FAKTORIZACE (1)

Dvě čísla za 200 000 dolarů

Představte si, že máte k dispozici veškerou výpočetní techniku na Zemi, všechny znalosti současné vědy a hodně peněz. Jak velké číslo byste byli schopni faktorizovat? Pokusíme se tuto otázku zodpovědět a seznámíme vás se známými metodami faktorizace.

NEJEN STARÝ MATEMATICKÝ PROBLÉM

Jak dlouho by vám trvalo rozložení čísla 323 na prvočinitele? S tužkou v ruce byste na řešení (17 a 19) přišli možná za několik minut, s kalkulačkou za desítky vteřin. Jistě není třeba zdůrazňovat, že při řádově větších číslech by to šlo o hodně pomaleji...

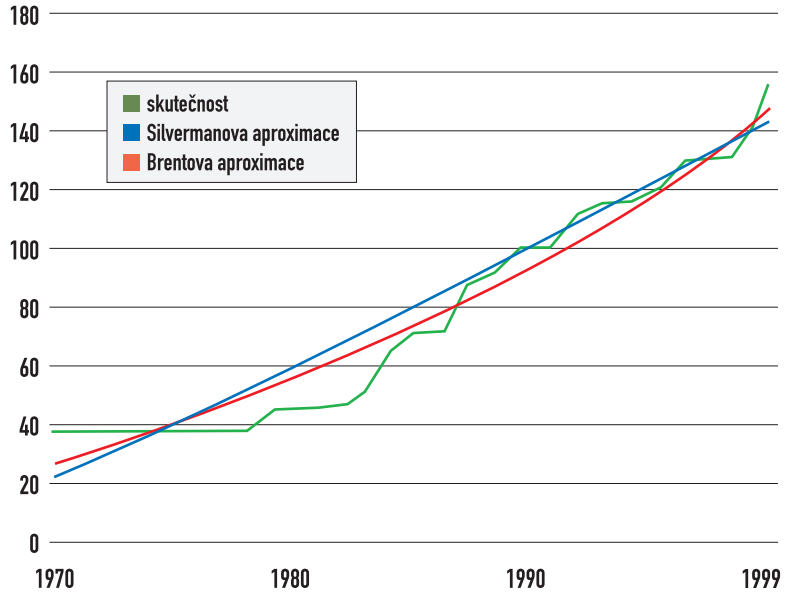
Faktorizace, tedy nalezení všech prvočinitelů složeného čísla, je velmi starý matematický problém, který **není dosud uspokojivě vyřešen**. Známe sice metody, jak jej řešit, ale s rostoucí délkou předloženého čísla roste i složitost těchto metod, a to tak drasticky, že od určité hranice už nemáme ani dost času, ani výpočetní kapacity k řešení. (Články k faktorizaci i ke schopnostem lušticího zařízení TWINKLE jsme již několikrát publikovali – viz infotipy. Jsou také na internetu a vážným zájemcům doporučuji je prostudovat.)

Aniž bychom se příliš vzdálili od obecné formulace, v dalším se soustředíme jen na **základní úlohu** faktorizace čísla, které je součinem **pouze dvou prvočísel**. Řešení tohoto problému totiž mj. umožňuje luštit asymetrickou šifru RSA! Zatím největší číslo n = p*q, které se (v r. 1999) podařilo faktorizovat, má 155 dekadických cifer (512 bitů). Ještě nějaké rezervy máme, ale čísla, která mají třeba tisíc cifer, jsou zatím zcela mimo naše možnosti. Rekordy ve faktorizaci speciálních čísel i modulů RSA ukazuje tabulka 1. Ještě připomeňme další mezníky – faktorizace čísel RSA-100 (1991, 100 cifer), RSA-110 (1992, 110 cifer) a RSA-120 (1993, 120 cifer) z minulé soutěže RSA – i když nebyly rekordní v daném roce.

