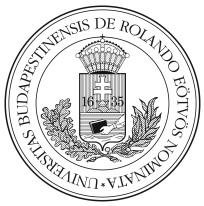
Számítógépes Számelmélet

Gyarmati Katalin

katalin.gyarmati@ttk.elte.hu

Eötvös Loránd Tudományegyetem

Egyetemi Jegyzet



ELTE TTK Matematikai Intézet

2022

ELTE jegyzetpályázat támogatásával készült

Lektorálta: Dr. Pethő Attila

az MTA r. tagja

(2023.08.31)

Tartalomjegyzék

1. Bevezetés

A jegyzet a modern számelmélet egy kriptográfiához és számítógépekhez is kapcsolódó alkalmazott területén ír le számos fontos eredményt. Tartalmazza az elemi műveletek időigényét, de leírja a gyors szorzás (FFT algoritmus) alapjait is. Betekintést nyújt a kriptográfia néhány fontos történelmi fejezetébe. Ismert és népszerű kriptográfiához kapcsolódó számelméleti algoritmusok alapos analizálását tartalmazza, így például az RSA helytelen alkalmazásai esetén milyen buktatók lehetnek (messze túlmenően azon, hogy az RSA-ban alkalmazott két prím nem lehet közel egymáshoz, illetve a faktorizáláshoz fűződő kapcsolaton túl is lehetnek problémák). Ismerteti a diszkrét logaritmus probléma megoldására vonatkozó modern módszereket (a fenti algoritmusok viszonylag lassúak). Szó van prímtesztekről és faktorizációs algoritmusokról is. A pszeudovéletlen generálás modern alapjait is áttekintjük. Ezentúl pedig röviden megismerkedhetünk az elliptikus görbéken alapuló kriptográfiával is. Megjegyezendő azonban, hogy a számítógépes számelmélet a jegyzetben tárgyaltaknál jóval tágabb terület, melyet a MathSciNet 11Y kóddal kategorizál, azonban terjedelmi okoknál csak a fentiekre szorítkoztam.

A jegyzet több fejezetének alapját a következő két könyv képzi: Neal Koeblitz, *A Course in Number Theory and Cryptography*, Springer, 1994. Abhijit Das, Computational Number Theory, CRC Press, 2013. A „Titkosírások a történelemben” szóló fejezet megírásának alapja legtöbbször a kapcsolódó Wikipédia oldalak voltak. AZ RSA helytelen alkalmazásairól szóló rész Dénes Tamás írására alapozódott. Az elliptikus görbékről szóló fejezet leginkább Andrea Corbellini internetes jegyzeteire és Lenstra faktorizációra vonatkozó cikkére támaszkodott. A fentieken kívül azonban számos más irodalmat is használtam. A teljes bibliográfia mindig az adott fejezet végén található, ahol is a felhasznált ábrák és képek forrását is feltüntettem.

Az olvasóknak kellemes időtöltést kívánok!

2. Titkosírások a történelemben

2.1. Üzenetek a világűrbe

Az 1960-as évek környékén Ivan Bellnek [1] eszébe jutott, hogy érdekes lenne egy üzenetet az űrbe küldeni, annak céljából, hogy idegen civilizációk hírt kaphassanak rólunk. Ma is érdekes feladvány Bell egykori üzenetét megfejteni. A-tól Z-ig a betűk különböző rádiójeleket jelölnek, a pont és pontosvessző különböző hosszúságú szüneteket a rádiójelek között. Fejtsük meg az üzenetet.

1. A. B. C. D. E. F. G. H. I. J. K. L. M. N. P. Q. R. S. T. U. V. W. Y. Z.

2. AA, B; AAA, C; AAAA, D; AAAAA, E; AAAAAA, F; AAAAAAA, G; AAAAAAAA, H; AAAAAAAAA, I; AAAAAAAAAA, J.

3. AKALB; AKAKALC; AKAKAKALD. AKALB; BKALC; CKALD; DKALE. BKELG; GLEKB. FKDLJ; JLFKD.

4. CMALB; DMALC; IMGLB.

5. CKNLC; HKNLH. DMDLN; EMELN.

6. JLAN; JKALAA; JKBLAB; AAKALAB. JKJLBN; JKJKJLCN. FNKGLFG.

7. BPCLF; EPBLJ; FPJLFN.

8. FQBLC; JQBLE; FNQFLJ.

9. CRBLI; BRELCB.

10. JPJLJRBLSLANN; JPJPJLJRCLTLANNN. JPSLT; JPTLJRD.

11. AQJLU; UQJLAQSLV.

12. ULWA; UPBLWB; AWDMALWDLDPU. VLWNA; VPCLWNC. VQJLWNNA; VQSLWNNNA. JPEWFGHLEFWGH; SPEWFGHLEFGWH.

13. GIWIHYHN; TKCYT. ZYCWADAF.

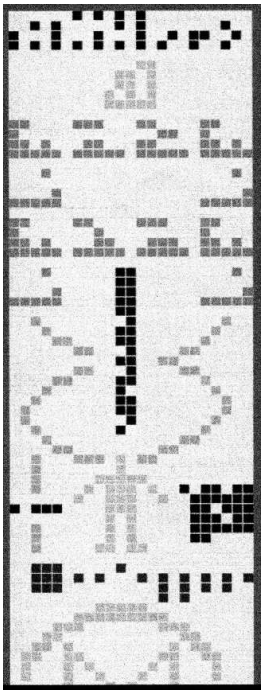
14. DPZPWNNIBRCQC.

A fenti üzenetet Bell 1960-ban, január 22-én a „The Japan Times" című újságban publikálta [1].

Bell üzenetével egy matematikai táborban találkoztam először [2]. Miután megfejtettük az üzenetet, kérdezgettük a tanárt, hogy kilőtték-e az üzenetet az űrbe. A kérdésre senki nem tudta a választ. Azon gondolkodtunk, hogy vajon bölcs dolog-e idegen civilizációknak üzenetet küldeni, hiszen nem biztos, hogy jóindulatúak. Most, 30 évvel az egykori tábor után, megpróbáltam az interneten megkeresni a választ, de erről semmit nem találtam. Viszont, találtam egy cikket, miszerint amerikai csillagászok Puerto-Rico szigetén található Arecibo csillagvizsgáló rádióadójával 1974-ben fellőttek egy üzenetet a világűrbe. Az üzenetet Dr. Frank Drake, Carl Sagan és társszerzőik írták, 7 részből állt, ami a következő fogalmakat kódolja:

1. A számok egy és tíz között.
2. A következő kémiai elemek rendszámai: hidrogén, szén, nitrogén, oxigén és foszfor – ezek a dezoxiribonukleinsav (DNS) alkotóelemei,
3. A DNS nukleotidjaiban található cukrok és bázisok képlete
4. A DNS-ben levő nukleotidok száma, és a DNS kettős spirál szerkezetének ábrázolása,
5. Az ember grafikus ábrázolása, egy átlagember testmagassága és a Föld lakossága, (akkor 4,5 milliárd)
6. A Föld Naprendszerének grafikája,
7. az Arecibo rádióteleszkóp grafikája és az adóantenna méretei.

Az üzenetet megfelelő méretű téglalapba rendezve, az a következőképpen nézett ki:



Az üzenet bővebb elemzése megtalálható a Wikipédia [3] oldalon. Sőt, ahogy tovább böngésztem a weben, néhány internetes oldalon azt is olvastam, hogy valakik megtalálhatták az üzenetet, mert az állítólag 2001 augusztusában visszajött. Állítólag, a dél-angliai Chilbolton csillagászati obszervatórium mellett találtak egy gabonakört, mely majdnem pontosan ugyanazt az ábrát tartalmazta mint a fenti üzenet. Az egyik különbség, hogy a földi elemek közé az idegenek bevették a szilíciumot is, jelezve, hogy náluk az a szerves élet alapja. Szerintem, ezt az információt azért érdemes fenntartásokkal kezelni.

Szomorú hír, hogy az Arecibo csillagvizsgálót lebontják, mert az épület veszélyesnek bizonyult.

Hivatkozások

[1] M. Gardner, *Mathematical games*, Scientific American 213 (2) (1965), 96-101.

[2] L. Pósa, Matematika tábor.

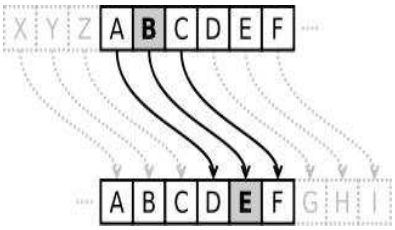
[3] Wikipédia, *Arecibói üzenet*, https://hu.wikipedia.org/wiki/Arecib%C3%B3i_%C3%BCzenet

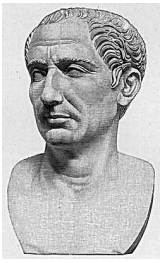
[4] Fotó, Wikipédia, https://en.wikipedia.org/wiki/Arecibo_message

2.2. Caesar-rejtjel

Ebben és a következő néhány alfejezetben pár régi titkosírásról lesz szó röviden, egészen az ókortól kezdve. Ezeknek a fejezeteknek az alapja az adott rejtjelekről szóló megfelelő Wikipédia oldalak, amelyeket kicsit lerövidítettem. A rejtjelek illusztrálására való példák is a Wikipédiáról valóak.

A Caesar-rejtjel valószínűleg az egyik legegyszerűbb és legszélesebb körben használt titkosítási módszer volt a maga korában, ugyanakkor a mai napig egyik legjobban ismert titkosírási módszer. Ez egy olyan helyettesítő rejtjel, amikor is minden egyes betűt az ábécében egy tőle meghatározott távolságra lévő betűvel helyettesítenek. Így példának okáért, ha mondjuk 3-mal toljuk el az ábécét, az angol ábécében az A-t a D-vel, a B-t az E-vel, C-t az F-fel stb. szükséges helyettesíteni. A magyar ábécére vonatkoztatva ez az A betű helyett C-t, az Á betű helyett CS-t jelent. A rejtjel az elnevezését Julius Caesar után kapta, aki ennek segítségével kommunikált tábornokaival.





Titkosítandó: ÉN ELMENTEM A VÁSÁRBA FÉL PÉNZZEL

Titkos szöveg: IŐ HÓÖHŐŰHÖ D ZEÜEÚÉD ÍIÓ TIŐCCHÓ

Bár nem tudjuk, mennyire volt hatékony ez a kód akkoriban, de azért úgy gondoljuk, hogy eléggé megbízható lehetett. Ezt megerősíti az a tény, hogy Caesar létfontosságú üzenetek titkosítására használta. Itt megemlítendő, hogy Caesar legtöbb ellenfele írástudatlan volt.

A rejtjel megfejtését először a 9. században említik, Al-Kindi gyakoriságelemzéshez kapcsolta [2]. Korábbi irodalmunk nincs a rejtjel feltöréséről, így elképzelhető, hogy Caesar rejtjelét is ekkoriban fejtették meg először.

Hivatkozások

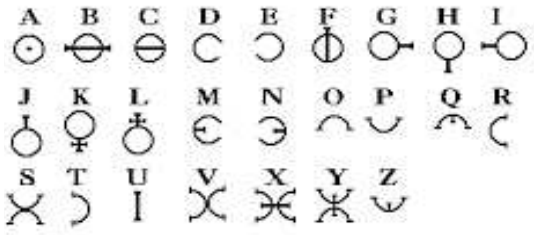
[1] Wikipédia, *Ceasar-rejtjel*, https://hu.wikipedia.org/wiki/Caesar-rejtjel.

[2] S. Singh, *Kódkönyv*, Park könyvkiadó, 2007.

[3] Fotók, Wikipédia, https://hu.wikipedia.org/wiki/Caesar-rejtjel.

2.3. Mono-alfabetikus rejtjel

A monoalfabetikus rejtjelezés során a szöveg betűihez különböző szimbólumokat rendelnek, ugyanahhoz a betűhöz mindig ugyanazt a szimbólumot.



Példa az egyszerű subpositio-ra:
Az [illegible] által használt titkosírás.

Ez a rejtjelezés gyakoriságelemzéssel azonban könnyen fejthető: pl. a magyar ábécében a leggyakoribb betű az E betű, a titkosított szövegben legtöbbször előforduló szimbólum az E betűnek felel meg. Hasonlóan megtalálható az ábécé második majd harmadik leggyakoribb betű kódja is, ez a magyar ábécében az A majd a T betű. Ezek után lehet még rövid szavacskákat is vizsgálni, pl. a magyar nyelvben nagyon gyakori az „AZ" szócska, vagyis az A betű kódját gyakran követi a Z betű kódja. Érdemes elolvasni a rejtjelezés talán legkorábbi és minden bizonnyal leghíresebb irodalmi megjelenését, Edgar Allan Poe, Aranybogár című novelláját [2], melyben lényegében egy monoalfabetikus rejtjel megfejtése történik. A rejtjelfejtés megjelenik még sok más helyen is, ezek közül csak kettőt említve: pl. Jules Verne, Sándor Mátyás [3] című regényében vagy Arthur Conan Doyle, Táncoló figurák [1] című novellájában is.

Hivatkozások

[1] A. C. Doyle, *Sherlock Holmes visszatér*, *Táncoló figurák*, Szukits Könyvkiadó, 2017.

[2] E. A. Poe, *Aranybogár*, http://vmek.oszk.hu/03500/03575/

[3] J. Verne, *Sándor Mátyás*, https://mek.oszk.hu/03200/03220/03220.pdf

[4] Wikipédia, *Helyettesítő rejtjel*, https://en.wikipedia.org/wiki/Substitution_cipher

[5] https://kripto.blog.hu/2014/09/29/az_elso_kripto-thriller_szerzo_edgar_allan_poe

[6] Ábra, *Példa az egyszerű subpositio-ra: Az alchymisták által használt titkosírás*, https://tanarbazar.blogger.hu/2017/04/30/kodjatszma

2.4. Vigenère-rejtjel

Ebben a fejezetben a történelem egyik legnevezetesebb titkosító módszeréről, az ún. Vigenère-rejtjelről lesz szó [11] alapján.

A Vigenére-kód vagy Vigenére-rejtjel egy olyan titkosítási módszer, amely különböző Caesar-kódokat használ, egy adott kulcsszó betűitől függ, hogy ezek közül a Caesar -rejtelek közül épp melyiket használja a kódolás. Tehát ez egy polialfabetikus kódolás.

Az első polialfabetikus kódolás alapos leírása és vizsgálata Leon Battista Albertitől származik 1467 körül. Alberti különböző Caesar-kódokat alkalmazott, ahol az adott Caesar-kódot néhány szó után egy másik Caesar-kódra cserélte le.

Később John Trithemius létrehozott egy olyan titkosító eljárást, amely már használta a Viegnére-kód legfőbb alkotóelemét, a Viegnére táblát. Blaise de Vigenére azonban később 1586-ban tette közzé a polialfabetikus titkosítását, ahol a kulcsszavak az eredeti szövegen alapultak (autókulcsolásos rejtjel). Ezt a titkosítást később Vigenére titkosításnak nevezték.

Valójában azonban amit Vigenére-rejtjelnek hívunk ma, azt Vigenére kódja előtt több mint 30 évvel találták fel. Ezt a kódot Giovan Battista Bellaso írta le először 1553-ban, La cifra del. Sig. Giovan Batista Belaso című könyvében.

A kód jól ismert, mert könnyen érthető és használható, de hosszú ideig, több évszázadig törhetetlen bizonyult; ezért franciául „le chiffre indéchiffrable" („feltörhetetlen kód") elnevezés ragadt rá.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Kódolás és dekódolás

A kódolandó szöveg mellett a titkosításhoz egy titkos kulcsra is szükségünk lesz, melynek hossza k. Ekkor a kódolandó szöveget (más szóval nyílt szöveget) k hosszú részekre osztjuk, és minden rész alá írjuk magát a titkos kulcsot. Készítünk egy táblázatot, ez lesz a Vigenére táblázat, amelyben az összes Caesar-kód szerepel, minden sorban pontosan eggyel eltolva szerepel az előző Caesar-kód. Ezután a nyílt szöveget, azzal a Caesar-kóddal rejtjelezzük, amelyiknek a táblázatban a kezdőbetűje megfelel a nyílt szöveg alatt lévő kulcsszó megfelelő betűjének.

Ezt a kódolást legjobban egy példával szemléltethetjük, amely példa a Wikipédia [11] oldalon is szerepel.

Kódolási példa

Legyen a nyílt szöveg a **Budapest felett az ég felhőtlen**, a kulcs pedig **Lojzi**. Ekkor nem szükséges az összes Caesar kód-ábécét felsorolni, csak a kódszó betűivel kezdődőket (és természetesen a táblázat elején az eredeti ábécét):

AÁBCDEÉFGHIÍJKLMNOÓÖŐPQRSTUÚÜŰVWXYZ

IÍJKLMNOÓÖŐPQRSTUÚÜŰVWXYZAÁBCDEÉFGH

JKLMNOÓÖŐPQRSTUÚÜŰVWXYZAÁBCDEÉFGHIÍ

LMNOÓÖŐPQRSTUÚÜŰVWXYZAÁBCDEÉFGHIÍJK

OÓÖŐPQRSTUÚÜŰVWXYZAÁBCDEÉFGHIÍJKLMN

ZAÁBCDEÉFGHIÍJKLMNOÓÖŐPQRSTUÚÜŰVWXY

A kódolás ekkor:

Nyílt szöveg: BUDAPEST FELETT AZ ÉG FELHŐTLEN

Kulcs: LOJZILOJ ZILOJZ IL OJ ILOJZILO

Kódolt szöveg: NGNZWÖÉB ÉMÜQBS IK RŐ GMÜUXSSÖY

A dekódolás hasonlóan történik, a kulcs sorában megkeressük a használt karaktert, és az azonos oszlopban szereplő betűjét írjuk az első sorban szereplő ábécéből.

Feltörése

A Vigenére kódot amiatt, hogy ugyanaz a betű sokféleképpen is kódolható, és ez rengeteg-féle variációs lehetőséghez vezethet, sokáig feltörhetetlennek és abszolút biztonságosnak gondolták. Azonban 1854-ben Charles Babbagenak sikerült feltörnie a kódot, de módszerét soha nem írta le. A

Vigenére kód első sikeres feltörési módját Friedreich Kaisiki publikálta. Észrevette ugyanis, hogy a kulcsnál lényegesen hosszabb szöveg esetén ismétlődések lesznek a kódolt szövegben. Két ismétlődés közötti hossz legtöbbször szükségszerűen a kulcsszó hosszának többszöröse, így ezeknek a hosszoknak a legnagyobb közös osztóját véve megkapjuk a kulcs hosszát: k-t, vagy annak párszorosát. Ekkora darabokra bontva a szöveget a visszafejtés egyszerűvé válik, gyakoriságelemzést alkalmazhatunk:

A kódolt szöveget ugyanis a betűk helye szerint k csoportba osztjuk. Az első csoportba az 1., $(k+1)$-edik, $(2k+1)$-edik, stb., a másodikba a 2., $(k+2)$-edik, $2k+2$-edik stb. karakterek kerülnek. Ezek után a csoportokra gyakoriságelemzést végzünk, ezzel egyszerre megkapjuk a szöveg és a kulcs betűit.

Hozzávetőlegesen, egy gyakorlott rejtjelfejtőnek nem túl hosszú kulcs esetén mintegy 6-9 órába telik megfejteni a kódolt szöveget.

Hivatkozások

[1] Bellaso, Giovan Battista, *La Cifra del Sig. Giovan Battista Belaso* (olaszul), Velence, (Olaszország), 1553. Elérhető: Museo Galileo (Florence (Firenze), Olaszország)

[2] A. A. Bruen, M. A. Forcinito, *Cryptography, Information Theory, and Error-Correction: A Handbook for the 21st Century*, John Wiley & Sons, 2011, p. 21.

[3] M. Gamer, *Die Polygraphia des Johannes Trithemius. Zwei Fassungen eines frühneuzeitlichen Handbuchs zur Geheimschrift*, megtalálható: Baier, Schultheiß, Jochen (szerk.). Würzburger Humanismus (németül), Tübingen, Germany: Narr Verlag, 2015, 121–141.

[4] F. W. Kasiski, *Die Geheimschriften und die Dechiffrir-Kunst* (németül), Berlin, (Németország): E.S. Mittler und Sohn.

[5] Laurence D. Smith, *Cryptography: The Science of Secret Writing*, Courier Corporation, 1955, o. 81.

[6] Keith M. Martin, *Everyday Cryptography*, Oxford University Press, 2012, o. 142.

[7] Megyesi Z., *Titkosírások*, Kisújszállás, Szalay könyvkiadó, 1999.

[8] D. Rodriguez-Clark, *Vigenère Cipher*, Crypto Corner, 2017, https://crypto.interactive-maths.com/vigenegravere-cipher.html

[9] J. Trithemius, *Liber Quintus Exordium Capit*, (5. könyv, 1. fejezet), Polygraphiae, libri sex (latinul), Reichenau, (Németország), 1518, Elérhető: George Fabyan Collection (Library of Congress; Washington, D.C., U.S.A.).

[10] B. de Vigenère, *Traicté des Chiffres, ou Secretes Manieres d'Escrire* (franciául), Párizs, Franciaország, Abel l'Angelier, 1586.

[11] Wikipédia, *Vigenére-rejtjel*, https://hu.wikipedia.org/wiki/Vigen%C3%A8re-rejtjel.

[12] Fotók, Wikipédia, https://hu.wikipedia.org/wiki/Vigen%C3%A8re-rejtjel.

2.5. Vernam féle titkosító eljárás

Pszeudovéletlen és véletlen sorozatok kriptográfiai alkalmazásai közül máig a legelterjedtebb az un. **Vernam féle titkosító eljárás**. Tegyük fel, hogy titkosítani szeretnénk egy szöveget. Ekkor minden betűhöz hozzárendelünk egy 0,1 sorozatot. Például:

A: 000001	B: 000010	C: 000011	D: 000101	E: 000110	É: 000111
F: 001000	G: 001001	H: 001010	I: 001011	Í: 001100	J: 001101
K: 001110	L: 001111	M: 010000	N: 010001	O: 010011	Ó: 010100
Ö: 010101	Ő: 010110	P: 010111	Q: 011000	R: 011001	S: 011010
T: 011011	U: 011100	Ú: 011101	Ü: 011110	Ű: 011111	V: 100000
X: 100001	Y: 100010	Z: 110011			

Az ily módon kódolt szöveget könnyű visszafejteni betűgyakoriság elemzéssel. (Például a magyar nyelvben az E betű a leggyakoribb, így a kódolt szövegben a 000110 fog előfordulni leggyakrabban.)

Ezért a kódolt szöveget úgy kódoljuk tovább, hogy bitenként összeadjuk egy pszeudovéletlen sorozattal, ahol az összeadás a modulo 2 összeadás. Az így kapott kódolási eljárás a **Vernam féle titkosító eljárás:**

$$\begin{aligned} \text{Üzenet} &: (a_1, \ldots, a_N) \in \{0,1\}^N \\ \oplus\ \text{Titkos kulcs} &: \underline{(e_1, \ldots, e_N)} \in \{0,1\}^N \\ \text{Kódolt üzenet} &: (f_1, \ldots, f_N) \in \{0,1\}^N. \end{aligned}$$

Összeadási szabály:

$$0 \oplus 0 = 0, \quad 1 \oplus 1 = 0,$$
$$0 \oplus 1 = 1, \quad 1 \oplus 0 = 1.$$

Ezt az eljárást Vernam a XX. század elején találta ki. Fontos, hogy egy kulcsot csak egy üzenet titkosításához használhatunk: ismételt alkalmazás esetén feltörhető az eljárás. Ha a kulcs valódi véletlen sorozat, akkor az eljárást **egyszer használatos kulcsnak**, angol nyelven pedig **one-time pad**-nek nevezzük. Ebben az esetben a titkosítás során az üzenet minden bitje (egymástól függetlenül) azonos valószínűséggel változik meg illetve marad ugyanaz. Ezért ekkor ez a titkosítási mód tökéletes biztonságot ad. Ezt a módszert az I. világháború idejében sokszor használták, és biztonsága miatt, ma is egyike a legmegbízhatóbb titkosítási módszereknek. A Vernam

féle titkosító eljárás egyetlen hátránya, hogy a titkos kulcsnak ugyanolyan hosszúnak kell lennie, mint az üzenetnek. Itt gondot jelenthet a titkos kulcs eljuttatása a kommunikáló feleknek. Erről bővebben a Diffie-Hellman kulcscseréről szóló fejezetben lesz szó.

Ezt ma már úgy oldják meg, hogy egy kisebb titkos kulcsból számítógépek segítségével generálnak egy véletlent imitáló (elegendően) hosszú un. pszeudovéletlen sorozatot.

Ennél bővebben a pszeudovéletlen sorozatokról a 4. fejezetben fogok írni.

Hivatkozások

[1] Wikipédia, *Gilbert Vernam*, https://en.wikipedia.org/wiki/Gilbert_Vernam

[2] Fotó, *Gilbert Vernam*, https://en.wikipedia.org/wiki/File:Gilbert_Vernam.jpg

[3] Ábra, *Számítógép*, http://azcoloriage.com/coloriage/31045

2.6. Nyilvános kulcsú rejtjelezés

A Vernam féle titkosító eljárásnál láttuk, hogy mind a kódolást, mind a dekódolást végző személy ugyanazt a titkos kulcsot használja. Az ilyen rejtjelezéseket szimmetrikus kulcsú titkosításnak nevezzük. Van azonban egy

másik fajtája a rejtjelezésnek, amikor is más-más kulcsot használ a kódoló és dekódoló fél. Előfordulhat például, hogy azt szeretnénk, hogy egy személynek vagy intézménynek bárki írhasson titkosított levelet. Ehhez közzétesznek egy nyilvános kulcsot, amelyet bárki ismerhet, és amellyel bárki küldhet titkosított üzenetet a fogadó félnek. A dekódolás azonban már egy másik titkos kulccsal történik, amelyet természetesen csak az ismer, akinek az üzenetet szánjuk, hiszen fontos, hogy csak ő olvashassa el a titkosított üzenetet. Erre kiváló példa az RSA titkosító eljárás, amelyről bővebben a 9. fejezetben olvashatunk, vagy a Diffie-Hellman kulcscsere, mely a 11. fejezetünk témája lesz. Gondolhatunk még a digitális aláírás problémájára is (ld. pl. 12.4 fejezet), ahol nagyon fontos, hogy aláírni mindenki tudjon, ugyanakkor hamisítás ne fordulhasson elő. Ezekben az esetekben a szokásos szimmetrikus kulcsú titkosítás csődöt mond, ugyanakkor az aszimmetrikus kulcsú titkosítással a fenti problémák hatékonyan megoldhatóak.

Hivatkozások

[1] Wikipédia, Nyilvános kulcsú rejtjelezés, https://hu.wikipedia.org/wiki/Nyilv%C3%A1nos_kulcs%C3%BA_rejtjelez%C3%A9s

Már most látható, hogy a számítógépes számelmélet erősen kapcsolódik több területhez is, a számelmélethez, kriptográfiához, algoritmusokhoz. Szerencsénkre ezekhez a területekhez jelentős magyar nyelvű irodalom is elérhető. Az alábbiakban ezek közül csak néhányat említek, amelyek biztosan kellemes időtöltés biztosítanak az olvasóknak:

Buttyán L., Vajda I., *Kriptográfia és alkalmazásai*, Budapest, Typotex Elektronikus Kiadó Kft, 2004.

Freud R., Gyarmati E., *Számelmélet*, Nemzedékek Tudása Tankönyvkiadó, 2006.

Gács P., Lovász L., *Algoritmusok*, Tankönyvkiadó 1989.

Gyarmati E., Turán P., *Számelmélet*, Tankönyvkiadó, 1969.

Györfi L., Győri, S., Vajda, I., *Információ- és kódelmélet*, Typotex Kiadó, 2000.

D. E. Knuth, *A Számítógép-Programozás Művészete (1. és 2. kötet) - Alapvető Algoritmusok, Szeminumerikus Algoritmusok*, Műszaki Könyvkiadó, Budapest, 1994.

Sárközy A., *Számelmélet és Alkalmazásai*, Műszaki Könyvkiadó, 1987.

3. Számelméleti Alapok

Ebben a fejezetben a kurzushoz szükséges számelméleti alapokat ismertetem. A fejezetben szereplő tételeket nem bizonyítjuk, de azok bizonyítása a legtöbb elemi számelmélettel foglalkozó könyvben megtalálható.

3.1. Kongruenciák

3.1. Definíció. *Azt mondjuk a kongruens b modulo m, ha az a és b egész számoknak az m pozitív egész számmal vett osztási maradéka ugyanaz. Más szóval:* $m \mid a - b$. *Jelölése:*

$$a \equiv b \pmod{m}.$$

Például: $15 \equiv 27 \pmod{12}$, de $3 \not\equiv 14 \pmod{12}$.

3.2. Tétel. *Ha* $a \equiv x \pmod{m}$ *és* $b \equiv y \pmod{m}$, *akkor*

$$a + b \equiv x + y \pmod{m} \quad \text{és} \quad ab \equiv xy \pmod{m}.$$

Az osztásra a következő szabály áll fenn:

3.3. Tétel. *Legyenek* a, b, c *egész számok* m *pedig pozitív egész szám. Ekkor*

$$ac \equiv bc \pmod{m}$$

esetén a kongruenciát c-vel osztva azt kapjuk, hogy

$$a \equiv b \pmod{\frac{m}{(c, m)}},$$

vagyis a modulust is osztanunk kell, mégpedig c és m legnagyobb közös osztójával.

Az elemi számelmélet egyik legnevezetesebb tétele az Euler-Fermat tétel. Euler 1736-ban publikálta a tételt, melyben a saját bizonyítását ismertette a Fermat tételre. Mielőtt a tételt ismertetném, szükséges az ún. Euler-féle φ-függvény definíciója.

3.4. Definíció. *Minden n pozitív egész számra jelölje $\varphi(n)$ azon pozitív egész számoknak a számát, amelyek n-hez relatív prímek és n-nél nem nagyobbak. Képlettel:*

$$\varphi(n) = |\{r : 1 \leq r \leq n \text{ és } (r,n) = 1\}|.$$

Ez a φ függvény multiplikatív, azaz ha a és b pozitív egész számok egymáshoz relatív prímek, akkor

$$\varphi(ab) = \varphi(a)\varphi(b).$$

Mint minden multiplikatív függvényre, a φ-re is fennáll a $\varphi(1) = 1$ összefüggés. Ha pedig $n > 1$ és prímtényezős felbontása $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_r^{\alpha_r}$, akkor

$$\varphi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_r}\right).$$

A φ függvény ismeretében, már kimondhatjuk a nevezetes Euler-Fermat tételt:

3.5. Tétel. (Euler-Fermat) *Ha a egész szám, m pedig a-hoz relatív prím pozitív egész, akkor*

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

A tételből nagyon egyszerűen következik a kis-Fermat tétel, melyet Fermat 100 évvel Euler előtt fedezett fel, 1636-ban, és amelyet 1640-ben bizonyítás nélkül ismertetett. A kis-Fermat tétel [1] az Euler-Fermat tétel speciális esete,

[1] Fermat sejtésként megfogalmazta, hogy az $x^n + y^n = z^n$ egyenletnek nincs pozitív egészekből álló megoldása ha $3 \leq n \in \mathbb{N}$. A sejtés évszázadokig nyitott volt. Végül Andrew Wiles bizonyította be 1994-ben mély számelméleti eszközöket használva. Azóta ezt a tételt nagy-Fermat tételként nevezzük.

akkor amikor az m modulus prímszám. A tételnek a következő két alakja ismert:

3.6. Tétel. (kis-Fermat) *Ha p prím és az a egész számra $(a, p) = 1$, akkor*

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

3.7. Tétel. (kis-Fermat) *Ha p prím akkor minden a egész számra:*

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

Fontos fogalom még a számelméletben a rend is:

3.8. Definíció. *Legyen m természetes szám, és a olyan egész szám, amelyre $(a, m) = 1$. Az a rendje modulo m, az a legkisebb pozitív egész r, amelyre*

$$a^r \equiv 1 \pmod{m}.$$

Jelölése $o_m(a)$.

A rend alaptulajdonságai a következők:

3.9. Tétel. *Legyen m természetes szám, a olyan egész szám, amelyre $(a, m) = 1$, és $x,\ y$ is természetes számok, ekkor, ha*

$$a^x \equiv a^y \pmod{m},$$

akkor

$$x \equiv y \pmod{o_m(a)}.$$

Ennek következménye az alábbi ($y = 0$-t véve):

3.10. Tétel. *Legyen m és $x$természetes szám, a egész szám, amelyre $(a, m) = 1$. Ekkor*

$$a^x \equiv 1 \pmod{m},$$

akkor és csak akkor teljesül, ha

$$o_m(a) \mid x.$$

Ebből és az Euler-Fermat tételből adódóan:

3.11. Következmény. *Legyen m természetes szám, $(a, m) = 1$ egész szám, ekkor*

$$o_m(a) \mid \varphi(m),$$

Fontos fogalom a primitív gyök is, melynek definíciója:

3.12. Definíció. *Legyen m természetes szám, g egész szám, ekkor g primitív gyök modulo m, ha*

$$o_m(g) = \varphi(m).$$

Primitív gyökök kapcsán a következő két tételt igen gyakran használjuk:

3.13. Tétel. *A g egész szám akkor és csak akkor primitív gyök modulo m, ha az $1, g, g^2, \ldots, g^{\varphi(m)-1}$ halmaz kiadja azokat és csak azokat a mod m maradékosztályokat amelyek relatív prímek m-hez, és mindegyiket pontosan egyszer.*

3.14. Tétel. *Az m modulushoz, akkor és csak akkor létezik primitív gyök, ha $m = 2, 4$ vagy $m = p^\alpha, 2p^\alpha$ alakú szám, ahol p páratlan prím, α pedig természetes szám.*

Végezetül az egyik legrégebbi számelmélet tételt, a több mint 2000 éves kínai maradéktételt ismertetjük, melynek legegyszerűbb formája a következő:

3.15. Tétel. (Kínai maradéktétel) *Ha $k \in \mathbb{N}$, $m_1, \ldots, m_k \in \mathbb{N}$ páronként relatív prímek, akkor az*

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ &\vdots \\ x &\equiv a_k \pmod{m_k} \end{aligned}$$

lineáris kongruencia rendszer megoldható, és a megoldások egy maradékosztályt alkotnak $\bmod m_1 \ldots m_k$.

