1)/2} \not\equiv 1 \pmod{p}$ alebo $a^{(p-1)/2} \not\equiv -1 \pmod{p}$, potom p určite nie je prvočíslo
4) ak je $a^{(p-1)/2} \equiv 1 \pmod{p}$ alebo $a^{(p-1)/2} \equiv -1 \pmod{p}$, potom pravdepodobnosť toho, že p nie je prvočíslo nebude väčšia než 50%

Tento test zopakujeme t-krát (samozrejme vždy s iným náhodne vybraným číslom a). Ak bude test úspešný t-krát, potom **pravdepodobnosť** toho, že p nebude prvočíslom bude $1/2^t$.

Rabin-Millerov test

Tento veľmi jednoduchý test je v podstate zjednodušenou formou algoritmu doporučovaného normou pre digitálne podpisy. Pre náhodne zvolené číslo p spočítame číslo b, pričom b je počet delení dvojčlenu $(p-1)$ číslom 2. (t.j. 2^b je najväčšia mocnina čísla 2, ktorá delí $(p-1)$). Potom určíme také m, pre ktoré bude platiť $p = 1 + 2^b * m$. Ďalej postupujeme pomocou nasledujúcich krokov:

1) zvolíme náhodné číslo a, tak aby platilo $a < p$
2) položíme $j = 0$ a $z = a^m \bmod p$
3) ak bude $z = 1$ alebo $z = p - 1$, potom p testom prejde a môže byť prvočíslom
4) ak bude $j > 0$ a $z = 1$, potom p prvočíslom nebude

väčšími!) číslami je s využitím bežne dostupných technických prostriedkov v podstate len vec rutiny. Naviac matematici pracujú s takýmito číslami často aj bez technických prostriedkov.

[7] Vzhľadom na veľkosť množiny prvočísel narazíme na prvočíslo veľmi rýchlo.

5) položíme $j = j + 1$, ak bude $j < b$ a $z \neq p - 1$, položíme $z = z^2 \bmod p$ a vrátime sa ku kroku (4); ak bude $z = p - 1$, tak p testom prejde a môže byť prvočíslom
6) ak bude $j = b$ a $z \neq p - 1$, potom p nie je prvočíslom.

Pravdepodobnosť toho, že testom prejde ako prvočíslo zložené číslo klesá v tomto teste rýchlejšie ako v predchádzajúcich a je rovná hodnote $1/4^t$, pričom t je počet iterácií.

Prvočísla generované uvedenými pravdepodobnostnými testmi sa niekedy označujú tiež termínom **prvočísla priemyselnej kvality**. V niektorých kryptografických algoritmoch (napr. RSA) sa využíva číslo n, ktoré je súčinom dvoch veľkých prvočísel p a q. Niekedy sa od týchto čísel vyžaduje, aby to boli tzv. **silné prvočísla** (strong primes). Tieto prvočísla majú určité vlastnosti, ktoré sťažujú rozklad čísla n na prvočinitele s využitím špecifických postupov. Medzi doporučované vlastnosti patria najmä tieto:

- najväčší spoločný deliteľ čísel $(p-1)$ a $(q-1)$ má byť malý
- obe čísla $(p-1)$ a $(q-1)$ majú mať veľké prvočiniteľe p' a q'
- obe čísla $(p'-1)$ a $(q'-1)$ majú mať veľké prvočiniteľe
- obe čísla $(p'+1)$ a $(q'+1)$ majú mať veľké prvočiniteľe
- obidva čísla $(p-1)/2$ a $(q-1)/2$ majú byť prvočísla (táto podmienka zabezpečuje zároveň splnenie prvých dvoch podmienok).

LITERATÚRA

[1] Přibil, J. – Kodl, J.: Ochrana dat v iformatice. Vydavatelství ČVUT, Praha 1996, ISBN.

[2] Grošek, O. – Porubský, Š.: Šifrovanie – algoritmy, metódy, prax. Grada, 1992, ISBN 80-85424-62-2.

[3] Koblitz, N.: A Course in Number Theory and Cryptography. Springer-Verlag, New York, 1994, ISBN 3-540-94293-9.

[4] Menezes, J.A. – Oorschot, P.C. – Vanstone, S.A.: Applied Cryptography. CRC Press, New York 1997, ISBN 0-8493-8523-7.