Z tabulky i z obrázku 1 je zřejmé, že faktorizace velkých čísel nijak zvlášť rychle nepostupuje a spíše to vypadá, že se problém táhne jako med. Je vidět, že zde chybí nějaký zásadní objev, který by postup výrazně urychlil – i když nevíme, zda si takový pokrok vlastně máme přát. Ten, kdo by takový objev učinil, by totiž byl schopen nabourat se prostřednictvím luštění RSA do nejdůležitějších oblastí mezinárodního bankovnictví, elektronického obchodu, bezpečnostních systémů a počítačových sítí...

Protože jsou k dispozici údaje o faktorizaci významných speciálních čísel i údaje o faktorizaci čísel z dosavadní soutěže společnosti RSA Security Inc. (dále jen RSASI), lze závislost velikosti faktorizovaného čísla na roku faktorizace různými způsoby odhadovat a aproximovat. Uvedeme dva názory na tento vývoj. První (Silverman) říká, že pro počet cifer faktorizovaného čísla D lze odvodit lineární vztah D = 4,23 * (rok - 1970) + 23, druhý (Brent) má mnohem optimističtější aproximaci $D = ((\text{rok} - 1928{,}6) / 13{,}24)^3$, viz obrázek 1. Faktorizace 1024bitového modulu RSA by se tak podle prvního odhadu mohla uskutečnit v roce 2037 a podle druhého v roce 2018. Obrázek 1 nasvědčuje spíše Brentově aproximaci, ale jaký bude skutečný vývoj, je opravdu „ve hvězdách“. Ze Silvermanových úvah vyplývá, že určitý konzervativismus je na místě, protože úloha faktorizace se netýká jen počtu operací, ale i potřebné kapacity počítačové paměti. Faktor paměti oproti tomu podceňuje vůči Silvermanovi kontroverzní a vzhledem k úspěchům faktorizace optimistická práce Lenstry a Verheula (viz infotipy).

Obr. 1. Závislost délky čísla na roku faktorizace



BUDE SESTROJEN „RSA-CRACKER“?

Protože jde o dost peněz, vědeckou ctižádost i komerční zájmy (zejména RSASI), vedou se dosti ostré diskuse k možnostem faktorizace v příštích letech a desetiletích. Mnoho laiků u nás i ve světě například z faktorizace 512bitového čísla, uskutečněné v roce 1999, činí ukvapené závěry o bezpečnosti 1024bitového modulu a doporučuje délky modulů 2048 bitů nebo raději 4096 bitů! Líbí se jim zdvojnásobovat délky modulů, aniž by tušili, co je za tím ukryto. Naproti tomu autoři vědeckých článků na toto téma si jasně uvědomují, že odhady možného vývoje jsou pouhým žonglováním s čísly a sliby, a proto jasně uvádějí předpoklady svých závěrů: „Když se situace bude vyvíjet...“, „Za předpokladu, že by...“ apod. Další →

Rok	Počet cifer	Číslo n	Kdo faktorizoval	Metoda	Hardware
1970	39	2128 + 1	Brillhart/Morrison	CF	IBM Mainframe
1978	45	2223 – 1	Wunderlich	CF	IBM Mainframe
1981	47	3225 – 1	Gerver	QS	HP-3000
1982	51	591 – 1	Wagstaff	CF	IBM Mainframe
1983	63	1193 + 1	Davis/Holdridge	QS	Cray
1984	71	1071 – 1	Davis/Holdridge	QS	Cray
1986	87	5128 + 1	Silverman	QS	LAN Sun
1987	90	5160 + 1	Silverman	QS	LAN Sun
1988	100	11104 + 1	Internet	QS	Distribuované výpočty
1990	111	2484 + 1	Lenstra/Manasse	QS	Distribuované výpočty
1991	116	10142 + 1	Lenstra/Manasse	QS	Distribuované výpočty
1994	129	RSA-129	Atkins	QS	Distribuované výpočty
1996	130	RSA-130	Montgomery	GNFS	Distribuované výpočty
1999	140	RSA-140	Montgomery	GNFS	Distribuované výpočty
1999	155	RSA-512	Montgomery	GNFS	Distribuované výpočty

Poznámka: CF – metoda řetězových zlomků, QS – kvadratické síto, GNFS – General Number Field Sieve

Tabulka 1. Rekordy ve faktorizaci čísel

skupina laiků však interpretuje jen závěry, aniž by si uvědomovala jejich předpoklady!