Hivatkozások

[1] Freud R., Gyarmati E., *Számelmélet*, Nemzedékek Tudása Tankönyvkiadó, 2006.

[2] Gyarmati E., Turán P., *Számelmélet*, Tankönyvkiadó, 1969.

[3] G. Hardy, E. M. Wright, *An Introduction to the Theory of Numbers*, Oxford University Press, 2008, sixth edition.

3.2. Legendre szimbólum

Feladat: Bizonyítsuk be, hogy nincs olyan négyzetszám, amelynek a hármas maradéka 2.

Megoldás: Jelölje x^2 a szóban forgó négyzetszámot. 3 különböző esetet vizsgálunk az x egész szám hármas maradéka szerint:

$$\begin{aligned} x &\equiv 0 \pmod 3 \Rightarrow x^2 \equiv 0^2 = 0 \pmod 3 \\ x &\equiv 1 \pmod 3 \Rightarrow x^2 \equiv 1^2 = 1 \pmod 3 \\ x &\equiv 2 \pmod 3 \Rightarrow x^2 \equiv 2^2 = 4 \equiv 1 \pmod 3. \end{aligned}$$

Azaz a négyzetszámok 0-val vagy 1-gyel kongruensek modulo 3. Mi a helyzet nagyobb prímekre?

Legyen $p = 11$. Ekkor
$$\begin{aligned} x &\equiv 0 \pmod{11} \Rightarrow x^2 \equiv 0^2 = 0 \pmod{11} \\ x &\equiv 1 \pmod{11} \Rightarrow x^2 \equiv 1^2 = 1 \pmod{11} \\ x &\equiv 2 \pmod{11} \Rightarrow x^2 \equiv 2^2 = 4 \pmod{11} \\ x &\equiv 3 \pmod{11} \Rightarrow x^2 \equiv 3^2 = 9 \pmod{11} \\ x &\equiv 4 \pmod{11} \Rightarrow x^2 \equiv 16 \equiv 5 \pmod{11} \\ x &\equiv 5 \pmod{11} \Rightarrow x^2 \equiv 25 \equiv 3 \pmod{11} \end{aligned}$$

$$x \equiv 6 \pmod{11} \Rightarrow x^2 \equiv 36 \equiv 3 \pmod{11}$$
$$x \equiv 7 \pmod{11} \Rightarrow x^2 \equiv 49 \equiv 5 \pmod{11}$$
$$x \equiv 8 \pmod{11} \Rightarrow x^2 \equiv 64 \equiv 9 \pmod{11}$$
$$x \equiv 9 \pmod{11} \Rightarrow x^2 \equiv 81 \equiv 4 \pmod{11}$$
$$x \equiv 10 \pmod{11} \Rightarrow x^2 \equiv 100 \equiv 1 \pmod{11}.$$

Azaz x^2 a 0,1,3,4,5 és 9 értékeket veheti fel modulo 11.

Azaz 1,3,4,5 és 9 a **kvadratikus maradékok** modulo 11.

Míg 2,6,7,8 és 10 a **kvadratikus nem-maradékok** modulo 11.

Általánosabban, legyen p prímszám és $(a,p)=1$. Ekkor a **kvadratikus maradék modulo** p ha az

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

kongruencia megoldható és a **kvadratikus nem-maradék modulo** p ha az

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

kongruencia nem oldható meg.

Legyen továbbra is $(a,p)=1$. A **Legendre szimbólumot** $\left(\frac{a}{p}\right)$ következőképp definiáljuk:

$$\left(\frac{a}{p}\right) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{ha } a \text{ kvadratikus maradék modulo } p \quad \Leftrightarrow \quad x^2 \equiv a \pmod{p} \text{ megoldható} \\ -1 & \text{ha } a \text{ kvadratikus nem-maradék modulo } p \quad \Leftrightarrow \quad x^2 \equiv a \pmod{p} \text{ nem oldható meg} \end{cases}$$

Illusztráljuk a $p=11$ esetre vonatkozó eredményeinket egy táblázattal:

	1	2	3	4	5	6	7	8	9	10
Kvadratikus maradék modulo 11?	igen	nem	igen	igen	igen	nem	nem	nem	igen	nem

Ez az eloszlás véletlennek tűnik...

A kvadratikus maradékok fogalmára alapozva, definiálhatunk **bináris pszeudovéletlen sorozatokat**:

	1	2	3	4	5	6	7	8	9	10
Kvadratikus maradék modulo 11?	1	0	1	1	1	0	0	0	1	0

A kvadratikus maradékok száma $\frac{p-1}{2}$: Tekintsük a következő számokat $1^2, 2^2, 3^2, \ldots, (p-1)^2$ modulo p. Ebben a sorozatban

$$x^2 \equiv y^2 \pmod{p}$$

akkor és csak akkor áll fenn, ha

$$\begin{aligned} & p \mid x^2 - y^2 \\ & p \mid (x-y)(x+y) \\ & p \mid x-y \text{ vagy } p \mid x+y \\ & x \equiv \pm y \pmod{p} \\ & x = y \text{ vagy } p-y \end{aligned}$$

Azaz az $1^2, 2^2, 3^2, \ldots, (p-1)^2$ sorozat $\frac{p-1}{2}$ darab különböző elemet tartalmaz modulo p. **Vagyis a kvadratikus maradékok száma: $\frac{p-1}{2}$.** Ebből adódóan **a kvadratikus nem-maradékok száma $\frac{p-1}{2}$**. Régebbi koncepció szerint a 0 egyik halmazba sem tartozik, most mi is ezt követjük. Megjegyezzük azonban, hogy újabban a 0-t is kvadratikus maradékként definiálják.

A fenti egyszerű tény motiválta, hogy a Legendre szimbólum jól alkalmazható pszeudovéletlen objektumok konstruálása során.

A Legendre szimbólum értéke gyorsan számolható (kiterjesztését: Jacobi

szimbólumot használva.)

3.16. Tétel. *A Legendre-szimbólum alap tulajdonságai:*

(a) $(a,p)=(b,p)=1$, $a\equiv b\ (p) \Rightarrow \left(\frac{a}{p}\right)=\left(\frac{b}{p}\right)$.

(b) $(a,p)=1 \Rightarrow \left(\frac{a^2}{p}\right)=1$, *spec.:* $\left(\frac{1}{p}\right)=1$.

(c) $(a,p)=(b,p)=1 \Rightarrow \left(\frac{ab}{p}\right)=\left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$.

(d) *Euler-lemma:*

$$(a,p)=1 \quad \textit{esetén}$$
$$\left(\frac{a}{p}\right)\equiv a^{\frac{p-1}{2}} \pmod{p}.$$

(e) $$\left(\frac{-1}{p}\right)=(-1)^{\frac{p-1}{2}}=\begin{cases}+1, & \textit{ha } p=4k+1 \quad \text{alakú prím},\\ -1, & \textit{ha } p=4k-1 \quad \text{alakú prím}.\end{cases}$$

(f) $$\left(\frac{2}{p}\right)=(-1)^{\frac{p^2-1}{8}}=\begin{cases}1, & \textit{ha } p=8k\pm 1 \quad \text{alakú prím},\\ -1, & \textit{ha } p=8k\pm 3 \quad \text{alakú prím}.\end{cases}$$

(g) *Gauss kvadratikus reciprocitási tétele: Ha p,q páratlan prímek, akkor*

$$\left(\frac{p}{q}\right)=(-1)^{\frac{p-1}{2}\cdot\frac{q-1}{2}}\left(\frac{q}{p}\right).$$

Az alap tulajdonságok alapján a Legendre szimbólum könnyen számolható. Lássunk erre egy példát:

$$\begin{aligned}\left(\frac{12345}{331}\right)&=\left(\frac{3}{331}\right)\left(\frac{5}{331}\right)\left(\frac{823}{331}\right)\\ &=\left(\frac{3}{331}\right)\left(\frac{5}{331}\right)\left(\frac{161}{331}\right)\\ &=\left(\frac{3}{331}\right)\left(\frac{5}{331}\right)\left(\frac{7}{331}\right)\left(\frac{23}{331}\right)\end{aligned}$$

$$
\begin{aligned}
&= (-1)\left(\frac{331}{3}\right)\left(\frac{331}{5}\right)(-1)\left(\frac{331}{7}\right)(-1)\left(\frac{331}{23}\right)\\
&= -\left(\frac{1}{3}\right)\left(\frac{1}{5}\right)\left(\frac{2}{7}\right)\left(\frac{9}{23}\right)\\
&= -\left(\frac{1}{3}\right)\left(\frac{1}{5}\right)\left(\frac{2}{7}\right)\left(\frac{3^2}{23}\right)\\
&= -1\cdot 1\cdot 1\cdot 1\\
&= -1.
\end{aligned}
$$

Ebben az algoritmusban a lépések száma $O(\log p)$, igen ám, de néhány lépés során faktorizálni kell, ami nagyon időigényes. Baj: A faktorizáció nagyon időigényes!

Mielőtt továbbhaladnánk egy gyors megjegyzés következik. Az Edmund Landautól származó nagy ordó-jelölés a következő értelemben használatos: $f(x) = O(g(x))$ azt jelenti, hogy $|f(x)| \leq Cg(x)$, teljesül alkalmas C valós konstansra az értelmezési tartomány szóban forgó helyein. Ezzel ekvivalens jelölés: $f(x) \ll g(x)$. Amennyiben $f(x)/g(x) \to 0$ is teljesül, azt $f(x) = o(g(x))$-szel jelöljük (ez a kis ordó-jelölés).

Fontos kérdés: Hogyan lehet lehetőleg faktorizációs lépések nélkül, gyorsan meghatározni $\left(\frac{a}{p}\right)$-t?

Ehhez bevezetjük az ún. Jacobi-szimbólumot.

3.17. Definíció. *Ha $n \in \mathbb{N}$, $n > 1$, n páratlan, $a \in \mathbb{Z}$ $(a, n) = 1$, akkor $\left(\frac{a}{n}\right)$ Jacobi-szimbólum definíciója: ha n faktorizációja az $n = {p_1}^{\alpha_1} \dots {p_r}^{\alpha_r}$, akkor*

$$\left(\frac{a}{n}\right) \stackrel{\text{def}}{=} \left(\frac{a}{p_1}\right)^{\alpha_1} \dots \left(\frac{a}{p_r}\right)^{\alpha_r}.$$

A fenti definícióban a baloldalon a definiálandó $\left(\frac{a}{n}\right)$ Jacobi szimbólum szerepel, míg a jobboldalon Legendre szimbólumok szorzata szerepel.

Figyelem!!! Ha n összetett szám, úgy $\left(\frac{a}{n}\right)$-nek semmi köze az $x^2 \equiv a \pmod{n}$ kongruencia megoldhatóságához (ellentétben a Legendre-szimbólummal). Viszont, ha n páratlan prím, akkor a Legendre és Jacobi szimbólum definíciója egybeesik.

A Jacobi szimbólum ugyanúgy teljesen multiplikatív mint a Legendre szimbólum, illetve a $\left(\frac{-1}{n}\right)$, $\left(\frac{2}{n}\right)$-re vonatkozó tétel és Gauss kvadratikus reciprocitási tétele átvihető a Jacobi-szimbólumra.

3.18. Tétel. *Ha $n \in \mathbb{N}$ páratlan, akkor*

a)

$$\left(\frac{-1}{n}\right) = (-1)^{(n-1)/2} = \begin{cases} 1, & \text{ha } n = 4k+1 \quad \text{alakú egész szám,} \\ -1, & \text{ha } n = 4k+3 \quad \text{alakú egész szám.} \end{cases}$$

b)

$$\left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8} = \begin{cases} 1, & \text{ha } n = 8k \pm 1 \quad \text{alakú egész szám,} \\ -1, & \text{ha } n = 8k \pm 3 \quad \text{alakú egész szám.} \end{cases}$$

3.19. Tétel. *Ha $m, n \in \mathbb{N}$ páratlan számok és $(m, n) = 1$, akkor*

$$\left(\frac{m}{n}\right) = (-1)^{\frac{m-1}{2} \cdot \frac{n-1}{2}} \left(\frac{n}{m}\right).$$

3.18. Tétel bizonyítása. Legyen $n = {p_1}^{\alpha_1} \dots {p_r}^{\alpha_r}$. Ekkor

$$\begin{aligned} \left(\frac{-1}{n}\right) &= \left(\frac{-1}{p_1}\right)^{\alpha_1} \dots \left(\frac{-1}{p_r}\right)^{\alpha_r} \\ &= \left((-1)^{\frac{p_1-1}{2}}\right)^{\alpha_1} \dots \left((-1)^{\frac{p_r-1}{2}}\right)^{\alpha_r} \\ &= (-1)^{\alpha_1 \cdot \frac{p_1-1}{2} + \dots + \alpha_r \frac{p_r-1}{2}}. \end{aligned}$$

Ez pontosan akkor $+1$, ha a $\sum_{p_i \equiv 3 \pmod 4} \alpha_i$ páros, ami viszont ekvivalens azzal, hogy n egy $4k+1$ alakú természetes szám. Valamint

$$\begin{aligned}\left(\frac{2}{n}\right) &= \left(\frac{2}{p_1}\right)^{\alpha_1} \dots \left(\frac{2}{p_r}\right)^{\alpha_r} \\ &= \left((-1)^{\frac{{p_1}^2-1}{8}}\right)^{\alpha_1} \dots \left((-1)^{\frac{{p_r}^2-1}{8}}\right)^{\alpha_r} \\ &= (-1)^{\alpha_1 \cdot \frac{{p_1}^2-1}{8} + \dots + \alpha_r \frac{{p_r}^2-1}{8}}.\end{aligned}$$

Az, hogy ez a kifejezés -1 vagy $+1$, attól függ, hogy $\sum \alpha_i \frac{{p_i}^2-1}{8}$ páros-e vagy páratlan. Ha bebizonyítjuk, hogy

$$\sum \alpha_i \frac{{p_i}^2-1}{8} \equiv \frac{n^2-1}{8} \pmod 2, \tag{3.1}$$

akkor készen vagyunk. Ehhez

$$\frac{p^2-1}{8} \equiv \begin{cases} 1 \pmod 2, & \text{ha } p \equiv \pm 3 \ (8), \\ 0 \pmod 2, & \text{ha } p \equiv \pm 1 \ (8). \end{cases}$$

Ez alapján

$$\begin{aligned}\sum \alpha_i \frac{{p_i}^2-1}{8} &\equiv \sum_{\alpha_i \text{ páratlan}} \frac{{p_i}^2-1}{8} \pmod 2, \\ &\equiv \sum_{p_i \equiv \pm 3 \ (8),\ \alpha_i \text{ páratlan}} 1 \pmod 2. \end{aligned} \tag{3.2}$$

Másrészt

$$\frac{n^2-1}{8} = \frac{{p_1}^{2\alpha_i} \dots {p_r}^{2\alpha_r} - 1}{8}.$$

Ekkor $p_1^{2\alpha_1} \cdots p_r^{2\alpha_r}$ 16-os maradékát kell vizsgálnunk. Tudjuk:

$$p^2 \equiv \begin{cases} 1 \pmod{16}, & \text{ha } p \equiv \pm 1 \ (8), \\ 9 \pmod{16}, & \text{ha } p \equiv \pm 3 \ (8), \end{cases}$$

így

$${p_i}^{2\alpha_i} \equiv \begin{cases} 1 \pmod{16}, & \text{ha } \alpha_i \text{ páros vagy } p \equiv \pm 1 \ (8), \\ 9 \pmod{16} & \text{különben.} \end{cases}$$

$$p_1{}^{2\alpha_1}\dots p_r{}^{2\alpha_r} \equiv 9^{\sum\limits_{p_i\equiv\pm3\ (8),\ \alpha_i\ \text{páratlan}} 1}$$

$$\equiv \begin{cases} 1, & \text{ha} \sum\limits_{p_i\equiv\pm3\ (8),\ \alpha_i\ \text{páratlan}} 1 \text{ páros}, \\ 9, & \text{ha} \sum\limits_{p_i\equiv\pm3\ (8),\ \alpha_i\ \text{páratlan}} 1 \text{ páratlan}\ . \end{cases} \pmod{16}$$

Ezek alapján:

$$\frac{p_1{}^{2\alpha_1}\dots p_r{}^{2\alpha_r}-1}{8} = \begin{cases} \text{páros}, & \text{ha} \sum\limits_{p_i\equiv\pm3\ (8),\ \alpha_i\ \text{páratlan}} 1 \text{ páros}, \\ \text{páratlan}, & \text{ha} \sum\limits_{p_i\equiv\pm3\ (8),\ \alpha_i\ \text{páratlan}} 1 \text{ páratlan}. \end{cases}$$

Ezt összevetve (3.2)-vel megkapjuk (3.1)-et, s ebből pedig következik a tétel.

3.19. Tétel bizonyítása. Legyen $n = p_1p_2\dots p_r$, ahol most a p_i prímek között azonosak is lehetnek. Továbbá, legyen $m = q_1q_2\dots q_s$, ahol most a q_i prímek között azonosak is lehetnek, fontos, hogy $p_i \neq q_j$. A Jacobi szimbólum multiplikativitása miatt:

$$\left(\frac{m}{n}\right) = \prod_{1\le i\le r,\ 1\le j\le s} \left(\frac{q_j}{p_i}\right),\ \left(\frac{n}{m}\right) = \prod_{1\le i\le r,\ 1\le j\le s} \left(\frac{p_i}{q_j}\right).$$

Legyen a p_i prímek között u darab, a q_j prímek között v darab $4k+3$ alakú egész szám. Gauss kvadratikus reciprocitási tétele alapján ekkor uv darab p_i, q_j párra teljesül, hogy

$$\left(\frac{q_j}{p_i}\right) = -\left(\frac{p_i}{q_j}\right),$$

a többi párra pedig azonos a kongruencia bal és jobb oldalán álló Legendre szimbólum. Vagyis:

$$\left(\frac{m}{n}\right) = (-1)^{uv}\left(\frac{n}{m}\right).$$

Viszont itt uv pontosan akkor páratlan, ha u és v is páratlan, ami azzal ekvivalens, hogy m és n is $4k+3$ alakú. Ezzel a tétel állítását beláttuk.

Példa Megoldható-e az

$$x^2 \equiv 7411 \pmod{9283}$$

kongruencia?

Meg kell vizsgálni, hogy 9283 prím-e. Mivel az, így a Legendre és Jacobi szimbólum definíciója megegyezik. A továbbiakban tehát számolhatunk Jacobi szimbólummal. Ez:

$$\begin{aligned}\left(\frac{7411}{9283}\right) &= (-1)^{\frac{7411-1}{2}\cdot\frac{9283-1}{2}}\left(\frac{9283}{7411}\right)\\ &= -\left(\frac{1872}{7411}\right) = -\left(\frac{2^4\cdot 117}{7411}\right)\\ &= -\left(\frac{2}{7411}\right)^4\left(\frac{117}{7411}\right)\end{aligned}$$

$\left(\frac{2}{117}\right)^4 = 1$ ezért marad a negatív előjel

$$\begin{aligned}&= -(-1)^{\frac{117-1}{2}\cdot\frac{7411-1}{2}}\left(\frac{7411}{117}\right)\\ &= -\left(\frac{40}{117}\right) = -\left(\frac{2}{117}\right)^3\left(\frac{5}{117}\right)\end{aligned}$$

$\left(\frac{2}{117}\right) = -1$ így

$$= (-1)^{\frac{5-1}{2}\cdot\frac{117-1}{2}}\left(\frac{117}{5}\right) = \left(\frac{2}{5}\right) = -1.$$

A fentiek alapján a kongruencia nem oldható meg.

Hivatkozások

[1] Freud R., Gyarmati E., *Számelmélet*, Nemzedékek Tudása Tankönyvkiadó, 2006.

[2] Gyarmati E., Turán P., *Számelmélet*, Tankönyvkiadó, 1969.

[3] G. Hardy, E. M. Wright, *An Introduction to the Theory of Numbers*, Oxford University Press, 2008, sixth edition.

3.3. Pár szó a lánctörtekről

Legyen x valós szám. Ekkor x-et szeretnénk felírni

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}} \tag{3.3}$$

alakban, ahol $a_0 \in \mathbb{Z}$, $a_1, a_2, \dots \in \mathbb{Z}^+$, továbbá $a_i \geq 1$ ha $i \geq 1$ (azaz a_0 akár 0 is lehet vagy negatív egész szám, de $a_i \geq 1$ ha $i \geq 1$). Ez x lánctört alakja. A definícióból nem nyilvánvaló, hogy minden valós szám felírható lánctört alakban, de ezt a 3.22. Tétel után be fogjuk bizonyítani. A fenti emeletes tört helyett gyakran csak a jóval helytakarékosabb $x = [a_0; a_1, a_2, \dots]$ jelölést használják, de mi a jegyzetben maradunk az emeletes törteknél a könnyebb átláthatóság kedvéért.

Ekkor $a_1 \geq 1$ miatt

$$a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}} > 1,$$

azaz

$$\cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}} < 1. \tag{3.4}$$

Ekkor (3.3) és (3.4)-ből, valamint hogy a_0 egész szám, következik az alábbi:

$$a_0 = [x]$$

és

$$\cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\ddots}}}} = \{x\}.$$

Így:

$$x = a_0 + x_0,$$

ahol

$$a_0 = [x], \qquad x_0 = \{x\} = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\ddots}}}}.$$

Ekkor

$$\frac{1}{x_0} = a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{\ddots}}}.$$

A fenti algoritmus ugyanígy folytatható. Még egy lépést megmutatunk:

$$\frac{1}{x_0} = a_1 + x_1,$$

ahol

$$a_1 = \left[\frac{1}{x_0}\right], \qquad x_1 = \left\{\frac{1}{x_0}\right\} = \frac{1}{x_0} - a_1.$$

Az eddigi számolásaink azt mutatják, hogy az a_i lánctört számjegyek egy algoritmussal is megadhatóak. Ez a következő:

$$a_0 = [x], \qquad x_0 = \{x\},$$

ha pedig $i \geq 1$, akkor

$$a_i = \left[\frac{1}{x_{i-1}}\right], \qquad x_i = \left\{\frac{1}{x_{i-1}}\right\} = \frac{1}{x_{i-1}} - a_i.$$

Ekkor

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \ldots \cfrac{1}{a_i + x_i}}}.$$

Továbbá az így definiált a_i-kre valóban fennáll a (3.3) összefüggés. Itt kicsi pontosítás azért szükséges még. Ha $\frac{1}{x_{i-1}}$ egész szám, akkor az algoritmus vagy megáll, vagyis $a_i = \frac{1}{x_{i-1}}$, vagy a következő lépésben áll meg, és $a_i = \frac{1}{x_{i-1}} - 1$, $a_{i+1} = 1$. Ez azt mutatja, hogy a racionális számoknak kétféle lánctört alakja is van. (Erre később is visszatérünk.)

A következőkben két egyszerű állítást ismertetünk bizonyítás nélkül. Ezek közül az első, hogy a fenti algoritmus pontosan akkor ér véget véges sok lépésben, ha x racionális. A második állításunk Lagrange tétele, mely a következőt mondja ki:

3.20. Tétel. (Lagrange tétel) *Ha x gyöke egy egész együtthatós másodfokú egyenletnek, akkor x lánctört alakjában a jegyek egy idő után periodikusak lesznek, és fordítva is, ha x lánctört alakjában a jegyek periodikusak, akkor x gyöke egy másodfokú egyenletnek.*

A lánctörtek tanulmányozása során fontos fogalom a következő:

3.21. Definíció. *Legyen x lánctört alakja:*

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}}.$$

Az i-edik közelítő tört definíciója:

$$\frac{p_i}{q_i} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots \cfrac{1}{a_{i-1} + \cfrac{1}{a_i}}}}.$$

A közelítő törtekre vonatkozó első tételünk az alábbi:

3.22. Tétel. *Legyen az x lánctört alakja a következő:*

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}}$$

Definiáljuk most a p_i és q_i egész számokat az alábbiak szerint:

$$\begin{aligned}
&p_0 = a_0, \quad q_0 = 1,\\
&p_1 = a_0a_1 + 1, \quad q_1 = a_1,\\
&p_i = a_ip_{i-1} + p_{i-2}, \quad q_i = a_iq_{i-1} + q_{i-2}, \text{ ha } i \geq 2.
\end{aligned}$$

Ekkor fennállnak a következők:

a)

$$\begin{aligned}
&\frac{p_0}{q_0} = \frac{a_0}{1},\\
&\frac{p_1}{q_1} = a_0 + \frac{1}{a_1},\\
&\frac{p_i}{q_i} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots \cfrac{1}{a_{i-1} + \cfrac{1}{a_i}}}}.
\end{aligned}$$

b) $p_i q_{i-1} - p_{i-1} q_i = (-1)^{i-1}$, *ha* $i \geq 1$.

c) $(p_i, q_i) = 1$.

A 3.22. Tétel elején kicsit másképp definiáltuk a p_i, q_i egész számokat, mint a 3.21. Definícióban. Azonban a tétel c) részéből kiderül, hogy az új definícióban $(p_i, q_i) = 1$, így a két definíció valóban egybeesik. Megjegyezzük továbbá, hogy a tétel a) és b) része tetszőleges a_i számokra is fennáll, nem szükséges feltenni, hogy az a_i egész számok, és ezt ki is fogjuk használni a fejezet későbbi részeiben.

A 3.22. Tétel bizonyítása. A tétel a) részét i-re vonatkozó teljes indukcióval igazoljuk.

Kezdőlépés: $i = 0$ és $i = 1$ esetén valóban

$$\frac{p_0}{q_0} = \frac{a_0}{1} \quad \text{és} \quad \frac{p_1}{q_1} = \frac{a_0 a_1 + 1}{a_1} = a_0 + \frac{1}{a_1},$$

így az állítás nyilvánvaló.

Ezután rátérünk az indukciós lépésre: Feltesszük, hogy $i = k$-ra beláttuk az állítást. Bebizonyítjuk $i = k + 1$-re is, vagyis igazoljuk, hogy

$$\frac{p_{k+1}}{q_{k+1}} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots \cfrac{1}{a_k + \cfrac{1}{a_{k+1}}}}}.$$

A fenti lánctörtben $k + 1$ darab lánctört jegy van, de egy ügyes jelöléssel felírható k darab lánctört jegy segítségével is. Legyen

$$a'_k = a_k + \frac{1}{a_{k+1}}.$$

Nyilván az a'_k nem egész szám, de mint említettük ez a kikötés a tétel a)

részében nem is szükséges. Ekkor

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots \cfrac{1}{a_k + \cfrac{1}{a_{k+1}}}}} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots \cfrac{1}{a_{k-1} + \cfrac{1}{a'_k}}}}.$$

Azonban az egyenlet jobb oldalán már csak k darab lánctört jegy van. Mivel feltettük, hogy az állítás igaz k darab lánctört jegyre, így

$$\frac{p_0}{q_0} = \frac{a_1}{1},\ \frac{p_1}{q_1} = a_0 + \frac{1}{a_1}, \ldots,\ \frac{p_{k-1}}{q_{k-1}} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots \cfrac{1}{a_{k-2} + \cfrac{1}{a_{k-1}}}}}.$$

A k-adik lánctört jegy ezután a'_k (nem pedig a_k), így a k-adik közelítő tört legyen $\frac{p'_k}{q'_k}$, ahol az indukciós feltevés miatt tudjuk, hogy

$$p'_k = a'_k p_{k-1} + p_{k-2},\ q'_k = a'_k q_{k-1} + q_{k-2}.$$

Ekkor

$$\begin{aligned}
\frac{p'_k}{q'_k} &= \frac{a_k{}' \, p_{k-1} + p_{k-2}}{a_k{}' \, q_{k-1} + q_{k-2}} \\
&= \frac{\left(a_{k-1} + \frac{1}{a_k}\right) p_{k-1} + p_{k-2}}{\left(a_{k-1} + \frac{1}{a_k}\right) q_{k-1} + q_{k-2}} \\
&= \frac{(a_{k-1}a_k + 1)p_{k-1} + a_k p_{k-2}}{(a_{k-1}a_k + 1)q_{k-1} + a_k q_{k-2}} \\
&= \frac{a_k(a_{k-1}p_{k-1} + p_{k-2}) + p_{k-1}}{a_k(a_{k-1}q_{k-1} + q_{k-2}) + q_{k-1}} \\
&= \frac{a_k p_k + p_{k-1}}{a_k q_k + q_{k-1}} = \frac{p_{k+1}}{q_{k+1}},
\end{aligned}$$

ami a bizonyítandó volt.

A tétel b) részét szintén teljes indukcióval lehet igazolni. Kezdőlépés: legyen $i = 1$. Ekkor

$$p_i q_{i-1} - p_{i-1} q_i = p_1 q_0 - p_0 q_1 = (a_0 a_1 + 1) \cdot 1 - a_0 a_1 = 1.$$

Ezután rátérhetünk az indukciós lépésre. Feltesszük, hogy az állítást beláttuk $i = k$-ra. Belátjuk $i = k + 1$-re. Ehhez:

$$\begin{aligned} p_{k+1} q_k - p_k q_{k+1} &= (a_k p_k + p_{k-1}) q_k - p_k (a q_k + q_{k-1}) \\ &= p_{k-1} q_k - p_k q_{k-1} = -(-1)^{k-1} = (-1)^k. \end{aligned}$$

Végül bebizonyítjuk a tétel c) részét is: Legyen $d \stackrel{\text{def}}{=} (p_i, q_i)$,

$$\begin{aligned} &d \mid \underbrace{p_i q_{i-1}}_{d|} - \underbrace{p_{i-1} q_i}_{d|} = (-1)^{i-1}, \\ &d \mid 1 \;\Rightarrow\; d = 1. \end{aligned}$$

Ezzel a tétel állításait beláttuk.

A tétel b) részében szereplő egyenletet $q_i q_{i-1}$-gyel osztva kapjuk:

$$\frac{p_i}{q_i} - \frac{p_{i-1}}{q_{i-1}} = \frac{(-1)^{i-1}}{q_i q_{i-1}}. \tag{3.5}$$

Definíció szerint

$$\frac{p_{i+1}}{q_{i+1}} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots \cfrac{1}{a_i + \cfrac{1}{a_{i+1}}}}}.$$

Ha a_{i+1} helyébe $\frac{1}{x_i}$-t írnánk, ennek a törtnek az értéke x lenne, tehát:

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots + \cfrac{1}{a_i + \cfrac{1}{\cfrac{1}{x_i}}}}} \tag{3.6}$$

A 3.22. Tétel a) részének alkalmazása során nincs szükség arra, hogy a lánctört jegyek egészek, így alkalmazva az a) részt (3.6)-re kapjuk, hogy:

$$x = \frac{\frac{1}{x_i}p_i + p_{i-1}}{\frac{1}{x_i}q_i + q_{i-1}} = \frac{p_i + x_i p_{i-1}}{q_i + x_i q_{i-1}}$$

Egyszerű számolás mutatja, hogy a fenti tört (azaz az x valós szám is) $\frac{p_i}{q_i}$ és $\frac{p_{i-1}}{q_{i-1}}$ közé esik. A közelítő törtek x-hez konvergálnak, mivel $\left|\frac{p_i}{q_i} - \frac{p_{i-1}}{q_{i-1}}\right| \longrightarrow 0$ és az x szám $\frac{p_i}{q_i}$ és $\frac{p_{i-1}}{q_{i-1}}$ közé esik.

A következőkben két olyan tételt igazolunk, amelyekre a jegyzet későbbi fejezeteiben nagy szükség lesz. A 3.23. Tétel a faktorizációs algoritmusoknál lesz szükség, nevezetesen a lánctört algoritmusnál, míg a 3.24. Tétel egy komoly RSA-ra vonatkozó támadás alapja.

3.23. Tétel. *Legyen az $x \geq \frac{1}{2}$ valós szám i-edik közelítő törtje $\frac{p_i}{q_i}$. Ekkor:*

$$\left|{p_i}^2 - x^2 {q_i}^2\right| < 2x.$$

A 3.23. Tétel bizonyítása. Tudjuk, hogy

$$\left|p_i^2 - x^2 q_i^2\right| = q_i^2 \left|x - \frac{p_i}{q_i}\right| \left|x + \frac{p_i}{q_i}\right|.$$

Mivel x a $\frac{p_i}{q_i}$ és $\frac{p_{i+1}}{q_{i+1}}$ közelítő törtek közé esik, továbbá (3.5) fennáll így:

$$\left|x - \frac{p_i}{q_i}\right| \leq \left|\frac{p_{i+1}}{q_{i+1}} - \frac{p_i}{q_i}\right| = \frac{1}{q_i q_{i+1}}.$$

Továbbá

$$\left|x+\frac{p_i}{q_i}\right| = \left|\frac{p_i}{q_i} - x + 2x\right| \leq \left|\frac{p_i}{q_i} - x\right| + 2x \leq 2x + \frac{1}{q_i q_{i+1}}.$$

Vagyis:

$$\begin{aligned}\left|p_i^2 - x^2 q_i^2\right| &= q_i^2 \left|x - \frac{p_i}{q_i}\right| \cdot \left|x + \frac{p_i}{q_i}\right| \\ &\leq q_i^2 \frac{1}{q_i q_{i+1}} \left(2x + \frac{1}{q_i q_{i+1}}\right) \\ &= 2x\frac{q_i}{q_{i+1}} + \frac{1}{q_{i+1}^2}.\end{aligned}$$

Átrendezve:

$$\begin{aligned}\left|{p_i}^2 - x{q_i}^2\right| - 2x &< 2x\left(-1 + \frac{q_i}{q_{i+1}} + \frac{1}{2x{q_{i+1}}^2}\right) \\ &< 2x\left(-1 + \frac{q_i}{q_{i+1}} + \frac{1}{q_{i+1}}\right) \\ &< 2x\left(-1 + \frac{q_{i+1}}{q_{i+1}}\right) = 0.\end{aligned}$$

Így

$$\left|{p_i}^2 - x{q_i}^2\right| < 2x.$$

Ezzel a tétel állítását beláttuk.

3.24. Tétel. (Lagrange) *Ha*

$$\left|\frac{p}{q} - \alpha\right| < \frac{1}{2q^2}, \tag{3.7}$$

ahol p és q relatív prímek, akkor $\frac{p}{q}$ az α lánctörtjének egy közelítő törtje.

A 3.24. Tétel bizonyítása. A bizonyítás a következő lemmán alapul:

3.25. Lemma. *Ha*

$$x = \frac{P\zeta + R}{Q\zeta + S},$$

ahol $\zeta > 1$ és P, Q, R, S egész számokra

$$Q > S > 0, \quad PS - QR = \pm 1,$$

akkor $\frac{R}{S}$ és $\frac{P}{Q}$ az x két egymást követő közelítő törtje. Amennyiben $\frac{R}{S}$ az $n-1$-edik közelítő tört, $\frac{P}{Q}$ pedig az n-edik, akkor ζ az úgynevezett $n+1$-edik kiegészítő lánctört, vagyis

$$\zeta = a_{n+1} + \cfrac{1}{a_{n+2} + \cfrac{1}{\ddots}},$$

ahol az a_i természetes számok az x szám lánctört jegyei.

3.25. Lemma bizonyítása: Írjuk fel $\frac{P}{Q}$-t lánctört alakban:

$$\frac{P}{Q} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots \cfrac{1}{a_{n-1} + \cfrac{1}{a_n}}}} = \frac{p_n}{q_n}. \tag{3.8}$$

Itt tetszés szerint feltehetjük, hogy n páros vagy páratlan, ugyanis minden véges lánctörtnek két alakja van, az egyikben a lánctört jegyek száma páros, a másikban páratlan. Ez az állítás a következő észrevételre alapozódik: ha $a_k \geq 2$, akkor:

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots + \cfrac{1}{a_{k-1} + \cfrac{1}{a_k}}}} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots + \cfrac{1}{a_{k-1} + \cfrac{1}{a_k - 1 + \cfrac{1}{1}}}}}.$$

Így (3.8)-ben választhatjuk úgy n paritását, hogy

$$PS - QR = (-1)^{n-1}$$

teljesüljön. Ekkor $(P, Q) = 1$, $Q > 0$ és $(p_n, q_n) = 1$. Így (3.8) alapján $P = p_n$, $Q = q_n$. Vagyis

$$p_n S - q_n R = PS - QR = (-1)^{n-1} = p_n q_{n-1} - p_{n-1} q_n.$$

Átrendezve

$$p_n(S - q_{n-1}) = q_n(R - p_{n-1}).$$

Mivel $(p_n, q_n) = 1$, ezért

$$q_n \mid S - q_{n-1}. \tag{3.9}$$

Azonban

$$q_n = Q > S > 0, \ q_n > q_{n-1} > 0,$$

és így

$$|S - q_{n-1}| < q_n.$$

Ekkor (3.9) miatt ez csak úgy lehet, ha $S - q_{n-1} = 0$. Vagyis

$$S = q_{n-1}, \quad R = p_{n-1}.$$

Összefoglalva az eddigieket

$$x = \frac{p_n \zeta + p_{n-1}}{q_n \zeta + q_{n-1}}.$$

Tekintsük most azt a lánctörtet, amelynek az első n darab lánctört jegye megegyezik x lánctört jegyeivel, az $n+1$-edik lánctört jegy pedig „ζ”, ami ugyan nem egész szám, de mint mondtuk, a 3.22. Tétel a) és b) részében ez nem is kell, hogy kikötés legyen. A tétel értelmében, az $n+1$-edik közelítő törtre $\frac{p'_n}{q'_n}$-re tudjuk, hogy

$$p'_n = p_n \zeta + p_{n-1},$$

$$q'_n = q_n\zeta + q_{n-1}.$$

Így

$$\frac{p'_n}{q'_n} = \frac{p_n\zeta + p_{n-1}}{q_n\zeta + q_{n-1}} = x.$$

Írjuk fel ζ-t lánctört alakban, ahol a lánctört számjegyeit rendre $a_{n+1}, a_{n+2}, \ldots$ jelöli, vagyis

$$\zeta = a_{n+1} + \cfrac{1}{a_{n+2} + \cfrac{1}{\ddots}}.$$

A tételben szereplő feltétel miatt $a_{n+1} = [\zeta] \geq 1$, így valóban x lánctört alakja

$$x = a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}.$$

Ezzel pedig a lemmában szereplő összes állítást igazoltuk.

Térjünk vissza a 3.23. Tétel igazolásához. Legyen

$$\frac{p}{q} - \alpha = \frac{\varepsilon\theta}{q^2},$$

ahol a tételben szereplő (3.7) feltétel miatt feltehető, hogy

$$\varepsilon = \pm 1, \quad 0 < \theta < \frac{1}{2}.$$

Írjuk fel $\frac{p}{q}$-t lánctört alakban

$$\frac{p}{q} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots \cfrac{1}{a_{n-1} + \cfrac{1}{a_n}}}} = \frac{p_n}{q_n}, \tag{3.10}$$

ahol n paritását tetszés szerint választhatjuk, most legyen n olyan, hogy

$$\varepsilon = (-1)^{n-1}.$$

Definiáljuk ζ-t az

$$\alpha = \frac{\zeta p_n + p_{n-1}}{\zeta q_n + q_{n-1}}$$

összefüggéssel, ahol p_n/q_n az utolsó és p_{n-1}/q_{n-1} az utolsó előtti közelítő törtje $\frac{p}{q}$-nak (3.10) felírásában.

$$\frac{\varepsilon\theta}{q_n^2} = \frac{p_n}{q_n} - \alpha = \frac{p_n q_{n-1} - p_{n-1} q_n}{q_n\left(\zeta q_n + q_{n-1}\right)} = \frac{(-1)^{n-1}}{q_n\left(\zeta q_n + q_{n-1}\right)},$$

így

$$\theta = \frac{q_n}{\zeta q_n + q_{n-1}}.$$

Mivel $0 < \theta < \frac{1}{2}$

$$\zeta = \frac{1}{\theta} - \frac{q_{n-1}}{q_n} > 1.$$

A 3.25. Lemma alapján $\frac{p_{n-1}}{q_{n-1}}$ és $\frac{p_n}{q_n}$ egymást követő közelítő törtek α lánctört alakjában. Ezzel az állítást igazoltuk.

Hivatkozások

[1] Freud R., Gyarmati E., *Számelmélet*, Nemzedékek Tudása Tankönyvkiadó, 2006.

[2] Gyarmati E., Turán P., *Számelmélet*, Tankönyvkiadó, 1969.

[3] G. Hardy, E. M. Wright, *An Introduction to the Theory of Numbers*, Oxford University Press, 2008, sixth edition.

4. Pszeudovéletlenség

A pszeudovéletlen sorozatoknak rengeteg alkalmazásuk van. Használjuk például természeti jelenségek, gazdasági folyamatok valósághű szimulációjához vagy kulcsként különböző titkosítási eljárásokban. Pszeudovéletlen sorozatokat használunk a matematika és fizika sok ágában, különösen a kriptográfiában és a numerikus analízisben is.

Így például a 2.5. fejezetben egy különösen fontos alkalmazását láttuk pszeudovéletlen sorozatoknak, nevezetesen a Vernam ciphert [1]. A Vernam cipher az egyik legismertebb, legelterjedtebb titkosítási eljárás, mely a mai napig használatban van. Ahhoz, hogy megértsük a pszeudovéletlen generálás fontosságát, érdemes elolvasni egy Vernam cipherről szóló ismertetőt (akár ennek a jegyzetnek a 2.5. fejezetét).

Ebben a fejezetben úgy általában a véletlen és pszeudovéletlen generálást járjuk körbe, a teljesség igénye nélkül.

De mi is az a véletlenszám generálás? Az ókortól kezdve napjainkig a véletlen-generálás mindig fontos szerepet játszott. Kérdéses, hogy az általunk használt módszerek valóban véletlenszámokat állítanak-e elő. Általánosan elfogadott nézet, hogy valódi véletlenszámokat csak fizikai módszerekkel lehet generálni, de vajon ezek a fizikai módszerek tényleg megfelelőek-e?

A fejezetben azt vizsgálom, hogy egy adott hosszú 0,1 sorozat (például az

1100100110)

mikor tekinthető pszeudovéletlennek. Nyilvánvalóan az

1111111111

sorozat vagy az

1010101010

sorozat nem tekinthető pszeudovéletlennek, „túlságosan szabályosak” az előbbi csupa egyesből áll, az utóbbiban egyesek és nullák váltják egymást. A pszeudovéletlen generátorok matematikai formulák segítségével állítanak elő véletlennek kinéző számsorozatokat.

Hivatkozások

[1] G. S. Vernam, *Cipher printing telegraph systems for secret wire and radio telegraphic communications*, Transactions of the American Institute of Electrical Engineers 55 (1926), 109–115.

[2] Fotó, http://dibujosa.com/dibujosgratisapp.php?codigo=20886

4.1. Pszeudovéletlenség - Előzmények

Az elmúlt 70 évben rengeteg cikk született a pszeudovéletlenség témaköréből, melyekben különböző célok, megközelítések, matematikai módszerek széles skálája szerepel. Talán érdemes egy klasszikus könyvet kiemelni itt a sok közül, D. Knuth, A Számítógép-Programozás Művészete című könyvének 2. kötetében [3], a Szeminumerikus Algoritmusok-ban egész fejezetet szentel a pszeudovéletlen sorozatok előállításának.

Eleinte a pszeudovéletlenség fogalmát általában *bonyolultságelméleti úton* definiálták. Goldwasser [2] egy kitűnő áttekintő cikket írt erről a nézőpontról.

4.1. Definíció. *Egy véletlenbit-generátor egy olyan készülék vagy algoritmus, mely statisztikusan független és torzítatlan (angolul „unbiased") biteket állít elő.*

Hajdan tipikusan hardver bázisú generátorokat (pl. dióda) használtak, de szoftver bázisú generátorok (gépidő, memórianagyság és így tovább) is lehetségesek; valamelyik módszerrel előállítanak egy bitsorozatot, és utána ezt tesztelik bizonyos statisztikai tesztekkel, ezt hívják aposzteriori tesztelésnek. Ez igen komplikált, nehézkes és ma már nem kielégítő technika. Ezért ma már a véletlenbit-generátorokat pszeudovéletlen-bit-generátorokkal helyettesítik. Itt megjegyezendő, hogy az utóbbiak közül még ma is a programozásban legtöbbet használt módszer a lineáris kongruencia generátor ld. pl. [5], erről bővebben Knuth [3] könyvében is olvashatunk. Ebben a jegyzetben azonban inkább néhány modernebb módszert ismertetünk, de először lássunk egy fontos definíciót.

4.2. Definíció. *Egy* pszeudovéletlen-bit-generátor *egy olyan determinisztikus algoritmus, mely egy valóban véletlen k hosszú bináris sorozatot megadva, abból egy ℓ (mely k-nál sokkal nagyobb) hosszú, véletlenszerűnek látszó bináris sorozatot készít.*

A bemenetet (angolul „input"), mely k hosszú véletlen bináris sorozatot „magnak" (angolul „seed"), a készült kimenet (angolul „output") sorozatot pszeudovéletlen-bit-sorozatnak nevezzük.

Mitől „véletlenszerűnek látszó", azaz „jó" pszeudovéletlen sorozat egy bináris sorozat? Az alkalmazástól (is) függ, milyen véletlen tulajdonságot díjazunk.

Egy kriptográfiában gyakran használt követelmény a megjósolhatatlanság (angolul „inpredictability"):

4.3. Definíció. *Azt mondjuk, hogy egy pszeudovéletlen generátor kielégíti a következőbit tesztet, ha nincs olyan polinomiális idejű algoritmus, mellyel*

az első k jegy ismeretében a $k+1$-edik $1/2$-nél lényegesen nagyobb valószínűséggel megjósolható.

Bár sokszor igen fontos ez a teszt, de megjegyezzük, hogy pl. az oly sokszor használt Monte-Carlo-módszerekben nem követelik ezt meg.

Kritikája a következőbit tesztnek: lényegében csak rekurzív konstrukciók minősítésére alkalmas, tehát lehet, hogy egy generátor keresztülmegy, de pl. az első és az utolsó bit ismeretében esetleg az egész sorozat egyértelműen meghatározott. Továbbá polinomiális idejű algoritmus nem létezése csak feltételesen igazolható. Ráadásul, nehéz meghatározni, hogy a definícióban mit jelent pontosan az, hogy $1/2$-nél „lényegesen nagyobb"?

A legfontosabb, a következőbit tesztet (feltételesen) kielégítő konstrukció:

4.4. Definíció. *A Blum–Blum–Shub pszeudovéletlen generátor [1]:*

1. *Tekintsünk két „nagy" véletlen, $4k+3$ alakú p, q prímet, legyen $n = pq$.*
2. *Tekintsünk egy „véletlen" a számot („mag") $0 < a < n$, $(a, n) = 1$-gyel. Legyen x_0 a^2-nek a* mod n *maradéka.*
3. *Ha $x_0, x_1, \ldots, x_k$ már ismert, legyen x_{k+1} az ${x_k}^2$-nek a* mod n *maradéka.*
4. *Legyen z_i az x_i szám utolsó számjegye a 2-es alapú számrendszerben.*

A konstrukcióban szereplő x_i sorozat előbb-utóbb periodikus lesz, jelölje a legkisebb periódushosszát t. Nyilván (legfeljebb) a z_i sorozat első t elemét érdemes vizsgálni pszeudovéletlenség szempontjából. Ekkor:

4.5. Tétel. (Blum, Blum, Shub, [1]) *A Blum-Blum-Shub generátor által megadott $z_0, z_1, z_2, \ldots, z_{t-1}$ bitsorozat kielégíti a következőbit tesztet, feltéve, hogy nem létezik polinomiális idejű teszt a konstrukcióban szereplő n modulus faktorizálására.*

Nem bizonyítjuk. A bizonyítás megtalálható [1]-ben.

Mostanában a bonyolultságelméleti megközelítést egyre szélesebb körben kritizálják. Ugyanis:

1. Ez a megközelítés csak generátorokat minősít, de az egyes sorozatok aposzteriori tesztelését nem teszi elkerülhetővé.

2. Az egyik alapvető standard definíció, az úgynevezett „következőbit teszt", csak bizonyítatlan hipotéziseken alapuló tesztelést tesz lehetővé.

3. Általában végtelen hosszú számsorozatok esetén használatos csak, míg a gyakorlati alkalmazások során mindig csak véges hosszú sorozatot használunk.

Hivatkozások

[1] L. Blum, M. Blum, M.Shub, *A simple unpredictable pseudo-random number generator*, SIAM Journal on Computing. Society for Industrial & Applied Mathematics (SIAM), 15 (2) (1986), 364-383.

[2] S. Goldwasser, *Mathematical foundations of modern cryptography: computational complexity perspective*, ICM, 2002, vol. I, 245-272.

[3] D. E. Knuth, *A Számítógép-Programozás Művészete (2. kötet) - Szeminumerikus Algoritmusok*, Műszaki Könyvkiadó, Budapest, 1994.

[4] A. Sárközy, *Számítógépes Számelmélet*, egyetemi előadás.

[5] Wikipédia, *Linear congruential generator*, https://en.wikipedia.org/wiki/Linear_congruential_generator

4.2. Pszeudovéletlenség - Kvantitatív megközelítés

Láttuk: a pszeudovéletlenség fenti definíciója („következőbit teszt") a gyakorlatban nem igazán kielégítő. Ezért 1997-től kezdve Mauduit és Sárközy kidolgozták a pszeudovéletlenség egy másik, a gyakorlatban sokkal jobban kezelhető definícióját. Bitsorozatokról áttértek ± 1 sorozatokra, ennek technikai oka volt, a könnyebb számolhatóság (ugyanis ekkor a kapcsolódó leggyakrabban használt valószínűségi változók várható értéke 0).

Mauduit és Sárközy [7] a következő kvantitatív mértékeket vezették be:

4.6. Definíció. *Tekintsünk egy ± 1-ekből álló N hosszú $E_N = (e_1, e_2, \ldots, e_N) \in \{-1, +1\}^N$ sorozatot.*
Ekkor az eloszlási mértéket

$$W(E_N) = \max_{a,b,t} \left| \sum_{j=0}^{t-1} e_{a+jb} \right|$$

képlettel definiáljuk, ahol a maximum az összes olyan a, b, t-n fut, ahol $a, b, t \in \mathbb{N}$ és $1 \le a \le a + (t-1)b \le N$. A k-adrendű korrelációt

$$C_k(E_N) = \max_{M,D} \left| \sum_{n=1}^{M} e_{n+d_1} e_{n+d_2} \ldots e_{n+d_k} \right|,$$

képlettel definiáljuk, ahol $M, D = (d_1, d_2, \ldots, d_k)$ az olyan párokon fut, ahol $1 \le d_1 < d_2 < \cdots < d_k \le M + d_k \le N$.

Cassaigne, Mauduit és Sárközy [1] bebizonyította, hogy majdnem minden sorozatra (azaz az összes 2^N darab sorozat közül $(1-\varepsilon)2^N$ darabra) a fenti mértékek értéke $cN^{1/2}(\log N)^{1/2}$-nél kisebb. Így azt mondjuk, egy $E_N = (e_1, e_2, \ldots, e_N)$ sorozat erős pszeudovéletlen tulajdonságokkal bír, ha létezik olyan c_1 és c_k pozitív konstans, hogy

$$W(E_N) \le N^{1-c_1},$$

$$C_k(E_N) \le N^{1-c_k}.$$

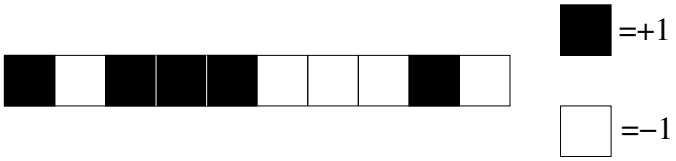
Kutatások során több szerző több olyan sorozatot generált, amelynek nagyon erős pszeudovéletlen tulajdonságai vannak, azaz:

$$W(E_N) \ll \sqrt{N}\log N,$$
$$C_k(E_N) \ll \sqrt{N}(\log N)^k.$$

A területen írt első cikkükben, 1997-ben Mauduit és Sárközy [7] a következő konstrukciót adta meg:

$$E_{p-1} = \left(\left(\frac{1}{p}\right), \left(\frac{2}{p}\right), \ldots, \left(\frac{p-1}{p}\right)\right).$$

Például, ha $p = 11$ ez a sorozat a következőképp illusztrálható:



(ld. 3.2. fejezet).

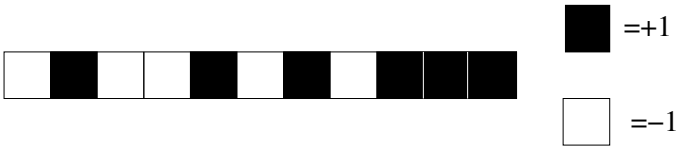
Mauduit és Sárközy [7] bebizonyították:

$$W(E_{p-1}) \ll p^{1/2}\log p \text{ és } C_\ell(E_{p-1}) \ll p^{1/2}\log p.$$

Ez a konstrukció azonban minden p prímre csak egyetlen sorozatot ad meg. Egy ügyes ötlettel Hoffstein és Lieman [5] ezt a konstrukciót úgy terjesztette ki, hogy minden egyes p prímre több sorozatot tudunk generálni egyszerre, megadva ezzel pszeudovéletlen sorozatoknak egy nagy családját:

$$E_p(f) = \left(\left(\frac{f(1)}{p}\right), \left(\frac{f(2)}{p}\right), \ldots, \left(\frac{f(p)}{p}\right)\right), \quad \text{ahol most } \left(\frac{0}{p}\right) \stackrel{\text{def}}{=} 1$$

és f egy legfeljebb $p-2$-edfokú polinom $\mathbb{Z}_p$ felett. Például, ha $p = 11$, $f(x) = x^2 + 1$ ez a sorozat a következőképp illusztrálható:



Hoffstein és Lieman konstrukciójukat numerikus számításokra alapozva adták meg, és azt sejtették, hogy erős pszeudovéletlen tulajdonságokkal rendelkezik, valamint eleget tesz a következőbit tesztnek. Így ők még nem bizonyítottak semmit a sorozat pszeudovéletlen tulajdonságairól.

Goubin, Mauduit és Sárközy [2] bebizonyították, hogy (néhány nem túl megszorító kikötést feltételezve az f polinomra) ez a sorozat erős pszeudovéletlen tulajdonságokkal rendelkezik:

$$W(E_p(f)),\ C_\ell(E_p(f)) \ll p^{1/2}\log p$$

Pontosabban:

4.7. Tétel. (Goubin, Mauduit, Sárközy, [2]) *Legyen p prímszám, $f(x) \in \mathbb{F}_p[x]$ egy k-ad fokú polinom, amely nem írható fel $cg(x)^2$ alakban, ahol $c \in \mathbb{F}_p$, $g(x) \in \mathbb{F}_p[x]$ Definiáljuk az $E_p(f) = (e_1, \dots, e_p)$ sorozatot a következő képlettel:*

$$e_n = \begin{cases} \left(\frac{f(n)}{p}\right) & \text{ha } (f(n), p) = 1, \\ +1 & \text{ha } p \mid f(n). \end{cases} \tag{4.1}$$

Ekkor

$$W(E_p(f)) \ll kp^{1/2}\log p.$$

Továbbá tegyük fel, hogy a korreláció rendjére, $\ell \in \mathbb{N}$-re fennáll a következő feltételek valamelyike:
(i) $\ell = 2$;
(ii) $\ell < p$ és 2 primitív gyök modulo p;
(iii) $(4k)^\ell < p$.
Ekkor:

$$C_\ell(E_p(f)) \ll k\ell p^{1/2}\log p.$$

Amennyiben $f(x) = cg(x)^2$ alakú, a sorozatunk, kivéve f zérushelyeit, ugyanazt az elemet tartalmazza csak, $\left(\frac{c}{p}\right)$-t, így a sorozat kulcsként teljesen alkalmatlan a kriptográfiában. Ezért a tételben szereplő első kikötés nagyon fontos. Szerencsére, ilyen polinomból nincs sok: az összes k-adfokú polinomból (számuk p^{k+1} darab), mindössze $2p^{[k/2]+1}$ darab $cg(x)^2$ alakú van...

Azóta számos további konstrukció született, de bizonyos értelemben még mindig ez a legjobb: erős pszeudovéletlen tulajdonságokkal rendelkezik, és a sorozat elemeinek generálása gyors. A teljesség kedvéért megmutatunk néhány ilyen konstrukciót.

Ezek a konstrukciók gyorsan számolhatóak, bizonyítottan „jó" pszeudovéletlen tulajdonságokkal rendelkeznek az általunk definiált értelemben, ezért aposzteriori tesztelésük nem szükséges, és bizonyítottan (nem csupán feltételesen) jó kriptográfiai tulajdonságokkal rendelkeznek.

4.8. Tétel. (Mauduit, Rivat, Sárközy [6]) *Legyen p prímszám, $f(x) \in \mathbb{Z}[x]$ egy k-ad fokú polinom és jelölje $r_p(n)$ az n egész szám legkisebb nem negatív maradéka* mod p. *Definiáljuk az $E_p(f) = (e_1, \dots, e_p)$ sorozatot a következő képlettel:*

$$e_n = \begin{cases} +1, & \text{ha } 0 \le r_p(f(n)) < p/2, \\ -1, & \text{ha } p/2 \le r_p(f(n)) < p. \end{cases}$$

Ekkor $2 \le \ell \le k-1$ esetén

$$W(E_p(f)) \ll kp^{1/2}(\log p)^2$$
$$C_\ell(E_p(f)) \ll kp^{1/2}(\log p)^{\ell+1}.$$

A konstrukció hiányossága: magas rendű korreláció nagy lehet. Még egy konstrukció:

4.9. Tétel. (Mauduit, Sárközy [8]) *Legyen p prímszám, $f(x) \in \mathbb{Z}[x]$ egy k-ad fokú polinom és jelölje $r_p(n)$ az n egész szám legkisebb nem negatív*

maradéka mod p. *Definiáljuk az* $E_p(f) = (e_1, \ldots, e_p)$ *sorozatot a következő képlettel:*

$$e_n = \begin{cases} +1 & \text{ha } (f(n), p) = 1 \text{ és } 0 < r_p(f(n)^{-1}) < p/2, \\ -1 & \text{különben.} \end{cases}$$

Továbbá tegyük fel, hogy a korreláció rendjére, $\ell \in \mathbb{N}$*-re fennáll a következő feltételek valamelyike:*

(i) $\ell = 2$*;*

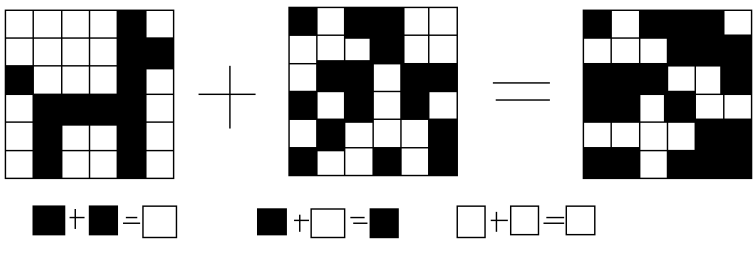
(ii) $(4k)^\ell < p$*;*

(iii) $k\ell < p/2$ *és* $f(x)$ *felírható a következő alakban:* $f(x) = (x + a_1)(x + a_2)...(x + a_k)$ *ahol* $a_i \in \mathbb{F}_p$*. Ekkor:*

$$W(E_p(f)) \ll kp^{1/2}(\log p)^2$$
$$C_\ell(E_p(f)) \ll k\ell p^{1/2}(\log p)^{\ell+1}.$$

Rendkívül fontos, hogy néhány fenti módon definiált sorozatot aposzteriori teszteléssel is ellenőrzött Rivat és Sárközy [9] cikkükben, ahol az említett aposzteriori tesztek az amerikai szabványyhivatal, "National Institute of Standards and Technology" által megjelölt ″1.4-sts. package″ csomagban találhatóak. Sőt, Rivat és Sárközy azt is bizonyította, hogy ha a pszeudovéletlen mértékek kicsik, akkor a sorozat „majdnem" eleget tesz a fenti tesztek közül jónéhánynak, és így az aposzteriori tesztelés egy része elkerülhetővé válik.

Előfordulhat azonban, hogy nem szöveget, hanem képet szeretnénk titkosítani, ekkor nem pszeudovéletlen sorozatokra, hanem ***pszeudovéletlen rácsokra*** van szükség.

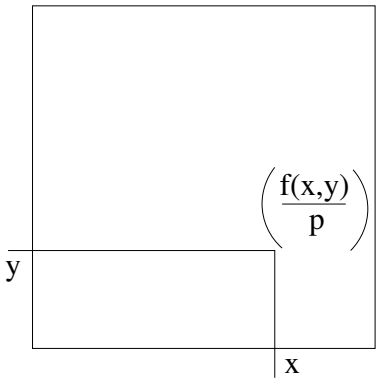


Társszerzőimmel, Sárközy Andrással és Cameron L. Stewarttal [3], [4] közösen a Legendre szimbólumon alapuló, erős pszeudovéletlen tulajdonsággal rendelkező rácsot generáltunk.

4.10. Konstrukció. (Gyarmati, Sárközy, Stewart, [3], [4])
Legyen p prím és $f(x,y) \in \mathbb{Z}[x,y]$ két változós polinom. Definiáljuk az $\eta: \{0,1,\dots,p-1\} \times \{0,1,\dots,p-1\} \to \{-1,+1\}$ függvényt a

$$\eta(x,y) = \begin{cases} \left(\frac{f(x,y)}{p}\right) & \text{ha } p \nmid f(x,y), \\ +1 & \text{ha } p \mid f(x,y), \end{cases}$$

képlettel.



Azt kaptuk, hogy az így definiált konstrukció több dimenziós pszeudovéletlen mértékei jól becsülhetőek. A konstrukció további nagy előnye, hogy nagyon gyorsan konstruálhatók a rács elemei.

Az eddig ismertetésre került konstrukciók a Legendre szimbólumra és egy f (egy vagy többváltozós) polinomra épültek.

Ahhoz hogy valaki le tudja programozni a sorozatot vagy rácsot, elég ismernie a p prím és az f polinom együtthatóinak értékét.

Tegyük fel, hogy Alíz és Bob úgy szeretne megállapodni a p prím és az f polinom együtthatóinak értékeiben, hogy ők tudják azokat, de rajtuk kívül senki más ne tudja meg. Erre a problémára alkalmazható a **Diffie-Hellman kulcscserélő eljárás**, amelyet részletesen a 12. fejezetben tárgyalunk.

Hivatkozások

[1] J. Cassaigne, C. Mauduit, A. Sárközy, *On finite pseudorandom binary sequences VII: The measures of pseudorandomness*, Acta Arith. 103 (2) (2002), 97-118.

[2] L. Goubin, C. Mauduit, A. Sárközy, *Construction of large families of pseudorandom binary sequences*, Journal of Number Theory 106 (1) (2004), 56-69

[3] K. Gyarmati, A. Sárközy and C. L. Stewart, *On Legendre symbol lattices*, Unif. Distrib. Theory 4 (2009), 81-95.

[4] K. Gyarmati, A. Sárközy and C. L. Stewart, *On Legendre symbol lattices, II*, Unif. Distrib. Theory 8 (2013), 47-65.

[5] J. Hoffstein, D. Lieman, *The distribution of the quadratic symbol in function fields and a faster mathematical stream cipher*, Progress in Computer Science and Applied Logic, Vol. 20, Birkhäuser, Verlag, Basel, 2001, 59-68.

[6] C. Mauduit, J. Rivat and A. Sárközy, *Construction of pseudorandom binary sequence using additive characters*, Monatshefte Math. 141 (2004), 197-208.

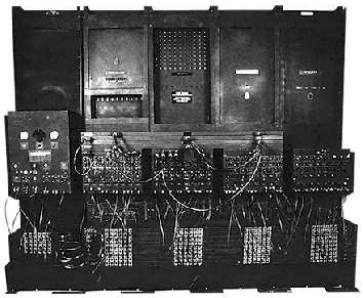
[7] C. Mauduit, A. Sárközy, *On finite pseudorandom binary sequences I: Measure of pseudorandomness, the Legendre symbol*, Acta Arith. 82 (4) (1997), 365-377

[8] C. Mauduit, A. Sárközy, *Construction of pseudorandom binary sequences by using the multiplicative inverse*, Acta Math. Hungar. 108 (2005), 239-252.

[9] J. Rivat and A. Sárközy, *On pseudorandom sequences and their application*, Lecture Notes in Comput. Sci. 4123, General theory of information transfer and combinatorics, Springer, Berlin / Heidelberg, 2006, 343-361.

5. Neumann-elvek

1946-ban megépítették az első teljesen elektronikusan működő számítógépet, az ENIAC-ot.



Az építés során szerzett tapasztalatok alapján Neumann János kidolgozta a számítógép építéséhez akkoriban nélkülözhetetlen alapelveket. Habár manapság a gyakorlat során ezek az alapelvek kissé (de csak kissé) módosultak, a mai napig ezek szemléltetik legjobban a számítógépek működését. A Neumanni alapelvek a következők:

1. Teljesen elektronikus működés.
2. Kettes számrendszer használata.
3. Belső memória használata.
4. Tárolt program elve. A számításokhoz szükséges adatokat és programutasításokat a gép azonos módon, egyaránt a belső memóriában (operatív tár) tárolja.
5. Soros utasítás-végrehajtás (az utasítások végrehajtása időben egymás után történjen)
6. Univerzális felhasználhatóság, Turing-gép (programozhatóság).
7. Szerkezet: öt funkcionális egység (aritmetikai egység, központi vezérlőegység, memóriák, bemeneti és kimeneti egységek).

Itt fontos megjegyezni, hogy nem az ENIAC volt az első számítógép a világon, például 1840-ben Thomas Fowler gyártott egy 3-as számrendszeren alapuló fából készült számítógépet, melynek mérete $1.8m \times 0.9m \times 0.3m$ volt. Az első elektronikus számítógépet Konrad Zuse építette 1943 körül és Z3-nak nevezte. A tenáris számítógépeket D. Knuth hozta újra divatba.

Hivatkozások

[1] H. H. Goldstine, *A számítógép Pascaltól Neumannig*, Budapest, Műszaki (2003).

[2] Kovács Gy., Szelezsán J., *Gondolatok Neumann János First Draft of a Report on the EDVAC című, 1945 júniusában megjelent tanulmányáról*, elérhető: Ki volt igazából Neumann János cikkgyűjtemény, Nemzeti Tankönyvkiadó, 2003.