A tak se opakuje podobná situace jako při diskusích o luštitelnosti algoritmu DES. Názory, že DES je nerozluštitelný apod., prostě nebylo možné „utlouci“, dokud nebyl sestrojen hmatatelný luštící stroj DES-Cracker (viz např. Chip 11/98, 12/98). Teď je to naopak, neexistují stoprocentně přesvědčivé protiargumenty na laické fantazie, že ten nebo onen modul RSA je příliš malý a bude zcela určitě vbrzku luštitelný. RSASI se s tím vyrovnala tak, že vypsala novou veřejnou soutěž na faktorizaci 576- až 2048bitových čísel, jak ukazuje tabulka 2 – povšimněte si hezkých dolarových vábniček v posledním sloupci. Všechny podrobnosti k soutěži naleznete na adrese v infotipech.

Je to skvělý komerční tah, ale poslouží i vědě, protože k problému přitáhne více lidí a ukáže, kde jsou reálné hranice. Pokud se najde dost zájemců o poskytnutí výpočetního času, mohlo by být poměrně brzo dosaženo faktorizace 576bitového modulu. Naproti tomu o 2048bitovém čísle RSASI říká, že by mělo vydržet desítky let. Současné metody však tuto faktorizaci odsouvají spíše do nekonečna... My se teď ale odpoutáme od dohadů a zaměříme se na fakta.

NEJDE O PRVOČÍSLO?

Dříve než se pustíme do faktorizace nějakého čísla, měli bychom si ověřit, že je to skutečně číslo složené, tj. že si z nás někdo nestřílí a nepodstrkuje nám prvočíslo. Na to existují testy s exaktní odpovědí ano/ne (naposledy jimi bylo ověřeno prvočíslo dlouhé přes 1500 cifer), například *Cohen-Lenstrův Jacobi sum test* nebo *Atkinův test*, ale vzhledem ke značné složitosti se nepoužívají.

V praxi se lépe osvědčují tzv. pravděpodobnostní testy, které jsou velmi rychlé a dobře programovatelné. Pokud o daném čísle tvrdí, že je složené, je to zaručeně pravda – pokud ale dojdou k závěru, že se jedná o prvočíslo, mohou se s určitou pravděpodobností mýlit. Ta se ale dá stlačit pod libovolně předem určenou mez vhodnou volbou bezpečnostního koeficientu. Zatím nejlepší a nejpoužívanější praktický test na prvočíselnost je Miller-Rabinův test, vycházející z testu Fermatova.

FERMATŮV A MILLER-RABINŮV TEST

Oba testy využívají malou Fermatovu větu, která říká, že pokud gcd(a, n) = 1, tj. největší společný dělitel čísel *a* a *n* je 1 a *n* je skutečně prvočíslo, potom a^{n-1} (mod n) = 1. Při testování složenosti se proto náhodně volí číslo *a* a zjišťuje se, zda tato rovnost platí. Pokud neplatí, *a* se nazývá *(Fermatův) svědek složenosti* a *n* je skutečně složené. Jestliže rovnost platí, *n* je pravděpodobně prvočíslo, ale nemáme ještě jistotu. Tu posílíme volbou dalšího náhodného čísla *a* a opět zkoumáme, zda *a* je svědek složenosti. Pokud ani po *t* pokusech nenalezneme svědka složenosti, je vysoce pravděpodobné, že předložené číslo *n* je prvočíslo. (Pro zajímavost poznamenejme, že Fermatův test s velkou pravděpodobností **nepozná** celou třídu zvláštních složených čísel, která se nazývají Carmichaelova[1], ale vzhledem k dalšímu to není nijak zvlášť alarmující výsledek.)