[3] Neumann J., *A számológép és az agy*, Gondolat Könyvkiadó, 1964.

[4] Neumann J., *First Draft of a Report on the EDVAC*, University of Pennsylvania 1945.

[5] Szelezsán János, *Neumann János az első, számítógépet alkalmazó »fizikus«*, Fizikai Szemle, 2003/12., o. 425.

[6] Fotó, *ENIAC*, http://www.feltalaloink.hu/tudosok/neumannjanos/html/neujantal1.htm

6. Elemi aritmetikai műveletek

A fejezet Das [1] és Koeblitz [3] könyvére épül.

Manapság a legelterjedtebb számítógépek 32 vagy 64 bitesek. De vajon mit jelent az, hogy egy számítógép 64 bites? Furcsamód itt a 64-es szám a számítógép által használt számrendszer alapjához kapcsolódik. Egy 64 bites számítógép alapja olyan B-alapú számrendszerbeli számokat használ, ahol $B = 2^{64}$. Ez bizony elég nagy szám, maga a számrendszer alapja is több mint egy 21-jegyű szám a tízes számrendszerben.

Az illusztráció kedvéért vegyünk egy nagy számot a tízes számrendszerben, melyet fel szeretnénk írni 256-os számrendszerben. Ez:

$$n = 12345678987654321.$$

Ekkor

$$n = 43 \times B^6 + 220 \times B^5 + \cdots + 84 \times B^4 + 98 \times B^3 + 145 \times B^2 + 244 \times B + 177,$$

azaz

$$n = (43, 220, 84, 98, 145, 244, 177)_B.$$

Ha otthon a jegyzetfüzetünkben számolgatunk, a legtöbben még mindig a tízes számrendszert használjuk. Az átváltás tízes számrendszerről B alapúra egyszerű, és ez fordítva is így van. A füzetünkben számolgatva a legtöbbet a következő standard aritmetikai műveleteket használjuk: ÖSSZEADÁS, KIVONÁS, SZORZÁS és MARADÉKOS OSZTÁS.

Általában ezekre a standard aritmetikai műveletekre alapozva adunk algoritmusokat bonyolultabb műveletekre. Két egyszerű példát említve, megadhatjuk így a gyökvonás algoritmusát bizonyos tizedesjegy pontosságig vagy inverz számítást modulo m. Egy-egy algoritmus használhatóságában az egyik elsődleges szempont az algoritmus futási ideje, azaz hogy mennyire gyors az adott algoritmus. De vajon miben mérjük egy algoritmus időigényét? Ebben a jegyzetben a futási időt a művelet elvégzéséhez szükséges bitoperációk

számával határozzuk meg. Így 1 egységnyi időt igényel pl. 2 bit összeadása, kivonása vagy modulo 2 összeadása. Vegyünk példaként egy egyszerű kettes számrendszerbeli összeadást:

$$\begin{array}{rr} & 1\,1\,0\,1\,0\,1\,1\,0\,0 \\ + & 1\,0\,0\,1\,1\,0\,1\,1\,0\,. \\ \hline = & 1\,0\,1\,1\,1\,0\,0\,0\,1\,0 \end{array}$$

Nézzük meg egy kicsit részletesebben ezt az összeadást! A műveletet az utolsó bitnél kezdjük, azaz 0+0=0. A következő bit sem okoz gondot, azaz 0+1=1. Viszont ezután 1+1=2, a 2 pedig kettes számrendszerben leírva 10, azaz keletkezik „1” maradék, amit át kell vinnünk a következő oszlopban. Bizony, a bitoperációknál ügyelnünk kell az átvitelekre is! Az összeadásnál 8-féle bitoperáció fordulhat elő:

	Bit	Volt-e maradék az előző oszlopban?
		Nem
Első számban:	0	
Második számban:	0	
Eredmény:	0	
Keletkezik-e új maradék?	Nem	

	Bit	Volt-e maradék az előző oszlopban?
		Igen
Első számban:	0	
Második számban:	0	
Eredmény:	1	
Keletkezik-e új maradék?	Nem	

	Bit	Volt-e maradék az előző oszlopban?
		Nem
Első számban:	0	
Második számban:	1	
Eredmény:	1	
Keletkezik-e új maradék?	Nem	

	Bit	Volt-e maradék az előző oszlopban?
		Igen
Első számban:	0	
Második számban:	1	
Eredmény:	0	
Keletkezik-e új maradék?	Igen	

	Bit	Volt-e maradék az előző oszlopban?
		Nem
Első számban:	1	
Második számban:	0	
Eredmény:	1	
Keletkezik-e új maradék?	Nem	

	Bit	Volt-e maradék az előző oszlopban?
		Igen
Első számban:	1	
Második számban:	0	
Eredmény:	0	
Keletkezik-e új maradék?	Igen	

	Bit	Volt-e maradék az előző oszlopban?
		Nem
Első számban:	1	
Második számban:	1	
Eredmény:	0	
Keletkezik-e új maradék?	Igen	

	Bit	Volt-e maradék az előző oszlopban?
		Igen
Első számban:	1	
Második számban:	1	
Eredmény:	1	
Keletkezik-e új maradék?	Igen	

A kivonásnál hasonlóan járunk el, csak ott „maradék” helyett „kölcsön” keletkezik. A bitoperációk közé soroljuk még az ÉS illetve VAGY műveleteket is, valamint a modulo 2 összeadást is azaz:

ÉS				
Első számban:	0	0	1	1
Második számban:	0	1	0	1
Eredmény:	0	0	0	1

VAGY				
Első számban:	0	0	1	1
Második számban:	0	1	0	1
Eredmény:	0	1	1	1

MOD 2 ÖSSZEADÁS				
Első számban:	0	0	1	1
Második számban:	0	1	0	1
Eredmény:	0	1	1	0

Természetesen egyéb, a fentiekből nem felépíthető műveletek is előfordulhatnak egy-egy algoritmus során, azonban ezeknek az összesített időigénye a legtöbbször elhanyagolható az eddig megadott bitoperációk összes időigényéhez képest. Így általában egy-egy algoritmus időigényét az algoritmus során

végzett bitoperációk darabszámával adjuk meg. Ha kiterjesztjük a bitoperációk definícióját kettes számrendszerről nagyobb alapú, mondjuk B-alapú számrendszerekre, akkor az időigény legfeljebb egy B-től függő konstansszorzóval változik. Továbbá egy egy művelet időigénye más és más lehet különböző algoritmusok esetén. Ezért célszerű az időigényre általában egy O()-t használó becslést megadni. (Egy-egy művelet *pontos időigényét* megadni általában nagyon nehéz feladat. Általában csak felső becslésre szorítkoznak a területen.)

Nyilvánvalóan két n darab számjegyből álló szám összeadásának időigénye $O(n)$. Hasonlóan két n jegyű szám kivonásának időigénye szintén $O(n)$.

6.1. Definíció. *Egy művelet időigényét az elvégzéséhez szükséges bit operációk számával mérjük. Jelölés:*

$$T(\dots) = \dots.$$

Így pl. a memóriához való hozzáférést, s.í.t. nem számoljuk.

6.2. Tétel.

$$T(k \text{ jegyű } + \ell \text{ jegyű }) \leq \max\{k, \ell\}$$

vagy

$$T(k \text{ jegyű } + \ell \text{ jegyű }) = O(\max\{k, \ell\}).$$

Nem azt írjuk, hogy „$= \max\{k, \ell\}$”, mert csak felső becslést adunk; sok esetben (pl. ahogy majd a szorzásnál látni fogjuk), létezik az elsőre megadott egyszerű becslésnél lényegesen jobb.

6.3. Következmény.

$$T(m+n) \leq \frac{\log \max\{m, n\}}{\log 2} + 1$$

A kivonás esetében hasonló a helyzet, mint az összeadásnál, csak esetleg „1 maradék” helyett „1 kölcsön” keletkezik. Ezért a kivonást ugyanúgy kell kezelni mint az összeadást, ezzel ennél bővebben itt nem foglalkozunk.

A következőkben rátérünk a szorzásra. Az általános iskolában tanult szorzást egy példával illusztráljuk:

$$
\begin{array}{ll}
\underline{11101} \times 1101 & \\
11101 & \rightarrow \text{másoljuk} \\
\ 11101 & \rightarrow \text{másoljuk} \\
\underline{\ \ 11101} & \rightarrow \text{2-vel csúsztatjuk} \\
101111001 &
\end{array}
$$

Elemezzük az ábrát. Az első szorzandó k számjegyű, a második ℓ számjegyű. Legyen a második szorzandóban ℓ' darab egyes. Ekkor nyilván $\ell' \leq \ell$.

A csúsztatás, másolás időigénye elhanyagolható. Vagyis van ℓ' darab sorunk, mindegyikben egy k jegyű szám, ezeket kell összeadnunk. Ezt úgy tesszük, hogy az első sorhoz adjuk a másodikat, a kapott részletösszeghez a harmadikat, majd a kapott részletösszeghez a negyediket, és így tovább. Kiszámolva az időigényt, $k\ell'$-t kapunk. Mivel $\ell' \leq \ell$, ennek az algoritmusnak az időigényére az alábbiakat kapjuk:

6.4. Tétel.

$$T(k \text{ jegyű } \times \ell \text{ jegyű }) \leq k\ell$$

6.5. Következmény.

$$T(k \text{ jegyű } \times k \text{ jegyű }) \leq k^2.$$

6.6. Következmény.

$$T(m \times n) \leq \left(\frac{\log m}{\log 2} + 1\right)\left(\frac{\log n}{\log 2} + 1\right).$$

Az alább megismertetett algoritmus a legegyszerűbb szintaktikájú, de messze nem a leggyorsabb. Nagyobb számok esetén, a 8. fejezetben látunk majd lényegesen komplikáltabb módszereket, de időigényét tekintve sokkal gyorsabb szorzásra.

Az alapműveletek bonyolultságát D. Knuth is részletesen tárgyalja [2]-ben.

Hivatkozások

[1] A. Das, *Computational Number Theory*, CRC Press, 2013.

[2] D. E. Knuth, *A Számítógép-Programozás Művészete (2. kötet) - Szeminumerikus Algoritmusok*, Műszaki Könyvkiadó, Budapest, 1994.

[3] N. Koeblitz, *A Course in Number Theory and Cryptography*, Springer, 1994.

6.1. Egyszerűbb elemi műveletek időigénye

Az előző fejezetekben láttuk mennyire fontosak a számrendszerek a számítógépeknél. A számrendszerekre vonatkozó egyik első alap tétel a következő:

6.7. Tétel. *$b \in \mathbb{N}$, $b > 1$ esetén minden $n \in \mathbb{N}$ egyértelműen felírható*

$$n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0$$

alakban, ahol $a_i \in \{0, 1, \ldots, b-1\}$ minden i-re és $a_k > 0$. Ezt az előállítást b alapú számrendszerben való előállításnak, $a_k, a_{k-1}, \ldots, a_1, a_0$-t az n szám (b alapú számrendszerbeli) számjegyeinek nevezzük.

Mi leginkább a kettes számrendszert fogjuk használni.

6.8. Következmény. *Minden $n \in \mathbb{N}$ egyértelműen felírható*

$$n = \varepsilon_k 2^k + \varepsilon_{k-1} 2^{k-1} + \cdots + \varepsilon_1 2 + \varepsilon_0$$

alakban, ahol $\varepsilon_i \in \{0, 1\}$ minden i-re és $\varepsilon_k = 1$. Ezt az előállítást n diadikus vagy bináris előállításának nevezzük. Itt az $\varepsilon_k, \varepsilon_{k-1}, \ldots, \varepsilon_1, \varepsilon_0$ számjegyeket biteknek nevezzük.

A következőkben bebizonyítjuk a 6.7. Tételt.

A 6.7. Tétel bizonyítása. Nézzük először azt az esetet, amikor $1 \leq n < b$. Ekkor nyilván n számjegyeinek száma 1, azaz $k = 0$ és $a_0 = n$. A továbbiakban tegyük fel, hogy $b \leq n$. Teljes indukcióval bizonyítjuk a tételt. Az indukció kezdőlépése az $n = b$ eset, amikor is $k = 1$ és $a_1 = 1,\ a_0 = 0$. Nézzük az indukciós lépést. Tegyük fel, hogy $n = 1, 2, \ldots, m-1$-re már beláttuk az állítást, és bizonyítsuk be $n = m$-re. Olyan $a_k, a_{k-1}, \ldots, a_1, a_0$ számjegyeket keresünk, amelyre

$$m = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0.$$

Ekkor

$$m = \left(a_k b^{k-1} + a_{k-1} b^{k-2} + \cdots + a_1\right) b + a_0,$$

ahol $0 \leq a_k < b$. Ez m-nek $m = qb + r$ alakú előállítása, ami a maradékos osztás tétele szerint létezik, és egyértelmű. Vagyis a_0 egyértelműen meghatározott: $a_0 = r$. $m \geq b$ miatt $q \geq 1$, $q < qb \leq m$, vagyis q-ra alkalmazható az indukciós feltevés:

$$q = a_k b^{k-1} + a_{k-1} b^{k-2} + \cdots + a_2 b + a_1$$

esetén az $a_k, a_{k-1}, \ldots, a_2, a_1$ számjegyek egyértelműen léteznek. Ebből adódik, hogy $m = qb + r$ felírásában is a számjegyek léteznek és egyértelműek.

Az $n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0$ számban a számjegyek száma $k+1 = \left[\frac{\log n}{\log b}\right] + 1$. Így az $n = (a_k a_{k-1} \ldots a_1 a_0)_b$ alak megkereséséhez $O(\log n)$ darab maradékos osztás kell. Ebből már lehet a b alapú számrendszerre való áttérés időigényességére következtetni.

Láttuk, hogy a b alapú számrendszerre való áttérés történhet maradékos osztás segítségével. Egyelőre azonban még nem határoztuk meg a maradékos osztás időigényét. Nézzük tehát a maradékos osztást. Legyen $m \geq n$. Ekkor

$$m = qn + r,$$

ahol $q, r \in \mathbb{Z}$ és $0 \leq r < n$. Mi lesz a maradékos osztásbeli q és r? Ezt egy példával illusztráljuk:

98 : 5

$$
\begin{array}{ll}
1100010 : 101 = \underbrace{10011}_{} & \\
\underline{101} & 19 \\
01001 & \\
\underline{\ \ 101} & \\
1000 & \\
\underline{\ \ 101} & \\
\underbrace{011}_{} & \\
3 &
\end{array}
$$

Minden lépésben legalább 1-gyel csökken az osztandó jegyek száma, amíg el nem fogy:

1. lépés: k jegy,

2. lépés: $k-1$ jegy,

$\vdots$

$k-\ell+1$. lépés: ℓ jegy.

Minden lépésben kivonás ℓ vagy $\ell+1$ bitoperációval, de az utóbbi esetben a legbaloldalibb számjegy nem informatív értékű, hiszen oda biztosan 0 kerül. Ez $(k-\ell+1) \times \ell \leq k\ell$ bitoperáció.

Az alapműveletekkel végeztünk. Néhány további, ezekre visszavezethető tétel bizonyítás nélkül:

6.9. Tétel. $T\big((k \text{ jegyű })^n\big) \leq k^2 \dfrac{(n-1)n}{2}$.

6.10. Következmény.

$$
\begin{aligned}
T(a^n) &< \left(\frac{\log a}{\log 2} + 1\right)^2 \frac{(n-1)n}{2} \\
&= O(n^2 (\log a)^2).
\end{aligned}
$$

6.11. Tétel. $T(n!) = O(n^2(\log n)^2)$.

6.12. Tétel. $T\left(\binom{n}{m} \text{ kiszámítása}\right) = O\big(m^2(\log n)^2\big)$.

Mielőtt továbbmennénk, egy alapvető fontosságú definíció:

6.13. Definíció. *Egy (számítási) algoritmust akkor mondunk polinomiális idejűnek (P), ha $k_1, \ldots, k_r$ jegyű számokból kiindulva $O\big({k_1}^{d_1} \ldots {k_r}^{d_r}\big)$ (ahol $d_1, \ldots, d_r$ adott nem negatív egész számok) lépésben adja az eredményt.*

Néhány példa:

$$\begin{aligned}
&T(k \text{ jegyű } + \ell \text{ jegyű}) \le \max(k, \ell) && P,\\
&T(k \text{ jegyű } \times \ell \text{ jegyű}) \le k\ell && P,\\
&T\big((k \text{ jegyű})^n\big) = O(k^2 n^2) && nemP,\\
&T(n!) = O\big(n^2(\log n)^2\big) && nemP.
\end{aligned}$$

6.14. Tétel. *Legyenek*

$$f(x) = \sum_{i=0}^{u} a_i x^i,$$
$$g(x) = \sum_{j=0}^{v} b_j x^j$$

egész együtthatós polinomok, ahol

$$\max_{i,j}\{|a_i|, |b_j|\} \le m, \quad v \le u.$$

Ekkor

$$T(f(x)g(x)) = O\big(uv((\log m)^2 + \log v)\big).$$

A 6.14. Tétel bizonyítása.

$$f(x)g(x) = \sum_{k=0}^{u+v} \underbrace{\left(\sum_{i+j=k} a_i b_j\right)}_{\text{ennek időigénye}} x^k$$

$$\underbrace{O\big(v((\log m)^2+\log v)\big)}$$
$$O\big((u+v+1)v\big((\log m)^2+\log v\big)\big)$$
$$= O(uv(\log m)^2+\log v).$$

A gyökvonásra számos algoritmus ismert. Ezek közül a leggyorsabbakra:

6.15. Tétel. $T([\sqrt{a}]) = O((\log a)^3)$.

Nem bizonyítjuk.

A következőben egy számrendszerről egy másikra való áttérés időigényét mutatjuk, így pl. annak időigényét, ha n bináris alakban adott és b alapúra akarunk áttérni:

6.16. Tétel.

$T(n$ *konverziója bináris alakról* b *alapú számrendszerre*$) = O((\log n)^2)$.

A 6.16. Tétel bizonyítása. A 6.7. Tétel bizonyításában a maradékos osztások időigényét összeadva adódik a tétel.

Végül: Probléma. Mennyi az időigénye annak, hogy eldöntsük, hogy n prímszám-e? Klasszikus; újabban a kriptográfia miatt is. A számítógépes számelmélet egyik legfontosabb problémája!

Amennyiben nem csak egy adott számról, hanem egy adott határig az összes prímet meg szeretnénk határozni, akkor a leggazdaságosabb az eratoszthenészi szita. Az egyik első, általános vagy középiskolában tanult módszer arra, hogy meghatározzuk, hogy egy adott n szám prímszám-e a következő: Felírjuk a prímeket $\sqrt{n}$-ig

$$p_1 = 2,\ p_2 = 3,\ \ldots,\ p_k \le \sqrt{n} < p_{k+1},$$

n-et sorban elosztjuk $p_1, p_2, \ldots, p_k$-val, ha valamelyikkel osztható, összetett; megállunk. Ha egyikkel sem: prím. (Feltételezzük, adottak a $\sqrt{n}$-nél kisebb prímek.)

6.17. Tétel. *$T(n$ prím eratoszthenészi szitával$) = O(\sqrt{n}\log n)$ nemP.*

A bizonyítás házi feladat.

Ennél sokkal gyorsabb módszerek is vannak. Pl.: Miller–Rabin valószínűségi teszt vagy Agrawal–Kayal–Saxena-algoritmus. Ezekről később lesz bővebben szó a 9. fejezetben.

A prímtesztelés valójában polinomiális idejű.

Hivatkozások

[1] N. Koeblitz, *A Course in Number Theory and Cryptography*, Springer, 1994.

6.2. Oszthatóság, euklideszi algoritmus

A számelmélet alapjai közé tartoznak a következők: oszthatóság, legnagyobb közös osztó, prím és felbonthatatlan, számelmélet alaptétele, vagyis az egyetemen a matematikus szakon az első év anyaga. Jól ismert, hogy a fenti fogalmak mindegyike kapcsolódik az eukleideszi algoritmushoz.

Kezdjük az euklideszi algoritmus időigényével.

6.18. Tétel. *$a \geq b$ esetén $T((a,b)$ =? euklideszi algoritmussal$) = O((\log a)^3)$.*

A 6.18. Tétel bizonyítása.

Először írjuk fel az euklideszi algoritmus lépéseit az a és b számokra:

$$\begin{aligned} a &= bq_1 + r_1, && \text{ahol } 0 \leq r_1 < |b| \\ b &= r_1 q_2 + r_2, && \text{ahol } 0 \leq r_2 < r_1 \\ r_1 &= r_2 q_3 + r_3, && \text{ahol } 0 \leq r_3 < r_2 \end{aligned}$$

$$\vdots$$

$$r_{k-2} = r_{k-1}q_k + r_k, \qquad \text{ahol } 0 \le r_k < r_{k-1}$$
$$r_{k-1} = r_k q_{k+1}.$$

Az algoritmus során mindig az előző maradékot osztjuk az aktuális maradékkal. Az utolsó nem nulla maradék, jelen esetben r_k lesz a és b legnagyobb közös osztója. Ekkor az r_i maradékokra fennáll:

6.19. Lemma. *Tegyük fel, hogy $a > b$ természetes számok. Legyen $r_{-1} = a$, $r_0 = b$. Ekkor $i = -1, 0, 1, 2, \ldots$ esetén*

$$r_{i+2} < \frac{r_i}{2}.$$

A 6.19. Lemma bizonyítása. Két eset: Ha $r_{i+1} \le \dfrac{r_i}{2} \;\Rightarrow$

$$r_{i+2} < r_{i+1} \le \frac{r_i}{2}.$$

Ha viszont $r_{i+1} > \dfrac{r_i}{2}$, akkor nézzük az $i+2$-edik maradékos osztást:

$$r_i = r_{i+1}q_{i+2} + r_{i+2},$$
$$r_{i+2} = r_i - r_{i+1}\underbrace{q_{i+2}}_{\ge 1} < r_i - \frac{r_i}{2} = \frac{r_i}{2}.$$

A 6.19. lemmát ismételten alkalmazva kapjuk:

6.20. Lemma. $r_i \le \dfrac{a}{2^{i/2}}$.

Most már becsülhetjük $T\big((a,b) =?$ euklideszi algoritmussal$)$-t.

Definiáljuk t-t

$$2^{(t-1)/2} \le a < 2^{t/2}\text{-nel.}$$
$$2^{t-1} \le a^2 < 2^t,$$
$$t - 1 \le \frac{\log a^2}{\log 2} < t, \qquad t - 1 = \left[\frac{\log a^2}{\log 2}\right].$$

Az osztások száma az euklideszi algoritmusban $k+1$, ahol az utolsó nem nulla maradék r_k, vagyis

$$r_k \geq 1.$$

6.20. lemma szerint

$$\begin{aligned} 1 \leq r_k &\leq \frac{a}{2^{k/2}}, \\ 2^k &\leq a^2 < 2^t, \\ k &< t, \\ k &\leq t-1. \end{aligned}$$

Lépésszám: $k+1 \leq t$. Egy lépés időigénye

$$\begin{aligned} &T(i\text{-edik osztás}) = T(r_{i-2} = r_{i-1}q_i + r_i) \\ &\leq \underbrace{r_{i-2}}_{\leq a} \text{ jegyeinek száma } \times \underbrace{r_{i-1}}_{\leq a} \text{ jegyeinek száma} \\ &\leq (a \text{ jegyeinek száma})^2 \leq \left(\frac{t}{2}+1\right)^2. \end{aligned}$$

Így végül:

$$\begin{aligned} T\big((a,b) \stackrel{?}{=} \text{euklideszi algoritmussal}\big) &\leq \text{lépésszám} \times \max_i T(i\text{-edik osztás}) \\ &\leq t\left(\frac{t}{2}+1\right)^2 = O(t^3) = O((\log a)^3). \end{aligned}$$

Megjegyzés: $T\big((a,b) \stackrel{?}{=} \text{euklideszi algoritmussal}\big) = O((\log a)^2)$ is igaz, de ez jóval komplikáltabb lenne.

Teljes indukcióval könnyen belátható, hogy minden i-re az r_i maradék felírható $ax_i + by_i$ alakban, ahol x_i és y_i egész számok (az egyik pozitív, a másik negatív). Ezt az utolsó nem nulla maradékra alkalmazva kapjuk a következőt:

6.21. Tétel. *$a \geq b$ esetén $T\big((a,b)$ felírása lineáris kombinációként $\big) = T\big((a,b) = ax + by$ egyenletben egy $x, y \in \mathbb{Z}$ megoldás megkeresése$\big) = O((\log a)^3)$.*

A számolás részleteinek kidolgozását az olvasóra bízzuk.

Jelölés: A mod m maradékosztályok által alkotott egységelemes gyűrűt Z_m-mel jelöljük.

6.22. Tétel. *Az $a \in \mathbb{Z}_m$ elemnek akkor és csak akkor van multiplikatív inverze, ha $(a, m) = 1$. Ha ez teljesül, akkor a multiplikatív inverz $O((\log m)^3)$ bitoperációval megtalálható.*

A 6.22. Tétel bizonyítása. Ha $(a, m) > 1$, akkor a többszörösei mind oszthatók (a, m)-mel, így az $ax = km+1$ egyenletnek nincs megoldása (hiszen ekkor $(a, m) \mid ax - km = 1$, ami ellentmondás). Azaz az $ax \equiv 1 \pmod{m}$ kongruenciának sincs megoldása, vagyis nem létezik inverz. A továbbiakban azt az esetet nézzük, amikor $(a, m) = 1$.
Következik $T(a^{-1} \equiv? \pmod{m})$ becslése. Feltesszük $0 < a < m$ és $(a, m) = 1$. Ekkor a 6.21. Tételt alkalmazva kapjuk, hogy $\exists\, x, y$

$$ax + my = (a, m) = 1$$

és x, y $O((\log m)^3)$ bitoperációval megtalálható. Ekkor

$$ax \equiv 1 \pmod{m},$$
$$x \equiv a^{-1} \pmod{m}.$$

6.23. Tétel. *Ha p prím, akkor $\forall \neq 0$ maradékosztálynak létezik multiplikatív inverze, és ez $O((\log p)^3)$ lépésben meghatározható.*

6.24. Következmény. *$\mathbb{Z}_p$ test, melyet $\mathbb{F}_p$-vel jelölünk.*

Lineáris kongruenciák megoldására vonatkozó tétel az elemi számelméletből:

6.25. Tétel. *Az*

$$ax \equiv b \pmod{m}$$

lineáris kongruencia megoldható $\Leftrightarrow (a,m) \mid b$. *Ha ez teljesül, akkor ez a lineáris kongruencia ekvivalens egy*

$$a'x \equiv b' \pmod{m'},$$

ahol

$$a' = \frac{a}{(a,m)}, \quad m' = \frac{m}{(a,m)}, \quad b' = \frac{b}{(a,m)} \quad (\Rightarrow \ (a',m') = 1)$$

alakú lineáris kongruenciával. $\Rightarrow$ *A megoldások egy maradékosztályt alkotnak* $\bmod m'$. *Ez a megoldás megkereshető* $O((\log m')^3)$ *bitoperációval.*

Hivatkozások

[1] N. Koeblitz, *A Course in Number Theory and Cryptography*, Springer, 1994.

6.3. Moduláris hatványozás

Ismert, hogy $T(a^n) = O\big(n^2(\log a)^2\big)$.

Mod m ugyanez sokkal gyorsabban számolható:

6.26. Tétel. $T(a^n \bmod m) = O\big((\log n)(\log m)^2\big)$.

A 6.26. Tétel bizonyítása. Írjuk n-et diadikus alakban

$$n = \varepsilon_0 + \varepsilon_1 \cdot 2 + \varepsilon_2 \cdot 2^2 + \ldots + \varepsilon_k 2^k.$$

ahol $\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{k-1} \in \{0,1\}$ és $\varepsilon_k = 1$. Ekkor $k = O(\log n)$. Az ε_i számjegyek meghatározására:

$$T(\text{bináris alak}) = O((\log n)^2).$$

Legyen $i = 0, 1, \ldots, k$ esetén a_i az a^{2^i} szám maradéka mod m. Azaz:

$$a_i \equiv a^{2^i} \pmod{m} \quad 0 \leq a_i < m,$$

ahol $i = 0, 1, \ldots, k$. Határozzuk meg ezeket a számokat sorban:

$$a_0 = a^{2^0} = a^1 = a.$$

Ha a_i adott, a_{i+1}

$$a_{i+1} \equiv a^{2^{i+1}} \equiv a^{2^i 2} \equiv \left(a^{2^i}\right)^2 \equiv {a_i}^2 \pmod{m_i}$$

Ekkor $0 \leq a_i < m$ miatt

$$T({a_i}^2) = O((\log m)^2).$$

Maradékos osztás m-mel:

$$T({a_i}^2 \pmod{m}) = O(\log(m^2) \log m) = O((\log m)^2),$$
$$T(a_0, a_1, \ldots, a_k) = O(k(\log m)^2) = O((\log n)(\log m)^2).$$

Továbbá:

$$a^n = a^{\sum_{i=1}^{k} \varepsilon_i 2^i} = \prod_{i=1}^{k} a^{\varepsilon_i 2^i} = \underbrace{\prod_{i=1}^{k} {a_i}^{\varepsilon_i}}$$

azon a_i-k szorzata, melyekre $\varepsilon_i = 1$.

Ezekkel sorban szorozva, legfeljebb $k = O(\log n)$ lépésben, melyek során mindig két darab $\leq m$ szorzata szerepel. Egy darab szorzás időigényére tudjuk, hogy

$$T(\text{szorzás}) = O((\log m)^2).$$

Tehát $T(a^n \bmod m) = O\big((\log n)(\log m)^2\big)$.

Hivatkozások

[1] N. Koeblitz, *A Course in Number Theory and Cryptography*, Springer, 1994.

7. Gyökvonás modulo p

A 3. fejezetben a Legendre-szimbólumról szóló résznél szó volt $x^2 \equiv a \pmod{p}$ megoldhatóságáról, de ha megoldható, hogyan lehet egy x megoldást megtalálni? Ebben a fejezetben a fenti problémára két polinomiális algoritmust is ismertetünk.

7.1. Tonelli-Shanks algoritmusa

Az első szóban forgó algoritmus az ún. Tonelli–Shanks algoritmus. Az itt tárgyalt változatot Daniel Shanks [3] fejlesztette ki 1973-ban, aki kifejtette:

„Azért késtem leírni a történelmi utalásokat, mert kölcsönadtam egy barátomnak a Dickson's History 1. kötetét, és soha nem kaptam vissza."

Tehát Dickson könyve szerint az algoritmus redundánsabb változata már 1891-ben létezett, és Tonelli [1] nevéhez kötődik. A jegyzetben az algoritmust Koeblitz [2] könyve alapján ismertetem.

Tehát a *probléma:* Adott $p > 2$ prímre, $a \in \mathbb{Z}_p$, melyre $(a,p) = 1$ és $\left(\frac{a}{p}\right) = +1$. Keressük $x^2 \equiv a \pmod{p}$ megoldását.

Legyen $p - 1 = 2^\alpha s$, $\alpha \in \mathbb{N}$, s páratlan. Egy algoritmust adunk x meghatározására, melynek hossza α (tehát ha $\alpha = 1$, azaz $p - 1 = 2(2k+1) = 4k + 2 \Leftrightarrow p = 4k + 3$, egy lépésben kapjuk az eredményt).

Tekintsünk egy n kvadratikus nem-maradékot (Riemann-sejtés mellett polinomiális időben található ilyen n, később visszatérünk erre). Számítsuk ki $b \overset{\text{def}}{\equiv} n^s = n^{(p-1)/2^\alpha} \pmod{p}$-t. Majd $r \overset{\text{def}}{\equiv} a^{(s+1)/2} \pmod{p}$-t.

7.1. Lemma. *$r^2 a^{-1}$ szám $2^{\alpha-1}$-edik egységgyök modulo p.*

A 7.1. Lemma bizonyítása. Az a számból akarunk gyököt vonni, tehát a kvadratikus maradék, és ekkor az Euler-lemma miatt $a^{(p-1)/2} \equiv 1 \pmod{p}$. A fentieket és az r szám definícióját használva:

$$(r^2 a^{-1})^{2^{\alpha-1}} = \left(a^{\frac{s+1}{2}\cdot 2} a^{-1}\right)^{2^{\alpha-1}} = (a^s)^{2^{\alpha-1}} = a^{\frac{p-1}{2}} \equiv 1 \pmod{p}.$$

Megjegyzés. Ha speciel $\alpha = 1$, akkor $2^{\alpha-1} = 2^0 = 1$, tehát $r^2a^{-1} \equiv 1 \pmod{p}$, vagyis az a szám gyöke r és kész.

De mi van, ha $\alpha > 1$?

Tudjuk r^2a^{-1} az $2^{\alpha-1}$-dik egységgyök. Így

$$r^2a^{-1} \equiv \omega \pmod{p},$$
$$\omega^{-1}r^2 \equiv a \pmod{p}.$$

Amiből látszik, hogy

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

megoldását $r\varepsilon$ alakban kell keresni, ahol ε egy 2^α-dik egységgyök. Most lesz szükségünk a b számra, amelynek a definíciója $b \stackrel{\text{def}}{=} n^s$ volt. Ekkor:

7.2. Lemma. *A b szám 2^α-adik primitív egységgyök modulo p.*

A 7.2. Lemma bizonyítása. a) A b szám 2^α-adik egységgyök:

$$b^{2^\alpha} = (n^s)^{2^\alpha} = n^{2^\alpha s} = n^{p-1} \equiv 1 \pmod{p}.$$

b) A b szám primitív 2^α-adik egységgyök: Indirekt, tegyük fel, hogy nem primitív, azaz a rendje $< 2^\alpha$. Mivel ez a rend 2^α-nak osztója és $< 2^\alpha$, így $2^{\alpha-1}$-nek is osztója. Vagyis a rend alaptulajdonságai miatt $b^{2^{\alpha-1}} \equiv 1 \pmod{p}$ is teljesül. Viszont az Euler lemma miatt:

$$b^{2^{\alpha-1}} \equiv (n^s)^{2^{\alpha-1}} \equiv n^{\frac{p-1}{2}} \equiv \left(\frac{n}{p}\right) \equiv -1 \pmod{p},$$

ami ellentmondás.

Visszatérve x megoldást $r\varepsilon$ alakban keressük, ahol ε alkalmas 2^α-adik egységgyök. Vagyis $\varepsilon = b^j$ alakú, ahol $0 \le j < 2^\alpha$. Ekkor $x = rb^j$, tehát olyan j-t keresünk, hogy

$$x^2a^{-1} \equiv (rb^j)^2a^{-1} \equiv 1 \pmod{p}.$$

Ezt a j-t bináris alakban felírva keressük.

$$j = j_0 + 2j_1 + 2^2 j_2 + \dots$$

itt $j < 2^\alpha$ miatt $2^{\alpha-1} j_{\alpha-1}$ kellene, hogy legyen az utolsó tag.

De b 2^α-adik primitív egységgyök

$$\left(b^{2^{\alpha-1}}\right)^2 \equiv 1 \pmod{p}$$

$$b^{2^{\alpha-1}} \equiv \begin{cases} -1 \\ +1 \end{cases}.$$

A 7.2. lemma miatt

$$b^{2^{\alpha-1}} \equiv -1 \pmod{p}. \tag{7.1}$$

Ezért ha j bináris alakja $2^{\alpha-1}$-gyel végződik, azaz $j_{\alpha-1} = 1 \Rightarrow$ ezt eldobva j-ből: $j' = j - 2^{\alpha-1}$ olyan, hogy

$$b^j \equiv b^{j'} \underbrace{b^{2^{\alpha-1}}}_{-1} \equiv -b^{j'} \pmod{p},$$

tehát ha $x = rb^j$ jó, $x = rb^{j'}$ is. Ezért feltehető, hogy csak $2^{\alpha-2}$-ig megyünk. Itt a j-ket rekurzívan j_0-tól kezdve határozzuk meg.

Első lépés: j_0 meghatározása. A 7.1. Lemma alapján:

$$(r^2 a^{-1})^{2^{\alpha-1}} \equiv 1 \pmod{p},$$
$$\left((r^2 a^{-1})^{2^{\alpha-2}}\right)^2 \equiv 1 \pmod{p},$$
$$(r^2 a^{-1})^{2^{\alpha-2}} \equiv \pm 1 \pmod{p}.$$

Legyen

$$j_0 \stackrel{\text{def}}{=} \begin{cases} 0, & \text{ha } (r^2 a^{-1})^{2^{\alpha-2}} \equiv 1 \pmod{p}, \\ 1, & \text{ha } (r^2 a^{-1})^{2^{\alpha-2}} \equiv -1 \pmod{p}. \end{cases} \tag{7.2}$$

Ekkor (7.1)-t felhasználva, majd pedig (7.2)-t:

$$\left((b^{j_0} r)^2 a^{-1}\right)^{2^{\alpha-2}} = b^{j_0 \cdot 2^{\alpha-1}} (r^2 a^{-1})^{2^{\alpha-2}} \equiv (-1)^{j_0} (r^2 a^{-1})^{2^{\alpha-2}} \equiv 1 \pmod{p}.$$

Most tegyük fel, hogy $1 \leq k \leq \alpha - 2$ és $j_0, j_1, \ldots. \; j_{k-1}$ már adott úgy, hogy

$$\left(\left(b^{j_0+2j_1+\ldots+2^{k-1}j_{k-1}}r\right)^2 a^{-1}\right)^{2^{\alpha-k-1}} \equiv 1 \pmod{p}. \tag{7.3}$$

Feladat: j_k-t meghatározni úgy, hogy (7.3) $k-1$ helyén k-val is teljesüljön. Ehhez

$$\left(\left(\left(b^{j_0+2j_1+\ldots+2^{k-1}j_{k-1}}r\right)^2 a^{-1}\right)^{2^{\alpha-k-2}}\right)^2 \equiv 1 \pmod{p}.$$

Itt a külső zárójelben lévő rész (jelöljük w-vel) $+1$-gyel vagy -1-gyel kongruens modulo p. Ha erre $w \equiv +1 \pmod{p}$, $j_k \stackrel{\text{def}}{=} 0$, míg ha $w \equiv -1 \pmod{p}$, $j_k \stackrel{\text{def}}{=} 1$.

Ekkor valóban

$$\begin{aligned}
&\left(\left(b^{j_0+2j_1+\ldots+2^{k-1}j_{k-1}+2^k j_k}r\right)^2 a^{-1}\right)^{2^{\alpha-k-2}} \equiv \\
&\equiv \left(b^{2^{k+1}j_k}\right)^{2^{\alpha-k-2}} \left(\left(b^{j_0+2j_1+\ldots+2^{k-1}j_{k-1}}r\right)^2 a^{-1}\right)^{2^{\alpha-k-2}} \\
&\equiv \left(b^{2^{\alpha-1}}\right)^{j_k} \left(\left(b^{j_0+2j_1+\ldots+2^{k-1}j_{k-1}}r\right)^2 a^{-1}\right)^{2^{\alpha-k-2}} \\
&\equiv (-1)^{j_k} \left(\left(b^{j_0+2j_1+\ldots+2^{k-1}j_{k-1}}r\right)^2 a^{-1}\right)^{2^{\alpha-k-2}} \equiv 1 \pmod{p}.
\end{aligned}$$

Végén $k = \alpha - 2$-t véve kapjuk $j_{\alpha-2}$-t, és ekkor

$$\left(b^{j_0+2j_1+\ldots+2^{\alpha-2}j_{\alpha-2}}r\right)^2 a^{-1} \equiv 1 \pmod{p}.$$

Tehát $x \equiv b^j r \pmod{p}$ (ahol $j = j_0 + 2j_1 + \ldots + 2^{\alpha-2}j_{\alpha-2}$) megoldása az $x^2 \equiv a \pmod{p}$ kongruenciának.

Hivatkozások

[1] L. E. Dickson, *History of the Theory of Numbers*, Vol. 1, Washington, Carnegie Institution of Washington, 1919, 215–216.

[2] N. Koeblitz, *A Course in Number Theory and Cryptography*, Springer, 1994.

[3] D. Shanks, *Five number-theoretic algorithms*, Proceedings of the Second Manitoba Conference on Numerical Mathematics (1973), 51–70.

7.2. Perelta algoritmusa

Ebben a fejezetben a négyzetgyökvonásra modulo p adunk egy alternatív trükkös módszert, Perelta [2] algoritmusát. Az algoritmust Robin Chapman jegyzete [1] alapján ismertetjük, aki Perelta algoritmusának egy nagyon ügyes mátrixokkal való leírását adta meg.

Azt mondjuk két egész számokból álló mátrix, A és B kongruensek modulo p, ha a megfelelő elemeik kongruensek modulo p. Jelölése:

$$A \equiv B \pmod{p}.$$

Mátrixok kongruenciái ugyanúgy kezelhetőek, ahogy az egész számok esetében.

Legyen p páratlan prím, a pedig egy kvadratikus maradék, azaz $\left(\frac{a}{p}\right) = 1$. Tegyük fel, hogy az

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

kongruenciát szeretnénk megoldani.

Első lépésben keresünk egy b egész számot, amelyre

$$\left(\frac{b^2 - a}{p}\right) = -1. \tag{7.4}$$

Ezt véletlen módszerekkel érjük el, annak az esélye, hogy egy véletlenül választott b-re $b^2 - a$ kvadratikus nem maradék körülbelül $1/2$. Így ezt a lépést

párszor megismételve, előbb-utóbb eljutunk egy olyan b maradékosztályhoz, amelyre (7.4) fennáll. Definiáljuk az A és B mátrixokat az

$$A = \begin{bmatrix} 0 & 1 \\ a & 0 \end{bmatrix}, \qquad B = bI + A = \begin{bmatrix} b & 1 \\ a & b \end{bmatrix},$$

ahol I a 2×2-es egységmátrix. Ezek után számoljuk ki a $B^{(p-1)/2}$ mátrixot modulo p, ismételt négyzetre emeléssel. Csodálatosképpen azt találjuk, hogy

$$B^{(p-1)/2} \equiv \begin{bmatrix} 0 & r \\ s & 0 \end{bmatrix} \pmod{p},$$

ahol $s^2 \equiv a \pmod{p}$.

Nézzük miért is működik ez az algoritmus. Először számítsuk ki az $B^p = (bI + A)^p$ mátrixot a binomiális tétel alapján:

$$B^p = \sum_{j=0}^{p} \binom{p}{j} b^{p-j} A^j \equiv b^p I + A^p \pmod{p}, \tag{7.5}$$

mivel $1 \leq j \leq p-1$ esetén a $\binom{p}{j}$ binomiális együttható osztható p-vel. A kis-Fermat Tétel szerint $b^p \equiv b \pmod{p}$. Az Euler lemma szerint $a^{(p-1)/2} \equiv \left(\frac{a}{p}\right) = 1 \pmod{p}$ (ld. 3. fejezet). Egyszerű mátrix szorzás mutatja, hogy $A^2 = aI$, így

$$A^p = A \cdot \left(A^2\right)^{(p-1)/2} = A \cdot (aI)^{(p-1)/2} = a^{(p-1)/2} A \equiv A \pmod{p}.$$

A fentieket összevetve (7.5)-mal:

$$B^p \equiv bI + A = B \pmod{p}. \tag{7.6}$$

Mivel a B mátrix determinánsa nem nulla modulo p, így létezik modulo p vett inverze B^*. Az (7.6) kongruenciát B^*-gal szorozva

$$B^{p-1} \equiv I \pmod{p} \tag{7.7}$$

adódik. Az $A^2 = aI$ egyenletet használva teljes indukcióval igazolható, hogy minden n természetes számra

$$B^n = x_n I + y_n A$$

alakú, ahol x_n és y_n egész számok. Valóban az indukció kezdőlépése $n = 1$ esetén $B = bI + A$ esetén $x_1 = b, y_1 = 1$. Tegyük fel, hogy az állítást beláttuk n-re, és most bebizonyítjuk $n + 1$-re:

$$\begin{aligned} B^{n+1} = B^n \cdot B = (x_n A + y_n I) \cdot (bA + I) = x_n b A^2 + (x_n + y_n b)A + y_n I \\ = x_n abI + (x_n + y_n b)A + y_n I = (x_n + y_n b)A + (x_n ab + y_n), \end{aligned}$$

így $x_{n+1} = x_n + y_n b$ és $y_n = x_n ab + y_n$ jó választás. Állításunkat $B^{(p-1)/2}$ felírva kapjuk, hogy létezik t és r, amelyre

$$B^{(p-1)/2} = tI + rA = \begin{bmatrix} t & r \\ ar & t \end{bmatrix}. \tag{7.8}$$

Emlékezzünk vissza arra, hogy $B^{p-1} = I$ (ld. (7.7)), viszont (7.8) mátrix négyzetének jobb felső eleme a $2rt$ szám, így azt kapjuk

$$2rt \equiv 0 \pmod{p},$$

amiből $r \equiv 0 \pmod{p}$ vagy $t \equiv 0 \pmod{p}$. Először kizárjuk azt az esetet, hogy $r \equiv 0 \pmod{p}$. Valóban, ha $r \equiv 0 \pmod{p}$, akkor (7.8) alapján $B^{(p-1)/2} \equiv tI \pmod{p}$. Így:

$$I \equiv B^{p-1} = (B^{(p-1)/2})^2 \equiv (tI)^2 = t^2 I \pmod{p}.$$

Azaz $t^2 \equiv 1 \pmod{p}$. Ekkor $\det\left(B^{(p-1)/2}\right) \equiv t^2 \equiv 1 \pmod{p}$, viszont a determinánsok szorzástétele miatt

$$\det\left(B^{(p-1)/2}\right) = (\det B)^{(p-1)/2} = (b^2 - a)^{(p-1)/2} \equiv \left(\frac{b^2 - a}{p}\right) \equiv -1 \pmod{p},$$

ami ellentmondás. Tehát $r \not\equiv 0 \pmod{p}$, s így $t \equiv 0 \pmod{p}$. Ekkor:

$$B^{(p-1)/2} \equiv rA = \begin{bmatrix} 0 & r \\ ra & 0 \end{bmatrix} \pmod{p}.$$

Így:

$$I \equiv B^{p-1} \equiv (rA)^2 = r^2 aI \pmod{p}.$$

Azaz $r^2 a \equiv 1 \pmod{p}$. Az s definíciója alapján $s \equiv ra \pmod{p}$, így $s^2 \equiv r^2 a^2 \equiv a \pmod{p}$, amivel a bizonyítást befejeztük.

Hátra van még annak bizonyítása, hogy a maradékosztályok legalább felére $b^2 - a$ kvadratikus nem-maradék, a másik felére kvadratikus maradék, kivéve az $x^2 \equiv a \pmod{p}$ kongruencia két megoldását s és $-s$-t, amikor is $b^2 - a \equiv 0 \pmod{p}$. Ehhez tekintsük a

$$\sum_{b=0}^{p-1} \left(\frac{b^2 - a}{p} \right)$$

szummát. Ha ez -1, akkor készen vagyunk. Ehhez írjunk a helyébe s^2-et, majd használjuk a következő azonosságot:

$$\begin{aligned} \sum_{b=0}^{p-1} \left(\frac{b^2 - a}{p} \right) &= \sum_{b=0}^{p-1} \left(\frac{b^2 - s^2}{p} \right) = \sum_{b=0}^{p-1} \left(\frac{(b-s)(b+s)}{p} \right) \\ &= \sum_{b=0,\ b \neq -s}^{p-1} \left(\frac{(b-s)/(b+s)}{p} \right) \end{aligned}$$

Könnyű ellenőrizni, hogy ahogy b fut az utolsó szumman $(b-s)/(b+s)$ egy teljes maradékrendszer elemeit veszi fel kivéve az 1-et. Ezért:

$$\sum_{b=0}^{p-1} \left(\frac{b^2 - a}{p} \right) = -1,$$

és ezzel az utolsó állításunkat is beláttuk.

Hivatkozások

[1] R. Chapman, *Perelta's algorithm*, https://empslocal.ex.ac.uk/people/staff/rjchapma/courses/nt13/peralta.pdf.

[2] R. C. Perelta, *A simple and fast probabilistic algorithm for computing square roots modulo a prime number*, I. E. E. E. Trans. Inform. Theory 32 (1986), 846-847.

7.3. Legkisebb kvadratikus nem-maradék

Végül a fejezet végén pár szó arról, hogy milyen módszerekkel érdemes kvadratikus nem-maradékot keresni modulo p. Minden bizonnyal a leghatékonyabb a véletlen módszer. Választunk egy véletlen n-et, és megnézzük, hogy a $\left(\frac{n}{p}\right)$ Legendre szimbólum értéke -1 lesz-e vajon. Ha igen, akkor n kvadratikus nem-maradék modulo p. Mivel modulo p összesen $(p-1)/2$ darab kvadratikus maradék és $(p-1)/2$ kvadratikus nem-maradék van, ezért ha $p \nmid n$, akkor az esély arra, hogy n kvadratikus nem-maradék pont 50%. Vagyis, ha mondjuk 200-szor próbálkozunk különböző n-ekkel, akkor annak az esélye, hogy egyszer sem találunk kvadratikus nem-maradékot modulo p kisebb mint $\frac{1}{2^{200}}$. Ez még annál is kisebb valószínűség, hogy az ötöslottón háromszor egymás után megütjük a főnyereményt.

Noha ez a módszer a gyakorlati életben tökéletesen működik, az elméleti matematikusok szerettek volna egy determinisztikus algoritmust is kvadratikus nem-maradék konstruálására. Ehhez talán legegyszerűbb ötlet, hogy sorra próbálkozunk a természetes számokkal, 2,3,4,5,6,7, s.i.t. (valójában elég a prímszámokat nézni), és az első kvadratikus nem-maradéknál megállunk. Ahhoz, hogy ez az algoritmus polinomiális idejű legyen az szükséges, hogy a legkisebb kvadratikus nem-maradékra éles felső becslésünk legyen, amely $(\log p)^k$ alakú. Ez a probléma azonban önmagában is érdekes. Jelöljük tehát $n(p)$-vel a legkisebb pozitív kvadratikus nem-maradékot modulo p. Burgess [2] 1957-ben a következőt bizonyította:

$$\forall \varepsilon > 0 \ \exists p_0(\varepsilon), \text{ hogy ha } p > p_0(\varepsilon) \text{ akkor } n(p) < p^{1/(4\sqrt{e})+\varepsilon}.$$

Ez sajnos még nem győz meg minket arról, hogy létezik determinisztikus polinomiális idejű algoritmus kvadratikus nem-maradék megtalálására. Azonban

az általánosított Riemann hipotézist feltételezve, Bach [1] bebizonyította, hogy

$$n(p) \leq 2(\log p)^2.$$

Így feltételes eredményünk ugyan van, de mint köztudott se a Riemann-sejtést, se az általános Riemann-sejtést nem bizonyította be még senki.

Hivatkozások

[1] E. Bach, *Explicit bounds for primality testing and related problems*, Math. Comp. 55 (191) (1990), 355-380.

[2] D. A. Burgess, *The distribution of quadratic residues and non-residues*, Mathematika 4 (2) (1957), 106-112.

8. Gyors szorzás

Amikor egy-egy algoritmust leprogramoznak a program sebessége sok apróságtól függ, az egyik ilyen, hogy egy-egy standard aritmetikai műveletnek mekkora az időigénye. Láttuk, hogy

$$T(a+b) = O(\max\{\log a, \log b\}),$$

és az iskolában tanult szorzást használva a szorzásra

$$T(a \times b) = O(\max\{\log a, \log b\}).$$

A hatvanas illetve hetvenes években találtak ennél gyorsabb módszereket a szorzás algoritmusára, a modern számítógépek már ezeket használják. Ebben a fejezetben ezeket a szorzásfajtákat ismertetjük Das [1] könyvére alapozva.

Hivatkozások

[1] A. Das, *Computational Number Theory*, CRC Press, 2013.

8.1. Karatsuba-Ofman szorzás

Az első gyors szorzás Karatsuba és Ofman algoritmusa [2], amelyet Das [1] könyve alapján ismertetünk.

Legyen a és b két n-jegyű szám B alapú számrendszerben. Az egyszerűség kedvéért tegyük fel, hogy n páros. Legyen $m = n/2$. Írjuk fel a-t és b-t

$$a = A_1 B^m + A_0, \quad b = B_1 B^m + B_0$$

alakban, ahol A_0, A_1, B_0, B_1 legfeljebb $m = n/2$ darab számjegyből álló számok. Ekkor

$$ab = (A_1 B_1) B^{2m} + (A_1 B_0 + A_0 B_1) B^m + A_0 B_0. \tag{8.1}$$

Elsőre úgy gondolnánk, hogy itt 4 darab szorzatot kell kiszámolni. Valójában azonban nekünk csak a következő 3 együttható kell: $A_1B_1, A_1B_0 + A_0B_1, A_0B_0$. Észrevehetjük, hogy

$$A_1B_0 + A_0B_1 = (A_1 + A_0)(B_1 + B_0) - A_1B_1 - A_0B_0.$$

Vagyis elég 3 darab szorzatot kiszámolni: $(A_1 + A_0)(B_1 + B_0)$, A_1B_1, A_0B_0. Sajnos az $(A_1 + A_0)(B_1 + B_0)$-ban két szorzótényező lehet $m + 1$-jegyű szám is. Ezen is segíthetünk, ugyanis

$$A_1B_0 + A_0B_1 = A_1B_1 + A_0B_0 - (A_1 - A_0)(B_1 - B_0).$$

Vagyis a következő 3 darab szorzást kell elvégeznünk: A_1B_1, A_0B_0, $(A_1 - A_0)(B_1 - B_0)$. A szorzótényezők mindegyike legfeljebb $m = \lceil n/2 \rceil$ jegyű. Így ha 2 darab n-jegyű szám összeszorzásának igényét t_n-nel jelöljük, akkor (8.1)-ben az együtthatók kiszámításának időigénye $3t_{\lceil n/2 \rceil} + O(n)$, mivel a 3 darab szorzás kiszámításának időigénye $3t_m = 3t_{\lceil n/2 \rceil}$ és van néhány összeadás és kivonás is, ezek időigénye $O(n)$. Vagyis a következő rekurziót kapjuk:

$$t_n \leq 3t_{\lceil n/2 \rceil} + O(n).$$

Ha $n = 2^k$ alakú, akkor

$$t_{2^k} \leq 3t_{2^{k-1}} + O(2^k).$$

Ekkor k-ra vonatkozó teljes indukcióval könnyen igazolható, hogy

$$t_{2^k} \leq 3^k + O(2^k).$$

Általános n-re t_n becslését a következőképpen kaphatjuk meg. Vegyük a legkisebb 2 hatványt, amely n-nél nagyobb vagyis

$$2^{k-1} < n \leq 2^k.$$

Ekkor

$$t_n \leq t_{2^k} \leq 3^k + O(2^k) \leq 3 \cdot 3^{k-1} + O(2^k) = 3 \cdot \left(2^{k-1}\right)^{\log 3 / \log 2} + O(n)$$

$= O(n^{\log 3/\log 2})$.

Azaz a Karatsuba-Ofman szorzás időigénye $O(n^{\log 3/\log 2})$. Már ez is lényegesen gyorsabb mint az iskolában tanult írásbeli szorzásnak az időigénye $O(n^2)$.

Hivatkozások

[1] A. Das, *Computational Number Theory*, CRC Press, 2013.

[2] A. Karatsuba, Yu. Ofman, *Multiplication of many digital numbers by automatic computers*, Doklady Akad. Nauk. SSSR, Vol. 145 (1962), 293-294.

8.2. Toom-Cook szorzás

Toom [2] és Cook [3] a következőképpen általánosította a Karatsuba-Ofman szorzást: Legyen a és b két n darab számjegyből álló szám. Legyen $m = \lceil n/3 \rceil$ és írjuk fel a-t és b-t

$$a = A_2R^2 + a_1R + A_0, \quad b = B_2R^2 + B_1R + B_0 \tag{8.2}$$

alakban, ahol $R = B^m$. Ekkor

$$c = ab = C_4R^4 + C_3R^3 + C_2R^2 + C_1R + C_0, \tag{8.3}$$

ahol

$$\begin{aligned} C_4 &= A_2B_2 \\ C_3 &= A_2B_1 + A_1B_2 \\ C_2 &= A_2B_0 + A_1B_1 + A_0B_2 \\ C_1 &= A_1B_0 + A_0B_1 \\ C_0 &= A_0B_0. \end{aligned}$$

Első ránézésre ez 9 darab szorzást jelent. Vagyis ha t_n-nel jelöljük az n-jegyű számok szorzásának időigényét, akkor a fentiekből (az összeadásokkal együtt) a

$$t_n \leq 9t_{\lceil n/3 \rceil} + O(n)$$

rekurzió következik. Azonban most is, hasonlóan a Karatsuba-Ofman algoritmushoz, (8.3)-ben az együtthatók kiszámításához elég 9-nél kevesebb szorzás is. Legyen

$$\begin{aligned} a(x) &\stackrel{\text{def}}{=} A_2x^2 + A_1x + A_0 \\ b(x) &\stackrel{\text{def}}{=} B_2x^2 + B_1x + B_0 \\ c(x) &\stackrel{\text{def}}{=} C_4x^4 + C_3x^3 + C_2x^2 + C_1x + C_0 = a(x)b(x) \end{aligned}$$

Itt x helyébe írhatunk bármilyen valós vagy komplex számot. Minden $k \in \mathbb{C}$-re fennáll a

$$c(k) = a(k)b(k) \tag{8.4}$$

összefüggés. Általában egy p polinom értelmezési tartománya $\mathbb{R}$ vagy $\mathbb{C}$, most viszont az értelmezési tartományt kibővítjük egy ∞ szimbólummal:

$$p(x) = r_nx^n + r_{n-1}x^{n-1} + \cdots + r_1x + r_0$$

esetén legyen $p(\infty) = r_n$. Nyilván ekkor a (8.4) képlet $k = \infty$ esetében is fennáll, azaz

$$c(\infty) = a(\infty)b(\infty).$$

A következőt fogjuk használni:

$$\begin{aligned} c(\infty) &= C_4 = A_2B_2 \\ c(0) &= C_0 = A_0B_0 \\ c(1) &= C_4 + C_3 + C_2 + C_1 + C_0 = (A_2 + A_1 + A_0)(B_2 + B_1 + B_0) \\ c(-1) &= C_4 - C_3 + C_2 - C_1 + C_0 = (A_2 - A_1 + A_0)(B_2 - B_1 + B_0) \\ c(-2) &= 16C_4 - 8C_3 + 4c_2 - 2C_1 + C_0 \end{aligned}$$

$$= (4A_2 - 2A_1 + A_0)(4B_2 - 2B_1 + B_0) \tag{8.5}$$

Ebből mátrixokat használva adódik

$$\begin{pmatrix} c(\infty) \\ c(0) \\ c(1) \\ c(-1) \\ c(-2) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 16 & -8 & 4 & -2 & 1 \end{pmatrix} \begin{pmatrix} C_4 \\ C_3 \\ C_2 \\ C_1 \\ C_0 \end{pmatrix}.$$

Beszorozva az inverz mátrixszal az

$$\begin{pmatrix} C_4 \\ C_3 \\ C_2 \\ C_1 \\ C_0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 2 & -1/2 & 1/6 & 1/2 & -1/6 \\ -1 & -1 & 1/2 & 1/2 & 0 \\ -2 & 1/2 & 1/3 & -1 & 1/6 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} c(\infty) \\ c(0) \\ c(1) \\ c(-1) \\ c(-2) \end{pmatrix}$$

összefüggés adódik. Azaz

$$\begin{aligned} C_4 &= c(\infty) \\ C_3 &= \left(12c(\infty) - 3c(0) + c(1) + 3c(-1) - c(-2)\right)/6 \\ C_2 &= \left(-2c(\infty) - 2c(0) + c(1) + c(-1)\right)/2 \\ C_1 &= \left(-12c(\infty) + 3c(0) + 2c(1) - 6c(-1) + c(-2)\right)/6 \\ C_0 &= c(0) \end{aligned} \tag{8.6}$$

Vagyis (8.3)-ban a C_4, C_3, C_2, C_1, C_0 kiszámításához elég 5 darab szorzatot kiszámolnunk, nevezetesen (8.6) jobboldalán álló összegekben a $c(\infty), c(0), c(1), c(-1), c(-2)$ számokat, melyek szorzat alakját (8.5) adja meg.

A Toom-Cook szorzást használva két n-jegyű szám összeszorzásának időigényére a

$$t_n \leq 5t_{\lceil n/3 \rceil} + O(n).$$

rekurziót kapjuk. Ha $n = 3^k$ alakú, akkor

$$t_{3^k} \leq 5t_{3^{k-1}} + O(3^k).$$

Ebből k-ra vonatkozó teljes indukcióval könnyen igazolható, hogy

$$t_{3^k} \leq 5^k + O(3^k).$$

Általános n-re t_n becslését a következőképpen kaphatjuk meg. Vegyük a legkisebb 3 hatványt, amely n-nél nagyobb vagyis

$$3^{k-1} < n \leq 3^k.$$

Ekkor

$$\begin{aligned} t_n \leq t_{3^k} \leq 5^k + O(3^k) \leq 5 \cdot 5^{k-1} + O(3^k) = 5 \cdot \left(3^{k-1}\right)^{\log 5/\log 3} + O(n) \\ = O(n^{\log 5/\log 3}). \end{aligned}$$

Azaz a Toom-Cook szorzás időigénye $O(n^{\log 5/\log 3})$. A Toom-Cook szorzás tovább általánosítható magasabb fokú polinomokra. Ha (8.2)-ben $m = \lceil n/k \rceil$ és

$$\begin{aligned} a &= A_{k-1}R^{k-1} + A_{k-2}R^{k-2} + \cdots + A_1R + A_0 \\ b &= B_{k-1}R^{k-1} + B_{k-2}R^{k-2} + \cdots + B_1R + B_0 \end{aligned}$$

alakú, ahol $R = B^m$, akkor

$$ab = C_{2k-1}R^{2k-1} + C_{2k-2}R^{2k-2} + \cdots + C_1R + C_0$$

szorzatban az együtthatók kiszámításához elegendő k darab szorzás. Így az időigény $O(n^{\log(2k-1)/\log k})$ lesz, ahol az O-ban szereplő konstans csak k-tól függ. Ha k-t n függvényében alkalmasan választjuk az elméletileg elérhető legjobb futási idő $O\left(n2^{5\sqrt{\log n}}\right)$, de a gyakorlati alkalmazások azt mutatják, hogy k-t nem érdemes 4-nél nagyobbnak választani.

Hivatkozások

[1] A. Das, *Computational Number Theory*, Discrete Mathematics and its Applications, CRC Press, 2013.

[2] A. L. Toom, *The complexity of a scheme of functional elements realizing the multiplication of integers*, Doklady Acad. Nauk SSSR, 4 (3) (1963), 714-716, 1963.

[3] S. A. Cook, *On the Minimum Computation Time of Functions*, PhD thesis, Department of Mathematics, Harvard University, 1966.

8.3. Gyors Fourier-Transzformáció

Schönhage és Strassen [3] 1971-ben megadott egy alternatív módszert a gyors szorzásra, mely polinomok kiértékelésére és interpolációra épült. Ennek az új módszernek az időigénye elérte az $O(n \log n \log \log n)$ határt, így gyakorlatilag a leggyorsabb algoritmus olyan egész számok összeszorzására, melyek mérete legalább egy-két ezer számjegyből áll. A teljes Schönhage-Strassen algoritmus egy kissé komplikált ahhoz, hogy ebben a bevezető jegyzetben teljes egészében tárgyaljuk, így itt annak kissé leegyszerűsített változatát mutatjuk csak be Das [1] könyve alapján.

Tegyük fel, hogy a és b n-jegyű számok egy B alapú számrendszerben, ahol B most kettőhatvány, azaz $B = 2^r$. Legyen

$$2^{t-1} < n \leq 2^t$$

és

$$N = 2^{t+1}.$$

(Azaz N a legkisebb kettőhatvány, amely $2n$ vagy annál nagyobb.) Írjuk fel a-t és b-t is a B alapú számrendszerben, úgyhogy mindkét szám elejére írunk jó pár nulla számjegyet azért, hogy mindkét számra úgy tekintsünk, hogy

pontosan N $(= 2^{t+1})$ darab számjegyből áll. Azaz:

$$a = a_{N-1}B^{N-1} + a_{N-2}B^{N-2} + \cdots + a_1B + a_0$$
$$b = b_{N-1}B^{N-1} + b_{N-2}B^{N-2} + \cdots + b_1B + b_0. \tag{8.7}$$

Mivel a és b eredetileg (a szám elejére írt nullák nélkül) csak $n \leq \frac{N}{2}$ jegyű volt, ezért (8.7)-ban tudjuk, hogy $\frac{N}{2} \leq i, j < N$ esetén $a_i = 0$ és $b_j = 0$. Mind az a, mind a b számhoz (8.7) alapján rendelhetünk egy-egy polinomot:

$$\overline{a}(x) \stackrel{\text{def}}{=} a_{N-1}x^{N-1} + a_{N-2}x^{N-2} + \cdots + a_1x + a_0$$
$$\overline{b}(x) \stackrel{\text{def}}{=} b_{N-1}x^{N-1} + b_{N-2}x^{N-2} + \cdots + b_1x + b_0.$$

Nyilván ekkor

$$a = \overline{a}(B) \text{ és } b = \overline{b}(B).$$

Az a és b számok szorzata

$$ab = c = c_{N-1}B^{N-1} + c_{N-2}B^{N-2} + \cdots + c_1B + c_0$$

alakba írható, ahol

$$c_k = \sum_{\substack{0 \leq i,j \leq N-1 \\ i+j=k}} a_i b_j.$$

Hasonlóan $\overline{a}(x)$ és $\overline{b}(x)$ polinomokhoz, definiálhatunk egy $\overline{c}(x)$ polinomot:

$$\overline{c}(x) \stackrel{\text{def}}{=} c_{N-1}x^{N-1} + c_{N-2}x^{N-2} + \cdots + c_1x + c_0.$$

Ekkor $\overline{c}(x) = \overline{a}(x)\overline{b}(x)$, továbbá

$$c = ab = \overline{c}(B) = \overline{a}(B)\overline{b}(B).$$

Legyen ω_N egy N-edik primitív egységgyök. (Ha a komplex számok testében számolunk, akkor ω_N vehető $e^{2\pi i/N}$-nek. De a komplex számoktól különböző testet is vehetünk, erre később még visszatérünk.)