Miller-Rabinův (MR) test využívá jemnější fakt uvedený na obrázku 3 a je účinnější než Fermatův test (například *a* = 2 by pro *n* = 561 podle Fermatova testu svědčilo o prvočíselnosti, zatímco pro MR test by bylo svědkem složenosti). Pokud *n* je prvočíslo, je n-1 sudé, a dá se tedy zapsat ve tvaru $n–1 = 2^s * r$, kde *r* je liché číslo. V MR testu generujeme náhodně číslo *a* (opět to provádíme *t*-krát, kde *t* je bezpečnostní parametr) a počítáme postupně posloupnost a^r (mod n), a^{2r} (mod n), a^{4r} (mod n), ... až a^{n-1} (mod n); podle Fermatovy věty dospějeme

Obr. 2. První číslo určené k faktorizaci, které už určitě „padne" brzy.

188198812920607963838697239461650439807163563379417382700763356422988859715234665485319060606504743045317388011303396716199692321205734031879550656996221305168759307650257059

nakonec k jedničce. MR test je založen na faktu, že uvedená posloupnost musí mít buď tvar 1, 1, ..., 1, nebo x, y, ..., z, –1, 1, 1, ..., 1, kde *x* až *z* označují libovolná (i nepovinná) čísla. Pokud uvedená posloupnost tento tvar nemá, *n* je složené číslo s konečnou platností. Pokud test řekne, že *n* je prvočíslo, může se mýlit s pravděpodobností $(1/4)^t$. Proto podle toho, jakou chceme mít jistotu, volíme parametr *t* (například t = 10 dává pravděpodobnost chyby asi 1 k milionu).

METODY FAKTORIZACE

Snad každý umí faktorizovat dané číslo metodou „kanadských dřevorubců", kdy prostě zkoušíme dané číslo *n* dělit postupně všemi prvočísly 3, 5, 7, 11, ... Dělitele určitě najdeme, ale asi to nebude nejrychlejší. Skutečně, obecně se při takovém zkoušení můžeme dostat až do těsné blízkosti čísla $n^{1/2}$. Jenže jak to udělat jinak?

POLLARDOVA P-1 METODA

Tato metoda byla popsána v roce 1974 a také využívá malé Fermatovy věty. Opět hledáme prvočíselný faktor *p* čísla *n*. Jak víme, p–1 je sudé číslo, a má proto jeden z dělitelů dvojku. **Metoda je účinná, pokud ani další dělitelé p–1 nejsou příliš velká čísla** – jsou dejme tomu omezené shora číslem B (pak říkáme, že p–1 je B-hladké, resp. B-smooth). Základní myšlenka je založena na tom, že pokud máme nějaké číslo Q takové, že p–1 dělí Q, pak podle Fermatovy věty p dělí $a^Q - 1$. Protože p dělí také n (je to jeho hledaný faktor), nalezneme ho jako dělitel čísla d = gcd(a^Q–1, n). Pokud se stane, že d = n, algoritmus selže, ale v našem případě, kdy n má dva velké faktory, je to velmi nepravděpodobné.

Zbývá ještě definovat číslo Q. Protože víme, že p–1 má všechny faktory ≤ B, můžeme definovat $Q = \Pi_{q \le B}\, q^{M(q)}$, kde M(q) = [ln(n) / ln(q)] a [] označ-

MR test vyplývá z následujícího faktu:
Nechť n je liché prvočíslo. Vyjádříme n – 1 ve tvaru $2^s * r$, kde r je liché. Nechť a je libovolné číslo nesoudělné s n. Potom buď $a^r \bmod n = 1$, nebo pro nějaké j = 0... s–1 platí $a^r \bmod n = n–1$.