8.1. Definíció. *Az* $(a_{N-1}, a_{N-2}, \ldots, a_1, a_0)$ *sorozat diszkrét Fourier transzformáltján (DFT) azt az* $(A_{N-1}, A_{N-2}, \ldots, A_1, A_0)$ *sorozatot értjük, ahol*

$$A_k \stackrel{\text{def}}{=} \sum_{j=0}^{N-1} \omega_N^{kj} a_j. \tag{8.8}$$

8.2. Megjegyzés. *Az eddigi jelöléseket használva*

$$A_k = \overline{a}\left(\omega_N^k\right).$$

Legyen a $(b_{N-1}, b_{N-2}, \ldots, b_1, b_0)$ sorozat diszkrét Fourier transzformáltja $(B_{N-1}, B_{N-2}, \ldots, B_1, B_0)$ továbbá a $(c_{N-1}, c_{N-2}, \ldots, c_1, c_0)$ sorozat diszkrét Fourier transzformáltja $(C_{N-1}, C_{N-2}, \ldots, C_1, C_0)$. Az eddigi jelöléseket használva

$$B_k = \overline{b}\left(\omega_N^k\right) \quad \text{és} \quad C_k = \overline{c}\left(\omega_N^k\right).$$

Mivel $\overline{a}(x)\overline{b}(x) = \overline{c}(x)$, így

$$C_k = \overline{c}\left(\omega_N^k\right) = \overline{a}\left(\omega_N^k\right)\overline{b}\left(\omega_N^k\right) = A_k B_k.$$

Így ha gyorsan és hatékonyan ki tudjuk számolni egy sorozat diszkrét Fourier transzformáltját (azaz $(A_{N-1}, A_{N-2}, \ldots, A_1, A_0)$ és $(B_{N-1}, B_{N-2}, \ldots, B_1, B_0)$ sorozatokat is), akkor n darab szorzást végezve megkapjuk a $(C_{N-1}, C_{N-2}, \ldots, C_1, C_0)$ sorozatot. Ha $(C_{N-1}, C_{N-2}, \ldots, C_1, C_0)$ sorozatokra végzünk még egy diszkrét Fourier transzformációt (legyen az így kapott sorozat $(\widetilde{C}_{N-1}, \widetilde{C}_{N-2}, \ldots, \widetilde{C}_1, \widetilde{C}_0)$), akkor visszakapjuk az eredeti sorozat elemeit, kicsit pontosabban:

$$(c_{N-1}, c_{N-2}, \ldots, c_1, c_0) = \frac{1}{N}(\widetilde{C}_0, \widetilde{C}_1, \ldots, \widetilde{C}_{N-1}, \widetilde{C}_N). \tag{8.9}$$

A fenti összefüggés könnyen igazolható:

$$\widetilde{C}_k = \sum_{j=0}^{N-1} \omega_N^{kj} C_j = \sum_{j=0}^{N-1} \omega_N^{kj} \left(\sum_{i=0}^{N-1} \omega_N^{ji} c_i \right)$$

$$= \sum_{i=0}^{N-1} \left(\sum_{j=0}^{N-1} \omega_N^{j(k+i)} \right) c_i. \tag{8.10}$$

Itt

$$\sum_{j=0}^{N-1} \omega_N^{j(k+i)} = \begin{cases} N & \text{ha } N \mid k+i \\ 0 & \text{ha } N \nmid k+i \end{cases} = \begin{cases} N & \text{ha } i = N-k \\ 0 & \text{ha } i \neq N-k \end{cases}$$

Ezt (8.10)-be írva kapjuk $\widetilde{C}_k = NC_{N-k}$, ami igazolja (8.9)-t.

Vagyis a fenti algoritmus hatékonysága azon múlik, hogy mennyire gyorsan tudjuk kiszámítani egy sorozat DFT-ját.

Következő feladatunk egy N hosszú sorozat DFT-jának kiszámítására kell gyors algoritmust adnunk, illetve meg kell határoznunk ennek az algoritmusnak az időigényét. Az egyszerűség kedvéért a sorozat hossza N legyen mindig kettőhatvány (ez simán feltehető, hiszen a fejezet elején N-et kettőhatványként adtuk meg). A hamarosan ismertetésre kerülő sorozat időigényét N hosszú sorozat esetén jelöljük T_N-nel.

A DFT kiszámítására adott algoritmusunk rekurzív lesz. $N = 2^0$ hosszú sorozatra az algoritmus nagyon egyszerű, mivel (8.8)-ben a szumma csak egytagú, és így az időigény is $T_1 = 1$. Nézzük a rekurzív lépést. Tegyük fel, hogy az algoritmust már megadtuk $N = 2^0, 2^1, \ldots, 2^{r-1}$ esetben, rendre $T_1, T_2, T_4, \ldots, T_{2^{r-1}}$ időigénnyel. Most megszeretnénk adni az algoritmust $N = 2^r$ esetben is, és miután megadtuk az algoritmust, szeretnénk becsülni annak időigényét T_{2^r}-t.

Legyen a sorozatunk $(a_{N-1}, a_{N-2}, \ldots, a_1, a_0)$ és a primitív N-edik egységgyök ω_N, amellyel a DFT $(A_{N-1}, A_{N-2}, \ldots, A_1, A_0)$ sorozat elemeit meghatározzuk, azaz

$$A_k = \sum_{j=0}^{N-1} \omega_N^{kj} a_j.$$

Az $(a_{N-1}, a_{N-2}, \ldots, a_1, a_0)$ sorozathoz tartozó

$$\overline{a}(x) = a_{N-1}x^{N-1} + a_{N-2}x^{N-2} + \cdots + a_1 x + a_0$$

polinomot felírjuk két polinom összegeként, nevezetesen

$$\overline{a}(x) = \overline{a_0}(x^2) + x\overline{a_1}(x^2),$$

ahol

$$\overline{a_0}(y) \stackrel{\text{def}}{=} a_{N-2}y^{N/2-1} + a_{N-4}y^{N/2-2} + a_{N-6}y^{N/2-3} + \cdots + a_2y + a_0$$
$$\overline{a_1}(y) \stackrel{\text{def}}{=} a_{N-1}y^{N/2-1} + a_{N-3}y^{N/2-2} + a_{N-5}y^{N/2-3} + \cdots + a_3y + a_1.$$

A két polinom együtthatóiból képezhetünk egy-egy $N/2$ hosszú sorozatot. Ezek:

$$(a_{N-2}, a_{N-4}, a_{N-6}, \ldots, a_2, a_0) \tag{8.11}$$

és

$$(a_{N-1}, a_{N-3}, a_{N-5}, \ldots, a_3, a_1). \tag{8.12}$$

Mivel N páros (sőt kettőhatvány), ezért ha ω_N primitív N-edik egységgyök, akkor ω_N^2 primitív $N/2$-edik egységgyök. Számítsuk ki az (8.11) és (8.12) $N/2$ hosszú sorozatok DFT-ját. Ekkor a következő két sorozatot kapjuk

$$\left(A^{(0)}_{\frac{N}{2}-1}, A^{(0)}_{\frac{N}{2}-2}, \ldots, A^{(0)}_1, A^{(0)}_0\right)$$
$$\left(A^{(1)}_{\frac{N}{2}-1}, A^{(1)}_{\frac{N}{2}-2}, \ldots, A^{(1)}_1, A^{(1)}_0\right)$$

Ekkor

$$A^{(0)}_k = \sum_{j=0}^{N/2-1} \left(\omega_N^2\right)^{kj} a_{2j}$$
$$A^{(1)}_k = \sum_{j=0}^{N/2-1} \left(\omega_N^2\right)^{kj} a_{2j+1}$$

Az $\overline{a_0}(y)$ és $\overline{a_1}(y)$ polinomokat használva

$$A^{(0)}_k = \overline{a_0}(\omega_N^{2k}), \quad A^{(1)}_k = \overline{a_1}(\omega_N^{2k}).$$

Mivel $\overline{a}(x) = \overline{a_0}(x^2) + x\overline{a_1}(x^2)$, $x = \omega_N^k$-t írva

$$\overline{a}(\omega_N^k) = \overline{a_0}(\omega_N^{2k}) + \omega_N^k\overline{a_1}(\omega_N^{2k}).$$

Azaz

$$A_k = A_k^{(0)} + \omega_N^k A_k^{(1)}.$$

Mivel $\omega_N^{N/2} = -1$ (az indoklás egyszerű: $0 = \omega_n^N - 1 = (\omega_N^{N/2} - 1)(\omega_N^{N/2} + 1)$, de ω_N primitív N-edik egységgyök, így $\omega_N^{N/2} - 1 \neq 0$, vagyis $\omega_N^{N/2} + 1 = 0$). Ezért ha k helyébe $N/2 + k$-t írunk:

$$A_{N/2+k} = \overline{a}(\omega_N^{N/2+k}) = \overline{a_0}(\omega_N^{N+2k}) + \omega_N^{N/2+k}\overline{a_1}(\omega_N^{N+2k}).$$

$\omega_N^N = 1$ és $\omega_N^{N/2} = -1$-et használva

$$A_{N/2+k} = \overline{a_0}(\omega_N^{2k}) - \omega_N^k \overline{a_1}(\omega_N^{2k}) = A_k^{(0)} - \omega_N^k A_k^{(1)}.$$

Összefoglalva: A sorozatok DFT-ját kiszámítjuk $2T_{N/2}$ idő alatt, utána N darab szorzást és összeadást használva megkapjuk az eredeti sorozat DFT-ját. Időigényt tekintve a következő rekurziót kapjuk

$$T_1 = 1$$
$$T_N \leq 2T_{N/2} + N.$$

Ebből teljes indukcióval könnyen igazolható, hogy $N \geq 2$ esetén

$$T_N \leq 2N \log_2 N.$$

A fentebb megadott algoritmus során eddig nem vettük figyelembe, hogy a komplex számok felett az N-edik egységgyökök végtelen tizedestörtek. Knuth [2] észrevétele szerint $N = 2^{t+1}$ esetén elég az algoritmus során $6(t+1)$ tizedesjegy pontosságig számolni. Az így kapott algoritmus időigénye $O(n \log n \log\log n \ldots \log_k n)$, ahol ezúttal $\log_k n$-nel jelöljük a k-szor iterált logaritmus függvényt $\log_k n = \log\log\ldots\log n$ és k a legkisebb pozitív egész szám, amelyre $\log_k n = \log\log\ldots\log n < 2$. Schönhage és Strassen [3] a következőt javasolta: $\mathbb{C}$ helyett használjuk az algoritmust $\mathbb{Z}_{2^s+1}$ ahol 2 primitív $2s$-edik egységgyök, alkalmas s-sel. Az így elérhető futási idő $O(n \log n \log\log n)$. A részleteket ebben a jegyzetben kihagyjuk.

Néhány megjegyzés a programozók számára: Kis számokra az iskoláskönyv a legjobb módszer. Kicsit nagyobbakra Karatsuba-Ofman, még nagyobbakra Toom-Cook módszer, a legnagyobb számokra pedig Schönhage és Strassen által javasolt FFT-alapú szorzás. Az FFT szorzás néhány tízezer jegynél válik a leggyorsabbá, így valójában a gyakorlatban ez a legelterjedtebb szorzás a nagy számokra.

Hivatkozások

[1] A. Das, *Computational Number Theory*, Discrete Mathematics and its Applications, CRC Press, 2013.

[2] D. Knuth, *A Számítógép-Programozás Művészete (2. kötet), Szeminumerikus Algoritmusok*, Műszaki Könyvkiadó, Budapest, 1994, 4. fejezet.

[3] A. Schönhage, V. Strassen, *Schnelle multiplikation großer zahlen*, Computing 7 (1971), 281-292.

9. Prímtesztek

A prímtesztek olyan algoritmusok, amelyek megállapítják egy adott számról, hogy az prímszám-e vagy sem. Ellentétben a faktorizációval, a prímtesztek nem adják meg az adott szám prímtényezős felbontását, csupán annak megállapítására szorítkoznak, hogy a szóban forgó szám összetett-e vagy prím. Prímtesztelésre sokféle algoritmus létezik, és ma már a legmodernebb prímtesztek polinomiális idejű algoritmusok. Azonban a faktorizációról viszont úgy sejtjük, hogy az nem-polinomiális algoritmus.

A gyakorlati alkalmazások során a leghatékonyabbak az ún. valószínűségi prímtesztek, amelyek ugyan nem teljes bizonyossággal, de nagyon nagy valószínűséggel meg tudják állapítani egy természetes számról, hogy prímszám-e vagy összetett. Csupán az utolsó fejezetben ismertetünk polinomiális idejű és determinisztikus prímtesztet, azonban csak vázlatosan, mivel az lényegesen komplikáltabb mint az előző prímtesztek. Futási idejét tekintve ez a determinisztikus teszt lassabb mint a valószínűségi tesztek, de még mindig polinomiális idejű.

9.1. Próbaosztás

A legegyszerűbb prímtesztet úgy nevezték el, hogy próbaosztás. Legyen a bemenetként megadott szám n, és azt kell eldöntenünk, hogy vajon n prímszám-e. A próbaosztás során $d = 2, 3, 4, \cdots [\sqrt{n}]$ számokra megnézzük, hogy d osztója-e n-nek. Amennyiben találunk osztót, n összetett szám. Ha semelyik ilyen d nem osztója n-nek, akkor n prímszám. A fenti okoskodásnak alapja az, hogy minden összetett n számnak van $[\sqrt{n}]$-nél nem nagyobb osztója, ugyanis $n = ab$ esetén a vagy b kisebb egyenlő mint $\sqrt{n}$. A próbaosztás nem polinomiális algoritmus, időigénye $[\sqrt{n}](\log n)^2$. Az algoritmus alapja a jól ismert Eratoszthenészi szita, amelyre a legkorábbi ismert utalás Gerasai Nikomákhosz, Bevezetés az aritmetikába [1] című művében található (i.e.

2. század), amely a cirénei Eratoszthenésznek tulajdonítja azt. Amennyiben egy adott határig az összes prímet meg szeretnénk határozni, úgy máig Eratoszthenész szitája a leggyorsabb.

Hivatkozások

[1] R. Hoche and ed., *Nicomachi Geraseni Pythagorei Introductionis Arithmeticae Libri II*, Leipzig: B.G. Teubner, 1866, o. 31.

9.2. Fermat prímteszt

A Fermat prímteszt alapja a kis-Fermat tétel.

9.1. Tétel. (kis-Fermat) *Ha p prím és az a egész számra $(a, p) = 1$, akkor*

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

Ezután ismertetjük a tesztet [1] és [2] alapján. A Fermat prímteszt a következő: Legyen a bemenetként megadott szám n.

1. lépés: Vegyünk egy véletlenül választott a egész számot, amelyre $n \nmid a$.

2. lépés: Ellenőrizzük, hogy vajon

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

teljesül-e. Ha $a^{n-1} \not\equiv 1 \pmod{n}$, akkor a kis-Fermat tétel alapján n összetett. Ha $a^{n-1} \equiv 1 \pmod{n}$ teljesül, visszatérünk az 1. lépéshez, egy másik véletlenül választott a egész számmal.

Amennyiben a tesztet elég sok véletlenül választott a-ra megismételjük, és mindig azt kapjuk, hogy $a^{n-1} \equiv 1 \pmod{n}$, akkor a „valószínűleg” prímszám. Ha egyetlen a-ra is azt kapjuk, hogy $a^{n-1} \not\equiv 1 \pmod{n}$, akkor (a kis-Fermat tétel alapján) n biztos, hogy összetett szám. Mivel ez a teszt soha nem

mondja ki, hogy egy szám biztosan prím, valószínűségelméleti prímtesztnek nevezik.

Manapság a leggyakrabban használt prímtesztek valószínűségelméleti prímtesztek, mivel ezek futási ideje lényegesen gyorsabb a determinisztikus prímtesztek futási idejénél. A determinisztikus tesztekről később lesz még szó. Most térjünk vissza a Fermat prímteszthez. A teszt soha nem mondja ki, hogy egy n szám biztosan prímszám. De vajon igaz-e egyáltalán a kis-Fermat tétel megfordítása, azaz igaz-e a következő

Kérdés: Ha egy adott n egész számra, minden $(a, n) = 1$ egész számra teljesül, hogy

$$a^{n-1} \equiv 1 \pmod{n},$$

akkor biztos-e hogy n prímszám?

Erre a kérdésre tagadó a válasz. Legyen például $n = 561 = 3 \cdot 11 \cdot 17$. Bebizonyítjuk, hogy ekkor minden $(a, 561) = 1$-re $a^{560} \equiv 1 \pmod{561}$. Az 561 összetett szám, hiszen $561 = 3 \cdot 11 \cdot 17$. Lássuk a bizonyítást:

Legyen $(a, 561) = 1$. Ekkor $561 = 3 \cdot 11 \cdot 17$ miatt $(a, 3) = (a, 11) = (a, 17) = 1$. Írjuk fel a kis-Fermat tétel $p = 3$-ra:

$$a^2 \equiv 1 \pmod{3}.$$

A fenti kongruenciát 280. hatványra emelve

$$a^{560} \equiv 1 \pmod{3}$$

Azaz $3 \mid a^{560} - 1$.

Írjuk fel a kis-Fermat tétel $p = 11$-re:

$$a^{10} \equiv 1 \pmod{11}.$$

A fenti kongruenciát 56. hatványra emelve

$$a^{560} \equiv 1 \pmod{11}$$

Azaz $11 \mid a^{560} - 1$.

Írjuk fel a kis-Fermat tétel $p = 17$-re:

$$a^{16} \equiv 1 \pmod{17}.$$

A fenti kongruenciát 35. hatványra emelve

$$a^{560} \equiv 1 \pmod{17}.$$

Azaz $17 \mid a^{560} - 1$. Vagyis $3 \cdot 11 \cdot 17 = 561 \mid a^{560} - 1$, és ez volt a bizonyítandó állítás.

9.2. Definíció. *Egy n pozitív egész számot **Carmichael-szám**nak nevezünk, ha n összetett szám, és minden $(a, n) = 1$ esetén*

$$a^{n-1} \equiv 1 \pmod{n}.$$

A legkisebb Carmichael-szám az 561. A következő fejezetben bővebben lesz szó a Carmichael számokról. Visszatérve a Fermat prímteszthez, láthatjuk, hogy az a Carmichael-számokat prímként valószínűsíti noha azok összetettek. Így a Fermat prímteszt-et nem soroljuk a legideálisabb prímtesztek közé. Ennek ellenére a teszt viszonylag ritkán téved, hiszen Pomerence [3] egy eredménye szerint az x-nél kisebb Carmichael-számok száma kevesebb mint $x^{1-o(1)}$.

A továbbiakban tegyük fel, hogy az n szám összetett, de nem Carmichael-szám. Azaz létezik egy b egész-szám, amelyre

$$b^{n-1} \not\equiv 1 \pmod{n}.$$

Ezután az $(a, n) = 1$ egész számokat két osztályba osztjuk. Az a számot cinkos-nak nevezzük, ha teljesül rá az

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

kongruencia. Az elnevezést az indokolja, hogy az a számra teljesül a kis-Fermat tétel, noha az n modulus összetett szám. Így a Fermat prímtesztet erre az a-ra próbálva, a teszt (tévedve) azt valószínűsíti, hogy n prímszám. Az a számot tanúnak nevezzük, ha

$$a^{n-1} \not\equiv 1 \pmod{n}.$$

Ezt az elnevezést az indokolja, hogy erre az a-ra próbálva a Fermat prímtesztet, megkapjuk, hogy n összetett szám. Azaz az a szám „tanúskodik" n összetett szám volta mellett.

9.3. Tétel. *Legyen n szám egy összetett szám, amely nem Carmichael-szám. Ekkor az $(a, n) = 1$, $1 \leq a \leq n$ számoknak legalább a fele tanú.*

Ez a tétel azt jelent, hogy egy rögzített a-ra próbálva a tesztet, annak a valószínűsége, hogy téved a teszt (azaz azt valószínűsíti, hogy n prím) legfeljebb 50%. Viszont ha két darab a-ra is kipróbáljuk a tesztet, annak a valószínűsége, hogy a teszt téved $\left(\frac{1}{2}\right)^2$, azaz 25%. r darab a-ra próbálva, a tévedés valószínűsége $\left(\frac{1}{2}\right)^r$. Ez nagyobb r-ekre már roppant kicsi valószínűség. Már 300 darab a esetén is, a tévedés valószínűsége kevesebb mint $\left(\frac{1}{2}\right)^{300}$, ami kisebb mint a világegyetemben létező összes atomok együttes számának a reciproka...

Lássuk a tétel bizonyítását.

A 9.3. Tétel bizonyítása. Mivel n szám nem Carmichael szám, ezért létezik legalább egy darab tanú. Legyen ez b. Ekkor

$$b^{n-1} \not\equiv 1 \pmod{n}. \tag{9.1}$$

Soroljuk fel a cinkosokat: $a_1, a_2, \ldots, a_r$. Ekkor

$$a_i^{n-1} \equiv 1 \pmod{n} \qquad (1 \leq i \leq r).$$

Így azonban (9.1) alapján:

$$(ba_i)^{n-1} \not\equiv 1 \pmod{n}, \qquad (1 \leq i \leq r).$$

Vagyis $ba_1, ba_2, \ldots, ba_r$ is tanú. Azaz legalább annyi tanú van mint cinkos, ezzel pedig a tétel állítását beláttuk.

Hivatkozások

[1] D. Knuth, *A Számítógép-Programozás Művészete (2. kötet), Szeminumerikus Algoritmusok*, Műszaki Könyvkiadó, Budapest, 1994, 4. fejezet.

[2] N. Koeblitz, *A Course in Number Theory and Cryptography*, 2nd Edition, Springer, 1994.

[3] C. Pomerance, *On the distribution of pseudoprimes*. Math. Comp. 37 (1981), 587–593

9.3. Carmichael-számok

A Carmichael-számokat az előző fejezetben már definiáltuk, de azért nem árt ismételni, így most is leírom a definíciót.

9.4. Definíció. *Egy n pozitív egész számot* ***Carmichael-szám****nak nevezünk, ha n összetett szám, és minden $(a, n) = 1$ esetén*

$$a^{n-1} \equiv 1 \pmod{n}.$$

Ahogy az egész számok növekszenek, a Carmichael-számok egyre ritkásabban helyezkednek el a számegyenesen. Ennek illusztrálására mutatunk egy táblázatot. Jelölje $C(x)$ az 1 és x közé eső Carmichael számok számát. Ekkor:

n	1	2	3	4	5	6	7	8	9	10	11	12
$C(10^n)$	0	0	1	7	16	43	105	255	646	1547	3605	8241

n	13	14	15	16	17	18	19
$C(10^n)$	19279	44706	105212	246683	585355	1401644	3381806

Nagyon hosszú ideig sejtés volt, hogy végtelen sok Carmichael szám létezik. Végül, Alford, Granville és Pomerance [1] bebizonyította, hogy

$$C(x) > x^{2/7}.$$

Ezt egy kicsit Harman [3] megjavította:

$$C(x) > x^{0.33336704}.$$

1956-ban Erdős [2] azt sejtette, hogy $C(x) < x^{1-o(1)}$. Erdős heurisztikus érvelését 1981-ben Pomerance [5] megjavította. Belátta, hogy

$$C(x) < x^{1-\frac{(1+o(1))\log\log\log x)}{\log\log x}}.$$

A Carmichael-számok alakjára vonatkozó legfontosabb eredmény a Korselt-kritérium [4].

9.5. Tétel. (A. Korselt, 1899) *Egy n pozitív egész szám, akkor és csak akkor Carmichael-szám, ha n négyzetmentes, és n minden p prímosztójára teljesül, hogy $p-1 \mid n-1$.*

A 9.5. Tétel bizonyítása. Először azt látjuk be, hogy ha n négyzetmentes szám, és n minden p prímosztójára $p-1 \mid n-1$, akkor n Carmichael-szám. Valóban, legyen n prímtényezős felbontása

$$n = p_1 p_2 \dots p_r,$$

ahol a $p_1, p_2, \dots, p_r$ prímek különbözőek. Legyen

$$(a, n) = 1.$$

Ekkor

$$(a, p_1) = (a, p_2) = \dots = (a, p_r) = 1.$$

Mivel $(a, p_i) = 1$ a kis-Fermat tétel miatt

$$a^{p_i - 1} \equiv 1 \pmod{p_i}. \tag{9.2}$$

Mivel $p_i - 1 \mid n - 1$, ezért létezik k_i pozitív egész szám, hogy

$$n - 1 = k_i(p_i - 1)$$

Így az (9.2) kongruenciát k_i-edik hatványra emelve kapjuk, hogy

$$\begin{aligned} a^{k_i(p_i-1)} &\equiv 1 \pmod{p_i} \\ a^{n-1} &\equiv 1 \pmod{p_i} \\ p_i &\mid a^{n-1} - 1. \end{aligned} \tag{9.3}$$

Mivel (9.3) fennáll $i = 1, 2, \ldots, r$-re, ezért

$$\begin{aligned} p_1 p_2 \cdots p_r &\mid a^{n-1} - 1 \\ n &\mid a^{n-1} - 1 \\ a^{n-1} &\equiv 1 \pmod{n}. \end{aligned} \tag{9.4}$$

Ekkor (9.4) kongruencia minden $(a, n) = 1$ egész számra fennáll, tehát a Carmichael-szám.

Ezután rátérünk a tétel bizonyításának második részére, nevezetesen ha n Carmichael-szám, akkor egyrészt n négyzetmentes, másrészt n minden p prímosztójára $p - 1 \mid n - 1$. Először azt látjuk be, hogy n négyzetmentes. Indirekten bizonyítunk, azaz feltesszük, hogy létezik p prím, amelyre

$$n = p^k m$$

alakú, ahol $k \geq 2$ egész szám és az m egész számra $(m, p) = 1$. Tekintsük azt az a egész számot, amelyre

$$a \equiv 1 + p \pmod{p^2} \qquad \text{és} \qquad a \equiv 1 \pmod{m}.$$

A Kínai-maradéktétel miatt ilyen a egész szám létezik. A binomiális tétel alapján

$$a^{n-1} \equiv (1 + p)^{n-1} \pmod{p^2}$$

$$a^{n-1} \equiv 1 + \binom{n-1}{1}p + \binom{n-1}{2}p^2 + \cdots + \binom{n-1}{n-1}p^{n-1} \pmod{p^2}$$
$$a^{n-1} \equiv 1 + (n-1)p \pmod{p^2}. \tag{9.5}$$

Másrészt n Carmichael-szám, így

$$a^{n-1} \equiv 1 \pmod{n}$$
$$n \mid a^{n-1} - 1$$
$$p^2 \mid a^{n-1} - 1$$
$$a^{n-1} \equiv 1 \pmod{p^2}.$$

Ezt (9.5)-gyel összevetve kapjuk, hogy

$$1 \equiv 1 + (n-1)p \pmod{p^2}$$
$$p^2 \mid (n-1)p$$
$$p \mid n-1,$$

ami ellentmond $p \mid n$-nek. Ezzel beláttuk, hogy n-nek nincs prímnégyzet osztója, azaz n valóban négyzetmentes.

Ezután rátérhetünk annak bizonyítására, hogy n minden p prímosztójára teljesül, hogy $p-1 \mid n-1$. Írjuk fel n-et $n = pm$ alakban. Mivel n négyzetmentes, feltehetjük, hogy $(m,p) = 1$. Tekintsünk egy g primitív gyököt modulo p. Ekkor $g, g^2, g^3, \ldots, g^{p-2} \not\equiv 1 \pmod{p}$, de $g^{p-1} \equiv 1 \pmod{p}$, mivel g rendje $p-1$. Azaz az $1, g, g^2, \ldots$ végtelen sorozat periodikus lesz modulo p, és a periódushossz $p-1$, vagyis

$$g^k \equiv 1 \pmod{p} \Leftrightarrow p-1 \mid k. \tag{9.6}$$

Tegyük fel, hogy minden $(a,n) = 1$ esetén

$$a^{n-1} \equiv 1 \pmod{n}.$$

A kínai maradéktétel szerint létezik olyan a egész szám, amelyre

$$a \equiv g \pmod{p} \quad \text{és} \quad a \equiv 1 \pmod{m}.$$

Mivel g primitív gyök $(g,p)=1$, és így $(g+tp,p)=1$ is teljesül minden t egész számra. Azaz $(a,p)=1$. Hasonlóan $a\equiv 1 \pmod{m}$ miatt $(a,m)=1$ is teljesül. Azaz $(a,n)=(a,pm)=1$. Tehát

$$\begin{aligned} a^{n-1} &\equiv 1 \pmod{n} \\ n \mid a^{n-1}-1 \\ pm \mid a^{n-1}-1 \\ p \mid a^{n-1}-1 \\ a^{n-1} &\equiv 1 \pmod{p} \\ g^{n-1} &\equiv 1 \pmod{p}. \end{aligned}$$

Ekkor (9.6)-t használva kapjuk, hogy $p-1 \mid n-1$, ami a bizonyítandó állítás volt.

Hivatkozások

[1] W. R. Alford, A. Granville and C. Pomerance, *There are infinitely many Carmichael numbers*, Annals of Mathematics, Second Series, 139, No. 3 (1994), 703-722.

[2] P. Erdős, *On pseudoprimes and Carmichael numbers*, Publ. Math. Debrecen 4 (1956), 201–206.

[3] G. Harman, *On the number of Carmichael numbers up to x*, Bulletin of the London Mathematical Society 37 (2005), 641–650.

[4] A. R. Korselt, *Problème chinois*, L'intermédiaire des mathématiciens 6 (1899), 142–143.

[5] C. Pomerance, *On the distribution of pseudoprimes*. Math. Comp. 37 (1981), 587–593.

9.4. Egy példa a Fermat prímtesztre

A Fermat prímteszt során nagy hatványokat kell kiszámolnunk modulo n. Lássunk egy példát:

Kérdés: $2^{11} - 1 = 2047$ prímszám-e vagy sem? Teszteljük a Fermat prímteszttel!

Megoldás: Az algoritmus a következő:

1. lépés: Vegyünk egy véletlen $a \in \mathbb{Z}_{2047}$, $2047 \nmid a$ elemet.

2. lépés: Ellenőrizzük, hogy vajon

$$a^{2046} \equiv 1 \pmod{2047} \tag{9.7}$$

fenn áll-e? Ha nem áll fenn, akkor 2047 összetett, ha fennáll térjünk vissza az 1. lépéshez, egy másik a számmal. Ha sok a-ra fennáll az (9.7) kongruencia, akkor 2047 valószínűleg prímszám.

Az algoritmus első lépese során választunk egy véletlen $a \in \mathbb{Z}_{2047}$, $2047 \nmid a$ számot. Az egyszerűség kedvéért tegyük fel, hogy $a = 2$ (bár, ennek a valószínűsége roppant kicsi 1/2046...) Az algoritmus 2. lépés során azt kell ellenőriznünk, hogy teljesül-e a

$$2^{2046} \equiv 1 \pmod{2047} \tag{9.8}$$

kongruencia. Vegyük észre, hogy $2^{11} = 2048 \equiv 1 \pmod{2047}$. Ennek alapján

$$2^{2046} = \left(2^{11}\right)^{186} \equiv 1 \pmod{2047}.$$

Vagyis (9.8) valóban teljesül. Azaz eddig nem derült ki, hogy a 2047 összetett szám. Vagyis a 2 egy Fermat-cinkos. Ha sok-sok a-ra teljesül az, hogy $a^{2046} \equiv 1 \pmod{2047}$, akkor azt valószínűsíthetnénk, hogy a 2047 prím szám vagy Carmichael szám. Egyelőre azonban csak az $a = 2$ esetet vizsgáltuk. Tegyünk még egy próbát egy újabb véletlen a-val. Akárcsak az előbb most is

minden $a \in \mathbb{Z}_{2047}$, $2047 \nmid a$-t egyforma valószínűséggel választhatunk, de most az egyszerűség kedvéért tegyük fel, hogy $a = 3$. (Valójában a gyakorlatban igen gyakran előfordul, hogy a programozók, amikor olyan véletlen számokat keresnek, amelyek nem titkosak, azaz nem használják azokat kriptográfiai célokra, akkor egyszerűen egymást követő természetes számokat vesznek... Ez sokat levon a véletlen algoritmus valódi hatékonyságából. Amennyiben az olvasó programozni szeretne, természetesen javasoljuk az eféle egyszerűsítések elkerülését, noha most egy ilyen példát mutatunk be...) Amennyiben $a = 3$, akkor azt kell ellenőriznünk, hogy

$$3^{2046} \equiv 1 \pmod{2047} \tag{9.9}$$

teljesül-e. Vagyis egy nagyon nagy hatványt kell kiszámolnunk moduláris hatványozással. Ez történhet az ismételt négyzetre emeléssel. Ehhez először írjuk fel a 2046-ot kettőhatványok összegeként.

$$2046 = 1024 + 512 + 256 + 128 + 64 + 32 + 16 + 8 + 4 + 2.$$

Majd készítsünk egy táblázatot a $3^{2^k} \pmod{2047}$ hatványok értékeivel.

	3^1	3^2	3^4	3^8	3^{16}	3^{32}	3^{64}	3^{128}	3^{256}	3^{512}	3^{1024}
$\underset{2047}{\text{mod}}$	3	9	81	420	358	1250	639	968	1545	223	601

Ez alapján:

$$3^{2046} \equiv 3^{1024} \cdot 3^{512} \cdot 3^{256} \cdot 3^{128} \cdot 3^{64} \cdot 3^{32} \cdot 3^{16} \cdot 3^8 \cdot 3^4 \cdot 3^2$$
$$601 \cdot 223 \cdot 1545 \cdot 968 \cdot 639 \cdot 1250 \cdot 358 \cdot 420 \cdot 81 \cdot 9 \cdot 3 \pmod{2047}.$$

A szorzásokat rendre elvégezve, és az eredményt mindig redukálva mod 2047, végeredményül azt kapjuk, hogy

$$3^{2046} \equiv 1013 \not\equiv 1 \pmod{2047}.$$

Vagyis 2047 nem prímszám, és a 3 egy Fermat-tanú, míg a 2 egy Fermat cinkos volt. Valóban 2047 összetett, hiszen $2047 = 11 \cdot 23$. Nagy számokra

azonban nehéz megtalálni a prímfaktorokat, míg a Fermat teszt futási ideje gyors: $O(k \cdot \log^3 n)$, ahol k a fordulók száma, azaz ennyi véletlen a alapra megy a tesztelés.

9.5. Soloway-Strassen prímteszt

Az előző fejezetekben szó volt a Fermat-prímtesztről. Láttuk, hogy a teszt nem működik a Carmichael-számok esetében, hiszen a teszt a Carmichael-számokat is prímként valószínűsíti. A következő teszt megértéséhez szükség van a Legendre és Jacobi szimbólum fogalmára, lásd 3. fejezet. Láttuk, hogy ha n prímszám, akkor az $\left(\frac{a}{n}\right)$ Jacobi szimbólum értéke megegyezik az $\left(\frac{a}{n}\right)$ Legendre szimbólum értékével. A Soloway-Strassen prímteszt [2], [3] a következő tételen alapul:

9.6. Tétel. (Euler-lemma) *Tetszőleges p páratlan prímszámra, és a egész számra, ahol $(a, p) = 1$ teljesül az*

$$a^{(p-1)/2} \equiv \left(\frac{a}{p}\right) \pmod{p}$$

kongruencia, ahol $\left(\frac{a}{p}\right)$ a Legendre szimbólum.

A 9.6. Tétel bizonyítása. A kis-Fermat tétel alapján

$$\begin{aligned} a^{p-1} &\equiv 1 \pmod{p} \\ p \mid a^{p-1} - 1 \\ p \mid \left(a^{(p-1)/2} - 1\right)\left(a^{(p-1)/2} + 1\right), \end{aligned}$$

így $p \mid a^{(p-1)/2} - 1$ vagy $p \mid a^{(p-1)/2} + 1$. Tehát

$$a^{(p-1)/2} \equiv 1 \pmod{p} \quad \text{vagy} \quad a^{(p-1)/2} \equiv -1 \pmod{p}. \tag{9.10}$$

Ha $\left(\frac{a}{p}\right) = 1$, akkor

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

megoldható. Jelölje a fenti kongruencia egy megoldását x_0. Ekkor

$$x_0^2 \equiv a \pmod{p}.$$

Vagyis a kis-Fermat tétel alapján

$$a^{(p-1)/2} \equiv \left(x_0^2\right)^{(p-1)/2} = x_0^{p-1} \equiv 1 \pmod{p}.$$

Mivel $\left(\frac{a}{p}\right) = 1$, így

$$a^{(p-1)/2} \equiv \left(\frac{a}{p}\right) \pmod{p}.$$

Ezután rátérünk $\left(\frac{a}{p}\right) = -1$ eset bizonyítására. A fokszám tétel a következőt mondja ki:

9.7. Lemma. *Ha p prímszám és $f(x) \in \mathbb{Z}[x]$, akkor vagy $f(x)$ minden együtthatója p-vel osztható, vagy az*

$$f(x) \equiv 0 \pmod{p}$$

kongruenciának legfeljebb $\deg f$ *inkongruens megoldása van, ahol* $\deg f$ *az $f(x)$ polinom fokszámát jelöli.*

A fokszám tételt itt nem bizonyítjuk, annak bizonyítása bármelyik elemi számelmélettel foglalkozó könyvben megtalálható. A fokszám tétel alapján tudjuk, hogy az

$$x^{(p-1)/2} \equiv 1 \pmod{p}$$

kongruenciának legfeljebb $(p-1)/2$ darab megoldása van. Láttuk, hogy $\left(\frac{a}{p}\right) = 1$ esetén

$$a^{(p-1)/2} \equiv 1 \pmod{p},$$

így az $x^{(p-1)/2} \equiv 1 \pmod{p}$ kongruenciának a kvadratikus maradék a-k a megoldásai. Tudjuk, hogy összesen $(p-1)/2$ darab kvadratikus maradék

van, így ezzel az $x^{(p-1)/2} \equiv 1 \pmod{p}$ kongruencia összes megoldását meghatároztuk. Tehát $\left(\frac{a}{p}\right) = -1$ esetén $a^{(p-1)/2} \not\equiv 1 \pmod{p}$. Ekkor (9.10) alapján

$$a^{(p-1)/2} \equiv -1 \pmod{p}.$$

Mivel most $\left(\frac{a}{p}\right) = -1$, így ekkor is igazoltuk az

$$a^{(p-1)/2} \equiv \left(\frac{a}{p}\right) \pmod{p}$$

kongruenciát. Ezzel az Euler-lemmát igazoltuk.

Ezután ismertetjük a **Soloway-Strassen prímtesztet** [2], [3]. A teszt mögött Artjuhov [1] egy ötlete állt.

Tehát szeretnénk meghatározni egy adott n páratlan számról, hogy prímszám-e vagy összetett.

1. lépés: Veszünk egy véletlen a-t, amelyre $(a, n) = 1$.

2. lépés: Az ismételt négyzetre emelés algoritmussal kiszámoljuk $a^{(n-1)/2} \pmod{n}$-t.

3. lépés: Kiszámoljuk az $\left(\frac{a}{n}\right)$ Jacobi szimbólum értékét.

4. lépés: Ellenőrizzük, hogy vajon az $a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \pmod{n}$ kongruencia fenn áll-e. Ha a válasz NEM, akkor biztos, hogy n összetett, és a-t Euler-tanú-nak nevezzük. Ha a válasz IGEN akkor visszatérünk az 1. lépéshez egy másik a-val.

Ha sok a-ra megnézzük a fenti algoritmust, és mindig az jön ki, hogy $a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \pmod{n}$, akkor azt valószínűsíthetjük, hogy n prímszám. Ha az n szám összetett, és fennáll az $a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \pmod{n}$ kongruencia, akkor a-t Euler-cinkos-nak nevezzük.

Példa: -1 mindig Euler-cinkos, ugyanis a Jacobi szimbólumra mindig fennáll az

$$\left(\frac{-1}{n}\right) \equiv (-1)^{(n-1)/2} \pmod{n}$$

összefüggés.

9.8. Tétel. *Minden összetett páratlan n-re a* mod n *redukált maradékosztályoknak legalább a fele Euler-tanú.*

A 9.8. Tétel bizonyítása. Legyenek az Euler-cinkosok: $\ell_1, \ell_2, \ldots, \ell_k$, az Euler-tanúk pedig: $w_1, w_2, \ldots, w_m$. Az a bizonyítandó, hogy $k \leq m$. Először csak azt bizonyítjuk, hogy legalább egy Euler-tanú létezik. Ha n nem négyzetmentes, akkor n nem lehet Carmichael-szám, azaz létezik egy a maradékosztály, amelyre $(a, n) = 1$ és $a^{n-1} \not\equiv 1 \pmod{n}$. Így

$$a^{(n-1)/2} \not\equiv \pm 1 \pmod{n}$$

(hiszen ha $a^{(n-1)/2} \equiv \pm 1 \pmod{n}$, akkor négyzetre emelés után $a^{n-1} \equiv 1 \pmod{n}$-et kapnánk). Viszont az $\left(\frac{a}{n}\right)$ Jacobi szimbólum értéke ± 1, vagyis

$$a^{(n-1)/2} \not\equiv \left(\frac{a}{n}\right) \pmod{n}.$$

Azaz ekkor a Euler-tanú. Ha n négyzetmentes, írjuk fel n-et

$$n = p_1 p_2 \ldots p_r$$

alakban. Rögzítsünk egy kvadratikus nem-maradékot mod p_1, legyen ez m. Vagyis $\left(\frac{m}{p_1}\right) = -1$. Legyen a megoldása az

$$x \equiv m \pmod{p_1} \qquad x \equiv 1 \pmod{p_2 p_3 \ldots p_r}$$

szimultán kongruencia-rendszernek. A Kínai maradéktétel szerint ilyen a létezik, és egyértelmű mod $p_1 p_2 \ldots p_r$. Vagyis

$$a \equiv m \pmod{p_1} \qquad a \equiv 1 \pmod{p_2 p_3 \ldots p_r}.$$

Számítsuk ki az $\left(\frac{a}{n}\right)$ Jacobi-szimbólum értékét:

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right)\left(\frac{a}{p_2}\right)\ldots\left(\frac{a}{p_r}\right)$$

$$= \left(\frac{m}{p_1}\right)\left(\frac{1}{p_1}\right)\cdots\left(\frac{1}{p_r}\right)$$
$$= -1\cdot 1\cdots 1$$
$$= -1.$$

Ha

$$a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \pmod{n}$$

teljesül, akkor

$$a^{(n-1)/2} \equiv -1 \pmod{n}$$
$$n \mid a^{(n-1)/2}+1$$
$$p_2 \mid a^{(n-1)/2}+1$$
$$a^{(n-1)/2} \equiv -1 \pmod{p_2}$$
$$1^{(n-1)/2} \equiv -1 \pmod{p_2},$$

ami ellentmondás. Vagyis $a^{(n-1)/2} \not\equiv \left(\frac{a}{n}\right) \pmod{n}$. Azaz a Euler-tanú. Ezzel beláttuk, hogy minden összetett szám esetén létezik legalább egy Euler-tanú. Legyenek az Euler-cinkosok: $\ell_1, \ell_2, \ldots, \ell_k$, az Euler-tanúk pedig: $w_1, w_2, \ldots, w_m$. Azt már láttuk, hogy $m \geq 1$, a tétel bizonyításához azonban az kell, hogy $k \leq m$. Rögzítsünk egy darab Euler-tanút, mondjuk w_1-et. Bebizonyítjuk, hogy ekkor $w_1\ell_1, w_1\ell_2, \ldots, w_1\ell_k$ is Euler-tanúk. Mivel a fenti elemek inkongruensek modulo n, ezért ebből az állításból már valóban következik, hogy $m \geq k$. A Jacobi szimbólum multiplikativitása miatt

$$\left(\frac{w_1\ell_i}{n}\right) = \left(\frac{w_1}{n}\right)\left(\frac{\ell_i}{n}\right). \tag{9.11}$$

Mivel ℓ_i Euler-cinkos így $\left(\frac{\ell_i}{n}\right) \equiv \ell_i^{(n-1)/2} \pmod{n}$. Viszont w_1 Euler-tanú, így $\left(\frac{w_1}{n}\right) \not\equiv w_1^{(n-1)/2} \pmod{n}$. A fentieket (9.11)-gyel összevetve adódik, hogy

$$\left(\frac{w_1\ell_i}{n}\right) = \left(\frac{w_1}{p}\right)\left(\frac{\ell_i}{p}\right) \not\equiv w_1^{(n-1)/2}\ell_i^{(n-1)/2} = (w_1\ell_i)^{(n-1)/2} \pmod{n},$$

vagyis $w_1\ell_i$ valóban Euler-tanú. Ezzel a tétel állítását beláttuk.

A fenti tétel alapján, ha k darab a-ra próbáljuk a tesztet, annak a valószínűsége, hogy n összetett szám, de a teszt prímként valószínűsíti kisebb egyenlő mint 2^{-k}. Ez már $k = 100$ esetén is roppant kicsi valószínűség.

Hivatkozások

[1] M. M. Artjuhov, *Certain criteria for primality of numbers connected with the little Fermat theorem*, Acta Arith. 12 (1966/67), 355–364.

[2] R. M. Solovay, V. Strassen, *A fast Monte-Carlo test for primality*, SIAM Journal on Computing. 6 (1) (1977), 84–85.

[3] R. M. Solovay, V. Strassen, *Erratum: A fast Monte-Carlo test for primality*, SIAM Journal on Computing. 7 (1) (1978) 118.

9.6. Egy példa a Soloway-Strassen tesztre

Az alábbiakban meghatározzuk, hogy $n = 209$ prímszám-e vagy összetett, a Soloway-Strassen teszttel.

1. lépés: Vegyünk egy véletlen a-t. Legyen mondjuk $a = 153$.

2. lépés: Számítsuk ki $a^{(n-1)/2} \pmod n$-et az ismételt négyzetre emelés algoritmussal.

$$153^{(209-1)/2} \equiv 153^{104} \equiv 153^{64} \cdot 153^{32} \cdot 153^{8} \pmod{209}.$$

Készítsünk egy táblázatot $153^{2^k} \pmod{209}$ értékeivel.

	153^1	153^2	153^4	153^8	153^{16}	153^{32}	153^{64}
mod 209	153	1	1	1	1	1	1

Ezek alapján

$$153^{(209-1)/2} \equiv 1 \cdot 1 \cdot 1 \equiv 1 \pmod{209}.$$

3. lépés: Számítsuk ki az $\left(\frac{a}{n}\right)$ Jacobi-szimbólum értékét.

$$\begin{aligned}\left(\frac{153}{209}\right) &= (-1)^{(153-1)/2\cdot(209-1)/2}\left(\frac{209}{153}\right) = \left(\frac{209}{153}\right) = \left(\frac{56}{153}\right)\\ &= \left(\frac{2}{153}\right)^3\left(\frac{7}{153}\right) = (-1)^{(153^2-1)/8}\left(\frac{7}{153}\right) = \left(\frac{7}{153}\right)\\ &= (-1)^{(7-1)/2\cdot(153-1)/2}\left(\frac{153}{7}\right) = \left(\frac{153}{7}\right) = \left(\frac{6}{7}\right)\\ &= \left(\frac{2}{7}\right)\left(\frac{3}{7}\right) = (-1)^{(7^2-1)/8}\left(\frac{3}{7}\right) = \left(\frac{3}{7}\right)\\ &= (-1)^{(3-1)/2\cdot(7-1)/2}\left(\frac{7}{3}\right) = -\left(\frac{7}{3}\right) = -\left(\frac{1}{3}\right)\\ &= -1.\end{aligned}$$

Vagyis $153^{(209-1)/2} \not\equiv \left(\frac{153}{209}\right) \pmod{209}$. Azaz 153 Euler-tanú, és így 209 összetett (valóban $209 = 11 \cdot 19$).

9.7. Miller-Rabin prímteszt

A Miller-Rabin prímtesztet először Gary L. Miller [5] fedezte fel 1976-ban. A teszt Miller-féle változata determinisztikus volt, ugyanakkor a teszt megbízhatóságát egy bizonyítatlan sejtés, az általános Riemann hipotézis támasztotta alá. Michael O. Rabin [6] úgy módosította a tesztet 1980-ban, hogy az nem függött tovább a bizonyítatlan sejtéstől, viszont így a teszt már nem determinisztikus volt, hanem valószínűségi prímteszté módosult.

Először is megjegyezzük, hogy ha p prím, akkor az

$$x^2 \equiv 1 \pmod{p}$$

kongruenciából következik az

$$x \equiv \pm 1 \pmod{p}$$

kongruencia. Lássuk ennek a bizonyítását:

$$\begin{aligned} x^2 &\equiv 1 \pmod{p} \\ p &\mid x^2 - 1 \\ p &\mid (x-1)(x+1) \\ p &\mid x-1 \text{ vagy } p \mid x+1 \\ x &\equiv \pm 1 \pmod{p}. \end{aligned}$$

Fontos megjegyezni, hogy a fenti bizonyításban szükséges, hogy p prímszám. Összetett számokra nem is igaz az állítás. (Vegyük például a 8-at. Az $x^2 \equiv 1 \pmod 8$ kongruenciának 4 megoldása is van: $x \equiv \pm 1,\ \pm 3 \pmod 8$.)

A következőkben a Miller-Rabin prímteszt algoritmusa kerül ismertetésre. A teszt páratlan számokra működik. Tegyük fel, hogy n páratlan, és írjuk fel

$$n - 1 = 2^k r,$$

ahol $k \in \mathbb{N}$ és r páratlan pozitív egész. A kis-Fermat tétel szerint, ha n prím, akkor minden $a \in \mathbb{Z}$ számra, ahol $a \not\equiv 0 \pmod n$ tudjuk, hogy

$$a^{n-1} \equiv 1 \pmod{n}.$$

Itt $n - 1 = 2^k r$, azaz

$$a^{2^k r} \equiv 1 \pmod{n}. \tag{9.12}$$

Ha n prím, akkor $x^2 \equiv 1 \pmod n$ esetén $x \equiv \pm 1 \pmod n$ mindig teljesül, ezért (9.12)-ból következik, hogy

$$a^{2^{k-1} r} \equiv \pm 1 \pmod{n}.$$

Abban az esetben, ha $a^{2^{k-1} r} \equiv 1 \pmod n$, akkor $a^{2^{k-2} r} \equiv \pm 1 \pmod n$, és így tovább...

A fentiek alapján az a maradékosztályt, ahol $a \not\equiv 0 \pmod n$ Miller-Rabin cinkosnak nevezzük, ha

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

vagy az

$$a^r, a^{2r}, a^{4r}, \ldots, a^{2^{k-1}r} \pmod{n}$$

sorozat tartalmazza a -1-es maradékosztályt modulo n. Ha n prím, akkor minden $a \not\equiv 0 \pmod{n}$ maradékosztály Miller-Rabin cinkos. Egy a modulo n maradékosztályt Miller-Rabin tanúnak nevezünk, ha $(a, n) = 1$ és a nem Miller-Rabin cinkos, azaz

$$a^r \not\equiv 1 \pmod{n},$$

továbbá az

$$a^r, a^{2r}, a^{4r}, \ldots, a^{2^{k-1}r} \pmod{n}$$

sorozat nem tartalmazza a -1 maradékosztályt modulo n. Ha n prímszám, akkor nincs Miller-Rabin tanú modulo n. Azaz, ha találunk egy Miller-Rabin tanút, akkor biztos, hogy n összetett szám. Primitív gyökök segítségével bebizonyítható, hogy **összetett n-re az összes modulo n redukált maradékosztálynak legalább a 75%-a Miller-Rabin tanú. Így ha k darab a esetén megnézzük, hogy az adott a Miller-Rabin cinkos-e, annak a valószínűsége, hogy mindegyik a Miller-Rabin cinkos és n összetett kisebb egyenlő mint 4^{-k}.** Mielőtt azonban rátérnénk a bizonyításra, megjegyezzük, hogy Keith [4] trükkös, elemi bizonyítást adott a fentinél egy kicsit gyengébb állításra.

Lássuk tehát annak a bizonyítását, hogy ha $n > 9$ páratlan összetett szám, akkor az összes modulo n redukált maradékosztálynak legalább a 75%-a Miller-Rabin tanú.

Ez a következő tételen alapul, melynek bizonyítását Crandall és Pomerence [2] könyve alapján adjuk meg.

9.9. Tétel. *Legyen $n > 9$ páratlan összetett szám. Írjuk fel $n-1$-et a következő alakban $n - 1 = 2^k r$, ahol $k > 1$ természetes szám és r páratlan. Legyen*

$$B = \{a \in \mathbb{Z}_n^* : \ a^r = 1 \text{ vagy } a^{2^i r} = -1 \text{ valamilyen } 0 \leq i < k\text{-ra}\}.$$

Ekkor

$$\frac{|B|}{\varphi(n)} \leq \frac{1}{4}.$$

A 9.9. Tétel bizonyítása. Jelölje 2^ℓ a legnagyobb kettőhatványt, amelyre igaz, hogy 2^ℓ osztója $p-1$-nek az n minden p prímosztójára. Ekkor a B halmazt tartalmazza a következő halmaz:

$$B' = \{a \in \mathbb{Z}_n^* : \ a^{2^{\ell-1}r} = \pm 1\}.$$

Valóban, világos, hogy ha $a^r = 1$, akkor $a \in B'$. Másrészt, ha $a^{2^i r} = -1$ valamilyen $0 \leq i < k$-ra, akkor $a^{2^i} r \equiv -1 \pmod{p}$ teljesül n minden p prímosztójára Ebből adódóan 2^{i+1} *pontos osztója* a rendjének modulo p, azaz $2^{i+1} \mid o_p(a)$, de $2^{i+2} \nmid o_p(a)$. Ekkor $2^{i+1} \mid p-1$ minden p prímosztójára n-nek. Ebből adódóan $\ell \geq i+1$. Azaz $a^{2^{\ell-1}r} = (-1)^{2^{\ell-i-1}}$, ami -1 vagy $+1$ lehet. Így $B' \subset B$ valóban.

A kínai maradéktétel szerint azon a-k száma, amelyre $a^{2^{\ell-1}r} = 1$ pontosan

$$\prod_{p \mid n} g(p),$$

ahol $g(p)$ az $x^{2^{\ell-1}r} \equiv 1 \pmod{p^{\alpha_p}}$ kongruencia megoldásszáma, ahol p^{α_p} a legnagyobb p hatvány, amely n-et osztja. Mivel $\mathbb{Z}_{p^{\alpha_p}}^*$ ciklikus csoport (azaz létezik mod p^{α_p} primitív gyök)

$$g(p) = ((p-1)p^{\alpha_p}, 2^{\ell-1}r) = (p-1, r)2^{\ell-1}. \tag{9.13}$$

Valóban, ha q egy fix primitív gyök modulo p^{α_p}, akkor $x \equiv q^u \pmod{p^{\alpha_p}}$ pontosan akkor megoldása az $x^{2^{\ell-1}r} \equiv 1 \pmod{p^{\alpha_p}}$ kongruenciának, ha $p^{\alpha_p - 1}(p-1) \mid u 2^{\ell-1} r$. Tudjuk, hogy $(r, p) = 1$ ($r \mid n-1$ és $p \mid n$ miatt), így ekkor $p^{\alpha_p - 1} \mid u$ és $\dfrac{p-1}{(p-1, 2^{\ell-1}r)} \mid u$ is teljesül. Tehát ekkor $\dfrac{p^{\alpha_p-1}(p-1)}{(p-1, 2^{\ell-1}r)} \mid u$. A $0 \leq u < p^{\alpha_p-1}(p-1)$ intervallumon $(p-1, 2^{\ell-1}r)$ darab ilyen u van, és ezzel (9.13)-t igazoltuk.

Így

$$\left|\{a : \mathbb{Z}_n^* : \ a^{2^{\ell-1}r} = 1\}\right| = \prod_p (p-1, r)2^{\ell-1}. \tag{9.14}$$

Hasonlóan belátható, hogy

$$\left|\{a : \mathbb{Z}_n^* : \ a^{2^{\ell}r} = 1\}\right| = \prod_p (p-1, r)2^{\ell},$$

ami pontosan kétszerese (9.14)-nek. Azaz

$$\left|\{a : \mathbb{Z}_n^* : \ a^{2^{\ell-1}r} = -1\}\right| = \prod_p (p-1, r)2^{\ell-1}.$$

Tehát

$$|\mathcal{B}'| = 2\prod_p (p-1, r)2^{\ell-1}.$$

Így:

$$\frac{|B'|}{\varphi(n)} = 2\prod_{p|n} \frac{(p-1, r)2^{\ell-1}}{(p-1)p^{\alpha_p-1}}.$$

Tegyük fel, hogy a tétel állításával ellentétben $\frac{|B|}{\varphi(n)} > \frac{1}{4}$. Mivel $B \subset B'$ ekkor

$$\frac{1}{4} < 2\prod_{p|n} \frac{(p-1, r)2^{\ell-1}}{(p-1)p^{\alpha_p-1}}. \tag{9.15}$$

Vegyük észre, hogy $(p-1, r)2^{\ell-1} \mid \frac{p-1}{2}$, így (9.15) jobboldala legfeljebb $2^{1-\omega(n)}$, ahol $\omega(n)$ az n szám különböző prímosztóinak számát jelöli. Ebből adódóan $\omega(n) \leq 2$.

Tegyük fel, hogy $\omega(n) = 2$. Ekkor n-nek két különböző prímosztója van, jelöljük őket p-vel és q-val. Ha közülük valamelyik négyzete is osztója n-nek, mondjuk $p^2 \mid n$, azaz $\alpha_p \geq 2$, akkor (9.15) jobboldala legalább $\frac{2^{1-2}}{p} \leq \frac{2^{1-2}}{3} = \frac{1}{6}$, ami ellentmondás. Tehát $\alpha_p = \alpha_q = 1$, azaz $n = pq$. Ekkor (9.15) a következő alakba írható:

$$\frac{p-1}{(p-1, r)2^{\ell}} \cdot \frac{q-1}{(q-1, r)2^{\ell}} < 2.$$

Mivel a baloldalon a szorzótényezők egészek, mindkettő szükségszerűen 1. Vagyis

$$p-1=(p-1,r)2^{\ell},\ q-1=(q-1,r)2^{\ell}.$$

Vagyis mind $p-1$-nek, mind $q-1$-nek 2^{ℓ} pontos osztója, illetve $p-1$ és $q-1$ legnagyobb páratlan osztója szükségszerűen r-nek is osztója. Jelölje $p-1$ legnagyobb páratlan osztóját s, $q-1$ legnagyobb páratlan osztóját t. Az előbbiek szerint $s \mid r$ és $t \mid r$. Mivel $p \equiv 1 \pmod{s}$ és

$$pq-1=n-1=2^k r,$$

így

$$q-1 \equiv pq-1 \equiv 2^k r \equiv 0 \pmod{s}.$$

Azaz $s \mid q-1$, de $q-1$ legnagyobb páratlan osztója t, így $s \mid t$. Hasonlóan belátható $t \mid s$. Azaz $t=s$. Vagyis $p-1=q-1$, és ebben az esetben is ellentmondásra jutottunk.

Végül, ha $n=p^{\alpha}$ alakú, akkor (9.15) alapján

$$\frac{1}{4}<\frac{(p-1,r)2^{\ell}}{(p-1)p^{\alpha}} \leq \frac{p-1}{(p-1)p^{\alpha}}=\frac{1}{p^{\alpha-1}},$$

amiből $p^{\alpha-1}<4$. Ez csak úgy lehet, ha $p=3$ és $\alpha=2$, ellentmondva a tétel $n>9$ feltételének. Ezzel minden esetben ellentmondásra jutottunk, és így a tétel állítását beláttuk.

Erich Bach [1] bebizonyította, hogy ha az általánosított Riemann hipotézis igaz, akkor a legkisebb Miller-Rabin tanú $\leq 2\,(\log n)^2$.

Az eddig felsorolt prímtesztek nagyon gyorsak, és egyszerűségük miatt máig a legkedveltebb prímtesztek közé tartoznak. Elméletig azonban vannak ennél gyorsabb prímtesztek is, így pl. [3]-ben Grantham egy olyan tesztet ad meg, amely sebessége aszimptotikusan körülbelül háromszorosa a Miller-Rabin tesztnek. Azonban ez a prímteszt az eddigieknél lényegesen komplikáltabb, így a lépések ismertetésére nem térünk ki. Az érdeklődő olvasók azonban [3]-ben utána nézhetnek.

Hivatkozások

[1] E. Bach, *Explicit bounds for primality testing and related problems*, Mathematics of Computation, 55 (191) (1990), 355–380.

[2] R. Crandall, C. Pomerance, *Prime Numbers; a Computational Perspective*, Springer Verlag, New York 2001.

[3] J. Grantham, *A probable prime test with high confidence*, J. Number Theory, 72 (1998), 32–47.

[4] C. Keith, *Miller-Rabin primality test*, Published in Encyclopedia of Cryptography 2011 Mathematics, https://kconrad.math.uconn.edu/blurbs/ugradnumthy/millerrabin.pdf.

[5] G. L. Miller, *Riemann's Hypothesis and Tests for Primality*, Journal of Computer and System Sciences, 13 (3) (1976), 300–317.

[6] M. O. Rabin, *Probabilistic algorithm for testing primality*, Journal of Number Theory, 12 (1) (1980), 128–138.

9.8. Egy példa a Miller-Rabin prímtesztre

Legyen $n = 561$ $(= 3 \cdot 11 \cdot 17)$. Ez az n Carmichael-szám (ld. 9.2. fejezet), így a Fermat prímteszt nem mondja meg nekünk, hogy n összetett. Teszteljük azonban az 561-et a Miller-Rabin teszttel. Ekkor

$$561 - 1 = 2^4 \cdot 35.$$

Válasszunk egy véletlen a-t. Mondjuk legyen $a = 7$. Tekintsük a

$$7^{35}, 7^{70}, 7^{140}, 7^{280}, 7^{560} \pmod{561}$$

sorozatot. Először az ismételt négyzetre emeléssel kiszámoljuk 7^{35}-t.

	7^1	7^2	7^4	7^8	7^{16}	7^{32}
mod 561	7	49	157	-35	103	-50

Ekkor

$$7^{35} \equiv 7^{32} \cdot 7^2 \cdot 7 \equiv (-50) \cdot 49 \cdot 7 \equiv 241 \pmod{561}.$$

Ezután ismételt négyzetre emeléssel kiszámítjuk a $7^{35}, 7^{70}, 7^{140}, 7^{280}, 7^{560}$ (mod 561) sorozat elemeit:

	7^{35}	7^{70}	7^{140}	7^{280}	7^{560}
mod 561	241	-263	166	67	1

Mivel a $7^{35}, 7^{70}, 7^{140}, 7^{280}$ (mod 561) sorozat nem tartalmazza a -1 modulo 561 maradékosztályt, így a 7 Miller-Rabin tanú, és az 561 összetett szám a teszt szerint.

9.9. AKS prímteszt

2002-ben három indiai matematikus, Manindra Agrawal, Neeraj Kayal és Nitin Saxena [1] megalkotott egy determinisztikus, polinomiális algoritmust prímtesztelésre. Ez az algoritmus túl bonyolult ahhoz, hogy azt teljes részletességében itt ismertessük, de azért pár szót ejtünk róla. Az algoritmus alapja a következő tétel:

9.10. Tétel. *Legyen $a \in \mathbb{Z}$, $n \in \mathbb{Z}$ és $(a, n) = 1$. Ekkor n akkor és csak akkor prím, ha*

$$(x + a)^n \equiv x^n + a \pmod{n}.$$

A 9.10. Tétel bizonyítása.

Először a következőt látjuk be: ha n prím és $1 \leq i \leq n-1$, akkor $n \mid \binom{n}{i}$. Valóban $\binom{n}{i} = \frac{n!}{i!(n-i)!}$ egész szám, így $i!(n-i)! \mid n!$. Azonban n prímszám és így nem szerepel $i!(n-i)!$ prímtényezős felbontásában, azaz $i!(n-i)! \mid (n-1)!$ is teljesül. Tehát $\frac{(n-1)!}{i!(n-i)!}$ egész szám, és ebből adódóan $\binom{n}{i} =$

$\frac{n!}{i!(n-i)!} = n \cdot \frac{(n-1)!}{i!(n-i)!}$ osztható n-nel. Ezután rátérhetünk a tétel bizonyítására. Legyen n prímszám.

Tudjuk x^i együtthatója $(x+a)^n - x^n - a$-ban $\binom{n}{i}a^{n-i}$ ha $1 \leq i \leq n-1$. Ekkor az együttható osztható n-nel az előzőek alapján. A konstans tag pedig $a^n - a$, ami a kis-Fermat tétel miatt osztható n-nel.

Tegyük fel, hogy n összetett.

Legyen a q prím olyan, hogy $q^k \mid n$ valamilyen k-ra, de $q^{k+1} \nmid n$. Ekkor $q^k \nmid \binom{n}{q} = \frac{n(n-1)(n-2)\cdots(n-q+1)}{q!}$ és $(q^k, a) = 1$ miatt x^q együtthatója nem osztható n-nel az $(x+a)^n - x^n - a$ polinomban. Ezzel a tétel állítását beláttuk.

Az AKS teszt kiindulópontja a fenti tétel, azonban maga az algoritmus kissé komplikált, és a bizonyítása komoly számelméleti eszközöket igényel. Mi most bizonyítás nélkül csupán az algoritmus lépéseit ismertetjük:

(1) $n = a^b$ valamely $a \in \mathbb{N}$ $b > 1$-ra, ha igen, akkor n összetett.
(2) Megkeressük a legkisebb r-et, amire n rendjére modulo r fennáll az $o_r(n) > \log_2^2 n$ összefüggés. (Ez az r sosem nagyobb mint $\lceil (\log_2 n)^5 \rceil$.)
(3) Minden $2 \leq a \leq \min(r, n-1)$, ellenőrizzük, hogy a osztója-e n-nek. Ha igen, akkor n összetett.
(4) Ha $n \leq r$, akkor n prímszám. (Ez a lépés elhagyható, ha $n \geq 5.7 \cdot 10^6$.)
(5) Amennyiben létezik a $0 \leq a \leq \sqrt{\varphi(r) \log_2 n}$, hogy

$$(x+a)^n \not\equiv x^n + a \pmod{x^r - 1, n},$$

akkor n összetett, egyébként prím.

Az utolsó lépésben $\pmod{x^r - 1, n}$ olyan ekvivalenciarelációhoz kapcsolódik, amely $\mathbb{Z}_n$ felett értendő és az x^r polinom ekvivalens a konstans 1 polinommal (vagyis n is és $x^r - 1$ is ekvivalens 0-val).

Az algoritmus futási ideje $O\big((\log n)^{12+\varepsilon}\big)$. 2005-ben Lenstra és Pomerance [4] kitalálta az AKS-nek egy olyan variánsát, amelynek futási ideje

$O((\log n)^6 (\log\log n)^c)$-re. A fenti cikknek van egy frissített változata is 2016-ban: [5].

Fontos kiemelni, hogy a legnagyobb prímek kutatásában számos magyar rekord született az elmúlt 30 évben. Például Csajbók T., Farkas G., Járai A., Járai Z. és Kasza J. [2] az ELTE IK-ról tartotta a világrekordot 2006-ban a legnagyobb ikerprím kutatásban. További rekordok elérhetőek a következő honlapon is: http://compalg.inf.elte.hu/~ajarai/worldr.htm. Az ún. Járai módszerről pedig az érdeklődők érdekes cikket olvashatnak pl. [3]-ben is. Ezek a módszerek nem az AKS-t alkalmazzák, hanem speciális alakú prímekre (ilyenek pl. Mersenne prímek vagy általánosított Fermat prímek) kifejlesztett determinisztikus prímteszteket.