Vstup: liché číslo n > 2 a bezpečnostní parametr t > 0
Výstup: odpověď "n je prvočíslo" nebo "n je složené"

1. Vyjádři n ve tvaru $n-1 = 2^s * r$, kde r je liché
2. For i = 1...t do
   {
   Zvol náhodně číslo a , 1 < a < n-1
   Vypočítej $y = a^r \bmod n$
   If (y ≠ 1 and y ≠ n-1) then do
   j = 1
   While (j ≤ s-1 and y ≠ n-1) do:
   { $y = y^2 \bmod n$
   If (y = 1) then return ("n je složené")
   j = j +1
   }
   If (y ≠ n-1) then return ("n je složené")
   }
3. Return ("n je prvočíslo")

Obr. 3. Pseudokód Miller-Rabinova testu prvočíselnosti

Vstup: složené číslo n, které není mocninou prvočísla
Výstup: netriviální faktor d čísla n

1. Zvol hranici B (například 10^5 nebo 10^6)
2. Vyber náhodné číslo a z intervalu [2, n] a vypočti d = gcd(a,n). Je-li d ≥ 2, return(d).
3. Pro každé prvočíslo q ≤ B :
   {
   M(q) = [ln(n) / ln(q)]
   $a = a^{q \text{ na } M(q)} \pmod n$
   }
4. Vypočti d = gcd(a – 1, n). Je-li d = 1 nebo d = n, ukonči algoritmus s chybou.
5. Return(d)

Obr. 4. Pseudokód Pollardova p-1 algoritmu

Vstup: složené číslo n = p*q
Výstup: netriviální dělitel n

1. $x_0 = 2$
2. For i = 1,2,... do
   {
   $x_i = f_{(x_{i-1})} = x_{i-1}^2 + 1 \pmod n$
   $x_{2i} = f(f(x_{2i-2})) = (x_{2i-2}^2 + 1)^2 + 1 \pmod n$
   $d = \gcd(x_i - x_{2i}, n)$
   je-li 1 < d < n, ukonči smyčku a vrať hodnotu d
   je-li d = n, přeruš algoritmus a zvol jinou funkci f
   }

Obr. 5. Pseudokód Pollardova algoritmu s Floydovým trikem

Číslo	Počet bitů	Počet dekadických cifer	Odhad roku faktorizace*	Odměna za faktorizaci (v USD)
RSA-576	576	174	2006	10 000
RSA-640	640	193	2010	20 000
RSA-704	704	212	2015	30 000
RSA-768	768	232	2019	50 000
RSA-896	896	270	2028	75 000
RSA-1024	1024	309	2038	100 000
RSA-1536	1536	463	2074	150 000
RSA-2048	2048	617	2110	200 000

*rok, kdy by mohlo dojít k faktorizaci na základě Silvermanovy aproximace

Tabulka 2. Soutěžní čísla a odměny za jejich faktorizaci

čuje celou část čísla. Vidíme, že M(q) je nejvyšší možné a takové, aby ještě platilo $q^{M(q)} \leq n$, takže v Q jsou obsaženy všechny možné mocniny všech možných prvočinitelů čísla p–1. Proto p–1 musí dělit Q, čímž je myšlenka uzavřena. Praktickou realizaci algoritmu ukazuje pseudokód na obrázku 4 a konkrétní příklad v tabulce 3.

ZÁKLADNÍ MYŠLENKA FAKTORIZAČNÍCH METOD

Dalším možným trikem, jak zjistit nějaký dělitel čísla *n*, je nalézt čísla *x, y* tak, že $x^2 \equiv y^2 \pmod n$. Odtud potom vyplývá, že n dělí (x – y)(x + y), a pokud máme štěstí, prvočinitelé p, q čísla n budou „rozděleni" zvlášť do obou čísel x – y, x + y. Jednoho z nich pak snadno nalezneme jako největší společný dělitel x – y a n, tj. gcd(x – y, n). Zajímavé je, že toto je základní myšlenka všech dosud známých faktorizačních metod (obecných čísel). Pokud budete chtít udělat do faktorizace průlom, oprostěte se od ní a vymyslete něco jiného!

POLLARDOVA RÓ METODA

Teď se zastavíme u *Pollardovy ró metody*, objevené v roce 1975. Má širší význam a použití než jen pro faktorizaci, jak ukazuje i článek v Chipu 8/01 (viz infotipy). Naleznete v něm definici, rógraf i využití této metody pro hledání kolizí hašovacích funkcí. My teď popíšeme její variantu pro faktorizaci. Připomeňme si jen, že řecké písmeno **ρ,** po němž je metoda pojmenována, se náramně podobá obrázku, který dostaneme Pollardovou metodou – ocásek, který se napojuje na kruh.

Uvažujme náhodnou funkci f: S→S na konečné množině S o *n* prvcích (pro nás to budou čísla 0, 1 ..., n-1, protože vše počítáme modulo n), vyberme náhodně $x_0 \in S$ a sestrojme posloupnost $x_0, x_1, x_2, \ldots$ definovanou vztahem $x_{i+1} = f(x_i)$. Je to náhodná procházka po číslech množiny S, a protože S je konečná, po určité době se dostaneme do bodu, kde jsme už byli, tj. pro nějaká *r, s* bude platit $f(x_r) = f(x_s)$. Jakmile shoda nastane, nově počítané hodnoty x_{r+j} pro j = 1, 2, ... budou rovny předchozím x_{s+j} a graficky to bude znamenat, že jsme se už dostali do oblasti kruhu. Pokud volíme $f(x) = x^2 + 1 \pmod n$, dostáváme v době průseku, že $x_r^2 \equiv x_s^2 \pmod n$, čili naši cílenou rovnici!

Z teorie náhodných funkcí víme, že očekávaná délka ocásku λ i délka kruhové části μ jsou přibližně rovny $(\pi*n/8)^{1/2}$. Jejich součet je $(\pi*n/2)^{1/2}$ a udává střední dobu, po kterou musíme čekat na naši shodu. Protože očekáváme nalezení prvočinitele *p* řádově rovného $n^{1/2}$, vyplatí se nám nečekat na shodu $f(x_r) = f(x_s)$, ale sledovat jen okamžik, kdy $\gcd(f(x_r) – f(x_s), n)$ bude větší než 1 a menší než n. V tom případě je to přímo náš prvočinitel p! Střední doba čekání na shodu je pak $(\pi*p/2)^{1/2}$, což je mnohem příznivější, neboť p bude číslo blízké $n^{1/2}$.

Postup Pollardova p-1 algoritmu pro faktorizaci čísla n = 19048567		
Krok 1: B = 19		
Krok 2: a = 3, gcd(a, n) = 1		
Krok 3: prvočísla q = 2, 3, 5, 7, 11, 13, 17, 19		
q	**M(q**	**a**
2	24	2293244
3	15	13555889
5	10	16937223
7	8	15214586
11	6	9685355
13	6	13271154
17	5	11406961
19	5	554506
Krok 4 a 5: d = gcd(554506-1, n) = 5281		
Výsledný rozklad n = 5281 * 3607		

Tabulka 3. Pollardův p-1 algoritmus pro faktorizaci čísla 19048567

Postup Pollardova algoritmu pro faktorizaci čísla n = 455459			
i	**x (i)**	**x (2i)**	**d**
0	2	2	
1	5	26	1
2	26	2871	1
3	677	179685	1
4	2871	155260	1
5	44380	416250	1
6	179685	43670	1
7	121634	164403	1
8	155260	247944	1
9	44567	68343	743
10	416250		
11	171557		
12	43670		
13	62068		
14	164403		
15	42973		
16	247944		
17	193153		
18	68343		

Tabulka 4. Pollardův algoritmus pro faktorizaci čísla 455459

→ FLOYDŮV TRIK

U Pollardovy metody musíme ukládat hodnoty x_i, a u každé nově vytvořené musíme kontrolovat, zda se nerovná některé předchozí. To vyžaduje dost paměti. Floydovo vylepšení zde spočívá v tom, že hodnoty x_i neukládáme a místo toho začínáme s **párem** hodnot (x_1, x_2). Iterativně vypočítáváme (x_2, x_4), (x_3, x_6), (x_4, x_8), ... obecně podle vztahu $x_i = f(x_{i-1})$ a $x_{2i} = f(f(x_{2i-2}))$ a čekáme tak dlouho, až „shoda" nastane přímo v našem páru. Shodu nyní chápeme také jako bod, kdy největší společný dělitel čísel x_m a x_{2m} v našem aktuálním páru je smysluplný, tj. když $1 < d = \gcd(x_m - x_{2m}, n) < n$. V tom případě je *d* právě hledaný dělitel čísla n.

Pseudokód této metody je na obrázku 5, v tabulce 4 uvádíme příklad postupu pro faktorizaci čísla 455459. Ještě poznamenejme, že místo konstanty 1 v polynomu $f(x) = x^2 + 1$ můžeme použít jiné vhodné číslo (kromě 0 a –2).

ZATÍM TO „NEMÁ ŠŤÁVU"

Pollardova metoda je spolehlivá, ale hodí se spíše na „menší" čísla n, neboť její složitost je proporcionální číslu $n^{1/2}$. Říkáme „menší" čísla, ale i na domácím PC se klidně můžeme pustit do faktorizace deseticiferného čísla. U RSA nás však zajímají čísla mnohonásobně delší. O rafinovanějších metodách si povíme příště – nebude to ovšem nic pro domácí počítač, spíš asi tak pro sto milionů počítačů, každý s operační pamětí 170 GB RAM...

ZÁVĚR

Úloha faktorizace je starý matematický problém, mající své důsledky pro současnou kryptografii. Na jeho výpočetní složitosti je založen algoritmus RSA. Pokud by došlo k zásadnímu urychlení faktorizačních metod, musela by se odpovídajícím způsobem zvyšovat délka modulu RSA, aby se zvýšila jeho bezpečnost. Současné faktorizační metody však nic takového nenaznačují, naopak vývoj z hlediska kvality spíše stagnuje. V tomto článku jsme vysvětlili Pollardovy algoritmy, příště se budeme zabývat dalšími metodami.

Vlastimil Klíma | vlastimil.klima@i.cz

---

[1] Carmichaelovovo číslo *n* je složené číslo takové, že $a^{n-1} \equiv 1 \pmod n$ **pro všechna** celá čísla *a*, nesoudělná s *n*. V intervalu [2, N] je více než $N^{2/7}$ těchto čísel a vůbec nejmenší Carmichaelovo číslo je 561=3*11*17.

INFOTIPY

Vše o nové soutěži
▸ *http://www.rsasecurity.com/rsalabs/challenges/factoring/index.html*

O faktorizaci a zařízení TWINKLE
▸ Rosa, T.: „Na to vezmi LED!", Chip 8/99 a 9/99, „Jde to i bez Twinklu", Chip 10/99

O různých faktorizačních metodách
▸ Menezes, A. J., Oorschot, P. C., Vanstone, S. A.: „Handbook of Applied Cryptography", CRC Press, New York, 1997

Podstata algoritmu RSA
▸ Klíma, V.: „Bude nás podepisovat RSA?", Chip 9/00

Bezpečnost a faktorizace podle Silvermana
▸ *http://www.rsasecurity.com/rsalabs/bulletins/bulletin13.html*

Bezpečnost a faktorizace podle Lenstry a Verheula
▸ Lenstra, A. K., Verheul, E. R.: „Selecting Cryptographic Key Sizes", PKC2000, Australia, January 2000, nyní aktualizováno na *http://www.cryptosavvy.com/joc.pdf*

Pollardova ró metoda z jiného pohledu
▸ Rosa, T.: „Podpis k narozeninám", Chip 8/01

(Články z Chipu naleznete také v elektronické podobě na *http://www.decros.cz/bezpecnost/_kryptografie.html)*