Hivatkozások

[1] M. Agrawal, N. Kayal, N. Saxena, *PRIMES is in P*, Annals of Mathematics. 160 (2) (2004), 781–793.

[2] T. Csajbók, G. Farkas, A. Járai, Z. Járai, J. Kasza, *Report on the largest known Sophie Germain and twin primes*, Ann. Univ. Sci. Budap. Rolando Eötvös, Sect. Comput. 26 (2006), 181-183.

[3] G. Farkas, G. Gévay, P. Magyar, B. Szekeres *Járai's prime hunting methods reloaded (the largest known Cunningham chain of length 2 of the 2nd kind)*, Ann. Univ. Sci. Budap. Rolando Eötvös, Sect. Comput. 51 (2020), 69-75.

[4] H. W. Lenstra Jr., C. Pomerance, *Primality testing with Gaussian periods*, preliminary version July 20, 2005.

[5] H. W. Lenstra Jr., C. Pomerance, *Primality testing with Gaussian periods*, https://math.dartmouth.edu/~carlp/aks111216.pdf.

10. RSA

Ron Rivest, Adi Shamir és Len Adleman [1] az 1970-es évek közepe táján megalkotta az egyik legismertebb nyilvános kulcsú titkosítási eljárást, az RSA-t. (Az RSA elnevezés a szerzők nevének kezdőbetűiből ered...) Az RSA jó ideig számtalan informatikai, számítógépes, kommunikációs rendszerben jelentős szerepet játszott. Azonban ma már többen (bizonyos alkalmazások esetén) nem tartják elég biztonságosnak. Ennek oka az, hogy Peter Shor [2]-ben bebizonyította, hogy a faktorizáció és a diszkrét logaritmus kvantum algoritmussal polinom időben megoldható. Az elmúlt 30 évben sokat fejlődött a kvantum rezisztens kriptográfia, de hatékonyságban meg se tudják közelíteni az RSA-t és a diszkrét logaritmuson, illetve a diszkrét elliptikus logaritmuson alapuló módszereket.

Hivatkozások

[1] R. Rivest, A. Shamir, L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM. 21 (2) (1978), 120–126.

[2] P. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, Proceedings 35th Annual Symposium on Foundations of Computer Science (1994), 124 - 134.

10.1. RSA titkosítási algoritmus

Legyen $N = pq$ két nagy prím szorzata, ahol kettes számrendszerben minkét prím n darab számjegyből áll. Ezt az N-et nevezzük RSA modulusnak. Napjainkban N tipikus hosszúsága 4096 bit, amely 1234 decimális jegyet jelent. Eleinte az $n = 128$ bites modulus is biztonságosnak bizonyult, majd a támadások és technika fejlődésének hatására folyamatosan növekedett: 256,

512, 1024 majd 2048 bitre.

Legyen e, d két egész szám, ahol

$$ed \equiv 1 \pmod{\varphi(N)}.$$

Itt $N = pq$ miatt $\varphi(N) = (p-1)(q-1)$. Az e-t nyilvános, míg a d-t privát exponensnek nevezzük.

Az (N, e) páros a publikus kulcs, míg az (N, d) páros a privát vagy másképp titkos kulcs. Ez utóbbit csak az a személy ismeri, akinek a titkosított üzenetet küldjük, és aki dekódolja majd az üzeneteinket.

Az üzenetet tekinthetjük egy egész számnak, amelyre $0 < M < N$. (Az egyszerűség kedvéért most feltesszük, hogy $(M, N) = 1$ teljesül az üzenetre. Ez az üzenet esetleges kis módosításával könnyen elérhető. Valójában azonban nincs szükség az $(M, N) = 1$ feltételre, csak az általános esetben a dekódolás bizonyítása pár sorral hosszabb a lentieknél.)

A titkosított üzenet

$$C \equiv M^e \pmod{N} \qquad \leftarrow \text{RSA függvény.}$$

A dekódolás

$$ed \equiv 1 \pmod{\varphi(N)} \qquad \text{miatt}$$

$\exists k \in \mathbb{Z}^+$, melyre $ed = k\varphi(N) + 1$. Az Euler-Fermat tétel alapján

$$M \equiv C^d \pmod{N} \qquad \text{ez a dekódolás,}$$

ugyanis

$$C^d \equiv (M^e)^d = M^{k\varphi(N)+1} = M\left(M^{\varphi(N)}\right)^k \equiv M \cdot 1^k \equiv M \pmod{N}.$$

Az RSA egy egyirányú függvény, mivel a titkosítás a publikus kulcs (N, e) ismeretében könnyen elvégezhető moduláris hatványozással, melynek időigénye $O(\log e(\log N)^2)$ bitoperáció, azonban az invertálás d (azaz a privát) kulcs ismerete nélkül nagyon nehéz. Az RSA támadások nagyobb része arra irányul, hogy d ismeret nélkül, hogyan lehetne invertálni az RSA függvényt.

Pontosabban fogalmazva, ha csupán (N, e, C) hármas adott (és p, q, d titkos) akkor képesek vagyunk-e az eredeti M üzenetet C-ből visszaállítani.

Megjegyzés: p, q, e-ből d számolható, ekkor ugyanis d az

$$ex \equiv 1 \pmod{(p-1)(q-1)}$$

lineáris kongruencia megoldása. Azonban p, q titkos, csak $N = pq$ adott, p, q ismeretéhez N-et faktorizálni kell, ez azonban nagy N számok esetén nagyon lassú.

Van-e más az N faktorizációját megkerülő invertálás?

Hivatkozások

[1] R. Rivest, A. Shamir, L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM. 21 (2) (1978), 120–126.

10.2. RSA támadások

Az alábbiakban Dénes Tamás [3] ismeretterjesztő cikkére alapozva ismertetünk néhány ismert RSA támadást, amelyre, az RSA használatakor mindenképpen figyelni kell. A téma iránt érdeklődőknek ajánlom még a [4] cikk olvasását is.

Implementáció függő támadások

A számítógép szerverének vizsgálatával, pl. a szerver azon területének behatárolása, amelyben a kulcsok, akár kódolt formában tárolásra kerülnek.

Az áram felvételének időbeni mérése.

Ezek a támadások nem csupán elméleti megfontolásokat, hanem komoly technikát is igényelnek.

Matematikai szempontból izgalmasabb a következő "időbeni támadásoknak"-nak nevezett támadás.

Időbeni támadások

Ismert, ld. pl. Lovász-Gács könyve [5] hogy egy n jegyű szám négyzetre emelésének a bonyolultsága kisebb, mint két n jegyű szám összeszorzása. Ez nem csak elméleti eredmény, de praktikusan is így van. A moduláris hatványozásban pontosan ilyen műveletek szerepelnek. Készültek olyan implementációk, amelyek kihasználták ezt a különbséget és jelentős gyorsítást értek el.

Válaszul az elektromérnökök felhívták a figyelmet arra, hogy ha valaki a processzort meg tudja figyelni, akkor az időkülönbség miatt meg tudja állapítani, hogy mikor végzett az négyzetre emelést és mikor szorzást, amelyből a titkos dekódoló kitevő bináris alakja könnyen felírható. Ezek után már nem használták a fenti gyorsítási lehetőséget.

Helytelen alkalmazáson alapuló támadások

Gyakran csak p-t választják véletlen prímnek, q-t pedig a p-t követő legkisebb prímnek. Ekkor

$$N = p(p+2) \qquad \text{pl.},$$

amiből p nagyon gyorsan számolható. Ha p, q egymást követő prímek és $q - p = d$ viszonylag kicsi,

$$N = p(p+d)\text{-ből } p \text{ könnyen számolható} \ldots$$

Számtalan hasonló eset létezik, amit ugyan nem tilt meg az RSA algoritmus, azonban ezekben az esetekben feltörhető az RSA.

Az RSA körültekintő használata fontos!

Közös modulus

Fontos, hogy egy rögzített N modulust egyetlen ember használjon csak, ugyanis az azonos modulussal rendelkező emberek tudják fejteni egymás üzeneteit Simmons alábbi tétele alapján. (Az alábbi tétel teljes leírása J. DeLaurentis publikációjában [2] jelent meg először, aki a dolgozat elején említette, hogy a tétel Simmonstól származik.)

10.1. Tétel. (Simmons) *Legyen (N, e) egy RSA publikus kulcs. Ha adott a privát d exponens, akkor N faktorizációja hatékonyan elvégezhető. Fordítva is igaz ez: Ha N faktorizációja ismert, akkor minden e publikus exponenshez hatékonyan számolható a privát exponens d.*

Néhány számítógépes rendszer (közösség) nem törődik ezzel a tétellel, és minden felhasználónak ugyanazt az N modulust adják, bár mindenkinek különböző privát és publikus exponenst adnak... Emögött gyakran spórolási okok vannak, hiszen nagy p és q prímek generálása sokba kerül, és ha minden felhasználónak más és más prímpár kell, az bizony jócskán megsokszorozza az árakat... Azonban, bizony ezzel nem érdemes spórolni, ha fontosnak tartjuk az informatikai biztonságot! (Itt megjegyezzük azért, hogy ha nem kötjük ki, hogy p és q **biztosan** prím legyen, hanem csak nagyon-nagy valószínűséggel, az lényegesen felgyorsítja p és q generálási idejét. A gyakorlatban bőségesen elegendő prímszámnak egy olyan szám, amelyik átmegy mondjuk 511 Miller-Rabin prímteszten. Annak a valószínűsége ugyanis, hogy egy ilyen szám összetett kisebb, mint $4^{-511} = 2^{-1022}$, ami a gyakorlatban elhanyagolható valószínűség. Az 1024 bites páratlan számok száma 2^{1023}, ezért várhatóan legfeljebb két "csaló" szám van. Egyébként 511 Miller-Rabin prímtesztet lefuttatni sem tart sokáig, de a gyakorlatban már 100 teszt is bőven elegendő. A fentieknél a determinisztikus prímtesztek sokkal lassabbak.)

A 10.1. Tétel bizonyítása.

p, q, e adott $\Rightarrow$ d könnyen számolható

Ekkor d az

$$ex \equiv 1 \pmod{\varphi(N)}$$
$$ex \equiv 1 \pmod{(p-1)(q-1)}$$

lineáris kongruencia megoldása.

$\underline{N, e, d \text{ adott} \Rightarrow p, q \text{ könnyen számolható}}$

Legyen $k \stackrel{\text{def}}{=} ed - 1$.

$$ed \equiv 1 \pmod{\varphi(N)}$$
$$\varphi(N) \mid ed - 1$$
$$\varphi(N) \mid k.$$

$\varphi(N)$ páros, hiszen $\varphi(N) = (p-1)(q-1)$. Azaz k is páros: $k = 2^t r$, ahol r páratlan. Ekkor

$$\forall\ (g, N) = 1\text{-re} \qquad g^k \equiv 1 \pmod{N}$$

mivel $\varphi(N) \mid k$. Ebből adódóan $g^{k/2}$ második egységgyök modulo N:

$$\left(g^{k/2}\right)^2 = g^k \equiv 1 \pmod{N}.$$

A kínai maradéktétel szerint 4 darab egységgyök létezik modulo pq (most $N = pq$). Ezek közül két darab ± 1. A másik két egységgyök $\pm x$, ahol

$$x \equiv 1 \pmod{p} \qquad x \equiv -1 \pmod{q}.$$

Vagyis $p \mid x - 1$ és $p \mid N$ miatt

$$p \mid (x-1, N) \Rightarrow p = (x-1, N). \tag{10.1}$$

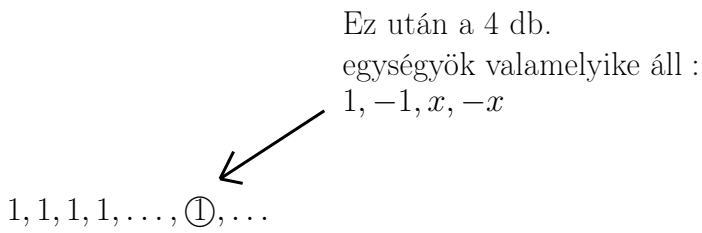
(Itt azt használtuk, hogy N-nek csak 4 darab osztója van: $1, p, q, pq$ így $(x-1, N)$ a fenti négy érték közül lehet olyan, ami p-vel osztható, vagyis

p és pq. Azonban $x-1$ ekkor q-val már nem osztható, tehát valóban $p = (x-1, N)$.)

Nézzük a

$$g^k, g^{k/2}, g^{k/4}, \ldots, g^{k/2^t} \qquad (\text{mod } N)$$

sorozatot. Ez néhány egyessel kezdődik:



Vagyis legalább $1/2$ eséllyel a sorozatnak az eleje

$$1, 1, 1, 1, \ldots, 1, \pm x, \ldots$$

alakú. Ebben az esetben valamilyen s-re $g^{k/2^s} \equiv \pm x \pmod N$, így g-re alkalmazott moduláris hatványozással gyorsan meghatározható x, és az euklideészi algoritmust használva, ha kiszámoljuk $x-1$ és N legnagyobb közös osztóját, akkor megkapjuk a p prímosztót (ld. (10.1)).

Kis privát exponens

Az RSA gyakorlati alkalmazásai során előfordulhat, hogy csökkentsék a kulcsok generálási idejét, vagy hogy gyorsabb legyen a dekódolás, véletlenül választott d privát exponens helyett egy kis d-t választanak. Azonban M. Wiener [6] által bizonyított tétel mutatja, hogy ha d egy megadott értéknél kisebbé válik, akkor az RSA megfejthető lesz...

Legyen N prímtényezős felbontása pq, ahol most $q < p$. Mivel p és q kettes számrendszerben ugyanannyi számjegyből áll, azt is tudjuk, hogy $p < 2q$. Wiener a következőt bizonyította:

10.2. Tétel. (Wiener) *Legyen $N = pq$ úgy hogy $q < p < 2q$ és legyen*

$$d < \frac{1}{3}N^{1/4}. \tag{10.2}$$

Ha adott (N, e) (azaz az RSA modulus és a publikus exponens) és d értékét ugyan nem ismerjük, de az tudjuk, hogy (10.2) teljesül rá, akkor a privát exponens d hatékonyan kiszámolható N és e értékéből (feltéve hogy d megoldása az $ed \equiv 1 \pmod{\varphi(N)}$ kongruenciának).

A 10.2. Tétel bizonyítása. Tudjuk

$$ed \equiv 1 \pmod{\varphi(N)}.$$

Legyen

$$ed = t\varphi(N) + 1 \qquad \Rightarrow ed - t\varphi(N) = 1.$$

Ekkor:

$$\left|\frac{e}{\varphi(N)} - \frac{t}{d}\right| = \frac{1}{d\varphi(N)}.$$

Mivel $\frac{e}{\varphi(N)}$ közel van $\frac{e}{N}$-hez, ezért $\frac{t}{d}$ egy approximációja $\frac{e}{N}$-nek, és ezért a (3.24. tétel következményeképpen) d a lánctört alakból megkapható. Precízen:

$$\begin{aligned}
N = pq \quad & q < p < 2q \\
q^2 < pq = N \quad & \Rightarrow q < \sqrt{N} \\
\frac{p^2}{2} < pq = N \quad & \Rightarrow p < \sqrt{2}\sqrt{N} \\
p + q < 3\sqrt{N} & \\
N - \varphi(N) = pq - (p-1)(q-1) = p + q - 1 < 3\sqrt{N} & \\
|N - \varphi(N)| < 3\sqrt{N}. &
\end{aligned}$$

Így

$$\left|\frac{e}{N} - \frac{t}{d}\right| = \left|\frac{ed - t\varphi(N) - tN + t\varphi(N)}{Nd}\right|$$

$$= \left| \frac{1 - t\left(N - \varphi(N)\right)}{Nd} \right| \leq \frac{3t\sqrt{N}}{Nd} = \frac{3t}{d\sqrt{N}}.$$

Itt

$$t\varphi(N) = ed - 1 < ed.$$

Mivel $e < \varphi(N)$

$$t\varphi(N) < \varphi(N)d$$
$$t < d < \frac{1}{3}N^{1/4}.$$

Azaz

$$\left| \frac{e}{N} - \frac{t}{d} \right| \leq \frac{3t}{d\sqrt{N}} < \frac{N^{1/4}}{d\sqrt{N}} = \frac{1}{dN^{1/4}} < \frac{1}{2d^2}.$$

A lánctörtek elmélete alapján, lásd 3.24. Tétel, az összes ilyen $\frac{t}{d}$ tört megkapható $\frac{e}{N}$ lánctört alakjából. Mivel

$$ed - t\varphi(N) = 1 \qquad \Rightarrow \qquad (t, d) = 1.$$

Azaz ha $r = \frac{t}{d}$ értéke adott, akkor mivel tudjuk, hogy $(t, d) = 1$, rögtön megkapjuk t-t és d-t is.

A modulus faktorizációján alapuló támadások

10.3. Tétel. *Ha adott egy $N = pq$ alakú szám, mely két különböző prím szorzata, akkor p, q kiszámítása lényegében polinomiálisan ekvivalens $\varphi(N)$-ével. Pontosabban:*

N, p, q-t ismerve, $T(\varphi(N)) = O(\log N)$

$N, \varphi(N)$-t ismerve $T(p, q) = O((\log N)^3)$.

A 10.3. Tétel bizonyítása.

a) $\varphi(N) = (p-1)(q-1) = pq-p-q+1 = N-p-q+1$ értéke kiszámolható 2 db kivonással és 1 összeadással, így az időigény $O(\log N)$.

b) Mivel $\varphi(N) = N - p - q + 1$, így

$$p + q = N - \varphi(N) + 1$$
$$pq = N$$

A gyökök és együtthatók közötti összefüggésből kapjuk, hogy p és q gyöke az

$$x^2 - (N - \varphi(N) + 1)x + N = 0$$

másodfokú egyenletnek. A megoldóképletet használva látjuk

$$p, q = \frac{N - \varphi(N) + 1 \pm \sqrt{(N - \varphi(N) + 1)^2 - 4N}}{2},$$

amely $O\left((\log N)^3\right)$ bitoperációt igényel.

Az elméleti RSA támadások legismertebb csoportja az, amelyben a támadás a modulus faktorizálására irányul, hiszen N prímtényezős felbontásából $\varphi(N)$ könnyen számolható, s ebből $d \equiv e^{-1} \pmod{\varphi(N)}$ is.

A faktorizációs algoritmusok lassúak, ezekről a következő fejezetben bővebben lesz szó.

Hivatkozások

[1] D. Boneh, *Twenty years of attacks on the RSA cryptosystem*, Notices of the American Mathematical Society. 46 (2) (1999), 203–213, http://crypto.stanford.edu/~dabo/abstracts/RSAattack-survey.html.

[2] J. DeLaurentis, *A further weakness in the common modulus protocol for the RSA cryptoalgorithm*, Cryptologia 8 (1984), 253–259.

[3] Dénes T., *Új eredmények az RSA kulcsok megfejtéséhez*, Híradástechnika, 2002/1. 47-55, http://www.titoktan.hu/_raktar/_e_vilagi_gondolatok/HTRSA.htm.

[4] Dénes T., *A Public Key System és az RSA biztonsági kérdései*, Híradástechnika, 2002/1. 47-55, http://www.titoktan.hu/_raktar/_e_vilagi_gondolatok/PKS-RSA.htm.

[5] Gács P., Lovász L., *Algoritmusok*, Tankönyvkiadó 1989.

[6] M. J. Wiener, *Cryptanalysis of short RSA secret exponents*, IEEE Transactions on Information Theory. 36 (3) (1990), 553–558.

11. Faktorizáció

Valamennyi eddig tárgyalt prímteszt (kivéve a „próbaosztást”, mely nagyon lassú), csak azt mutatja ki, hogy n összetett, de faktort általában nem ad, pl. az RSA-ban [13] is használt $n = pq$, $p, q > \frac{\sqrt{n}}{2}$ típusú számok faktorizálására nem használhatóak.

Sőt, láttuk, hogy Miller-Rabin prímteszttel [8], [12] annak eldöntése, hogy egy adott szám prímszám-e valószínűen polinomiális időben eldönthető. Ennél kicsit lassabb, de minden esetben polinomiális idejű prímteszt is létezik az AKS prímteszt [1]. Faktorizációra nem ismerünk polinomiális idejű algoritmust (talán nincs is ilyen).

Sok faktorizációs módszer ismert, jelen fejezetben azok közül ismertetek párat, amely Koeblitz [4] könyvében is szerepelnek. Ezek: Fermat faktorizáció [3], [7], faktor bázis algoritmus (amelyet angolul „Dixon’s random square method”-nak hívnak) [2], lánctört módszer [9]. A fentieken kívül számos más faktorizációs algoritmus is ismert, így például, Pollard ρ módszere [10] (ld. 13.5. fejezet), a kvadratikus szita [11], számtest szita [6] és elliptikus görbéken alapuló faktorizáció [5] (melyről a 13.5. fejezetben lesz szó).

Jelenlegi ismeretünk szerint 100 decimális számjegyig a kvadratikus szita a leggyorsabb, 100 decimális számjegy felett pedig a számtest szita.

Hivatkozások

[1] M. Agrawal, N. Kayal, N. Saxena, *PRIMES is in P*, Annals of Mathematics. 160 (2) (2004), 781–793.

[2] J. D. Dixon, *Asymptotically fast factorization of integers*, Math. Comp. 36 (153) (1981), 255–260.

[3] P. Fermat, *Oeuvres de Fermat*, vol. 2, 1894, o. 256.

[4] N. Koeblitz, *A Course in Number Theory and Cryptography*, Springer, 1994.

[5] H. W. Lenstra Jr., *Factoring integers with elliptic curves*, Annals of Mathematics. 126 (3) (1987), 649–673.

[6] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, J. M. Pollard, *The number field sieve*, kiterjesztett absztrakt: Proc. 22nd Annual ACM Sympos. Theory of Computing (STOC) (Baltimore, May 14-16, 1990), 564-572.

[7] J. McKee, *Speeding Fermat's factoring method*, Mathematics of Computation (68) (1999), 1729–1737.

[8] G. L. Miller, *Riemann's hypothesis and tests for primality*, Journal of Computer and System Sciences, 13 (3) (1976), 300–317.

[9] M. A. Morrison, J. Brillhart, *A method of factoring and the factorization of F7*. Mathematics of Computation. American Mathematical Society. 29 (129) (1975), 183–205.

[10] J. M. Pollard, *A Monte Carlo method for factorization*. BIT Numerical Mathematics. 15 (3) (1975), 331–334.

[11] C. Pomerance, *Analysis and comparison of some integer factoring algorithms*, in Computational Methods in Number Theory, Part I, H.W. Lenstra, Jr. and R. Tijdeman, eds., Math. Centre Tract 154, Amsterdam, 1982, pp 89-139.

[12] M. O. Rabin, *Probabilistic algorithm for testing primality*, Journal of Number Theory, 12 (1) (1980), 128–138.

[13] R. Rivest, A. Shamir, L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM. 21 (2) (1978), 120–126.

11.1. Fermat-faktorizáció

Tegyük fel, hogy egy olyan n páratlan számot akarunk faktorizálni, mely két közel egyforma nagy szám szorzata.

$$ab = n, \quad a > b, \quad a - b \text{ kicsi } a\text{-hoz képest.}$$

Első lépésben, nem n teljes prímtényezős felbontását szeretnénk megkapni, hanem csak n-nek egy valódi osztóját. (Ha n-et felírtuk ab alakban, ahol $1 < a, b < n$, akkor a-ra és b-re folytatva az algoritmust, előbb-utóbb megkapjuk n prímtényezős felbontását is.)

A következőkben azt mutatjuk meg, hogy ahelyett, hogy n-et faktorizálni szeretnénk, azaz olyan $a, b \in \mathbb{Z}$-t keresnénk, amelyre

$$n = ab \qquad \text{és} \quad 1 < a, b < n,$$

inkább olyan $x, y \in \mathbb{Z}$-t keresünk, amelyre

$$n = x^2 - y^2.$$

A továbbiakban feltesszük, hogy n páratlan. Az alábbiak miatt a fenti két probléma ekvivalens. Először legyen $n = ab$. Keressünk olyan x, y-t, hogy

$$\begin{aligned} x + y &= a, \\ x - y &= b. \end{aligned} \tag{11.1}$$

Ekkor:

$$x = \frac{a+b}{2}, \quad y = \frac{a-b}{2}$$

$$x^2 - y^2 = \left(\frac{a+b}{2}\right)^2 - \left(\frac{a-b}{2}\right)^2 = \frac{2ab}{4} + \frac{2ab}{4} = ab = n.$$

Megfordítva, ha $x^2 - y^2 = n$, akkor a-t, b-t (11.1)-gyel definiálva

$$x^2 - y^2 = (\underbrace{x+y}_{a})(\underbrace{x-y}_{b}) = ab = n.$$

Ezek után

11.1. Definíció. (Fermat faktorizációs algoritmus [1]) *Tegyük fel, hogy n nem négyzetszám. Vesszük az első n-nél nagyobb négyzetszámot, ez t^2, ahol $t = [\sqrt{n}] + 1$. Először legyen $x = t = [\sqrt{n}] + 1$. Ha*

$$x^2 - n = t^2 - n = \text{ négyzetszám } = y^2,$$
$$x^2 - y^2 = n.$$

Ha $t^2 - n \neq$ nem négyzetszám, tekintjük a következő négyzetszámot, azaz most $x = t + 1$:

$$x^2 - n = (t+1)^2 - n = \text{ négyzetszám } = y^2,$$
$$x^2 - y^2 = n.$$

Ha $(t+1)^2 - n$ nem négyzetszám, tekintjük $x = t + 2$-t és így tovább. Amennyiben találunk x-et ($\neq \frac{n+1}{2}$) és y-t ($\neq \frac{n-1}{2}$)-t, amelyre

$$n = x^2 - y^2,$$

akkor az $n = (x-y)(x+y)$ képlettel megkaptuk n egy nem triviális szorzattá bontását.

Hivatkozások

[1] P. Fermat, *Oeuvres de Fermat*, vol. 2, 1894, o. 256.

[2] N. Koeblitz, *A Course in Number Theory and Cryptography*, Springer, 1994.

11.2. Faktorbázis algoritmus

A Fermat faktorizáció [2] egy ügyes ötlettel tovább fejleszthető sokkal hatékonyabb algoritmussá.

Tegyük fel, hogy az n páratlan számot akarjuk faktorizálni. A Fermat faktorizáció [2] során láttuk, elegendő olyan x, y egész számokat keresni, amelyekre $x^2 - y^2 = n$. Ehelyett azonban elég olyan x, y egész számokat keresni,

amelyekre:

$$x^2 - y^2 \equiv 0 \pmod{n},$$
$$y \not\equiv \pm x \pmod{n}. \tag{11.2}$$

Ugyanis ekkor

$$n \mid x^2 - y^2 = (x-y)(x+y),$$
$$n \nmid x - y, \quad n \nmid x + y,$$
$$1 < (n, x-y), (n, x+y) < n.$$

Az Eukleidészi algoritmussal pl. $(n, x+y)$-t kiszámítva, megkapjuk n egy valódi osztóját, vagyis egy faktorát. A kapott fakorral n-et leosztva az eljárás folytatható, mindaddig, amíg megkapjuk n prímtényezős felbontását.

A (11.2)-et kielégítő (x, y) pár keresése az ún. faktorbázis algoritmussal történik ld. pl. [4], [1]. Az algoritmust D. H. Lehmer és R. E. Powers alkotta meg [4]-ben.

11.2. Definíció. *A $\{p_1, p_2, \ldots, p_h\}$, halmazt, ahol $p_1 < p_2 < \ldots < p_h$ halmazt* faktorbázisnak *nevezzük, ha $p_1 = -1$ és $p_2, \ldots, p_h$ különböző prímek. Továbbá a faktorizálandó $n \in \mathbb{N}$-et rögzítve $a \in \mathbb{N}$ esetén azt mondjuk, hogy a egy B-szám, ha $a^2 \mod n$ abszolút legkisebb maradéka B-beli számok szorzataként írható.*

Példa. Ha $n = 4633$ és $B = \{-1, 2, 3\}$, akkor 67, 68, 69 B-számok, hiszen

$$67^2 \equiv -144 \equiv (-1) \cdot 2^4 \cdot 3^2 \qquad (4633),$$
$$68^2 \equiv -9 \equiv (-1) \cdot 3^2 \qquad (4633),$$
$$69^2 \equiv 128 \equiv 2^7 \qquad (4633).$$

11.3. Definíció. (Faktorbázis algoritmus [4], [1]) *Legyen n, $B = \{p_1, p_2, \ldots, p_h\}$ adott. Keressünk $h+1$ db B számot, mondjuk $a_1, \ldots, a_{h+1}$-et.*

$$a_i{}^2 \equiv b_1^{\alpha_{i,1}} \cdots b_h^{\alpha_{i,h}} \pmod{n} \quad i = 1, \ldots, h+1\text{-re.}$$

Legyen $\alpha_{i,j} \mod 2$ *maradéka a* $\beta_{i,j}$*, tehát* $\alpha_{i,j} \equiv \beta_{i,j} \pmod 2$ *és* $0 \leq \beta_{i,j} \leq 1$*. Tekintsük a* $h+1$ *darab*

$$\underline{u_i} = (\beta_{i,1}, \ldots, \beta_{i,h}) \quad (i = 1, \ldots, h+1)$$

vektort az $\mathbb{F}_2$ *feletti* h*-dimenziós vektortérből. Tudjuk, hogy* $h+1$ *darab vektor* h*-dimenziós vektortérben lineárisan összefüggő, azaz* $\exists\, \varepsilon_1, \ldots, \varepsilon_{h+1} \in \mathbb{F}_2$ *elemek, hogy*

$$\varepsilon_1 \underline{u_1} + \ldots + \varepsilon_{h+1} \underline{u_{h+1}} = \underline{0}. \tag{11.3}$$

Ekkor:

$$\prod_{i=1}^{h+1} a_i^{2\varepsilon_i} = \prod_{i=1}^{h+1} p_1^{\varepsilon_i \alpha_{i,1}} \ldots p_h^{\varepsilon_i \alpha_{i,h}} \equiv \prod_{j=1}^{h} p_j^{\sum_{i=1}^{h+1} \varepsilon_i \alpha_{i,j}}.$$

Itt $\sum\limits_{i=1}^{h+1} \varepsilon_i \alpha_{i,j}$ *minden* $1 \leq j \leq h$ *esetén páros* (11.3) *miatt. Így legyen* $\sum_{i=1}^{h+1} \varepsilon_i \alpha_{i,j} \stackrel{\text{def}}{=} 2k_j$*. Ekkor:*

$$\Big(\underbrace{\prod_{i=1}^{h+1} {a_i}^{\varepsilon_i}}_{x} \Big)^2 \equiv \Big(\underbrace{\prod_{j=1}^{h} {b_j}^{k_j}}_{y} \Big)^2 \pmod n.$$

Ha itt $x \not\equiv \pm y\ (n)$*, kész vagyunk, találtunk* (11.2)*-nek megfelelő* x, y *párt, és ezzel sikerült* n*-et faktorizálnunk.*

Ha $x \equiv -y$ *vagy* $x \equiv +y \pmod n$*, új* ${a_1}', \ldots, {a_{h+1}}'$ B*-számokat keresünk és így tovább, egészen addig, amíg az algoritmussal olyan* x*,* y*-hoz jutunk, amire* (11.2) *teljesül, és amivel megtaláljuk* n*-nek egy nem triviális szorzattá bontását.*

Az általános faktor bázis algoritmusban [1] a B-számokat véletlen módszerrel keressük. Azaz véletlenszerűen kiválasztunk egy a egész számot, ellenőrizzük, hogy B-szám-e. Ha B-szám, akkor megtartjuk, ha úgy alakult, hogy nem B-szám, új jelöltet választunk, és így tovább, amíg nem találunk B-számot.

Példa. Előbbi példában

$$67^2 \to (1,4,2) \to (1,0,0) = \underline{u_1},$$
$$68^2 \to (1,0,2) \to (1,0,0) = \underline{u_2},$$
$$69^2 \to (0,7,0) \to (0,1,0) = \underline{u_3}.$$

Még egy kellene a biztos lineáris összefüggéshez, de itt speciel már az első két vektor is lineárisan összefüggő, hiszen

$$1\underline{u_1} + 1\underline{u_2} + 0 \cdot \underline{u_3} = (2,0,0) \to (0,0,0).$$

Ez alapján:

$$67^2 \cdot 68^2 \equiv (-1)^1 2^4 3^2 \cdot (-1)^1 3^2 = (-1)^2 2^4 3^4 \equiv (2^2 \cdot 3^2)^2 \equiv 36^2 \pmod{4633}.$$

Itt $67 \cdot 68 \equiv -77 \pmod{4633}$, így

$$77^2 \equiv 36^2 \pmod{4633}.$$

Azaz

$$4633 \mid \underbrace{(77-36)}_{41}\underbrace{(77+36)}_{113}.$$

Kiszámolva

$$(41, 4633) = 41\text{-t},$$

megkapjuk 4633 egy valódi osztóját 41-et, s valóban 4633 prímtényezős felbontása:

$$4633 = 41 \cdot 113.$$

A faktorbázis algoritmus [1] esetén tipikusan

$$B = \{-1\} \cup \{p : \ p \text{ prím }, \ p \le y\},$$

ahol y-t n függvényében úgy érdemes választani, hogy az algoritmus futási ideje optimális legyen. Ügyesen megválasztott y mellett az algoritmus futási ideje $\exp(c\sqrt{\log n \log\log n})$. A futási idő heurisztikus becslésének rövidített leírása megtalálható pl. Koeblitz könyvében [3] is.

Hivatkozások

[1] J. D. Dixon, *Asymptotically fast factorization of integers*, Math. Comp. 36 (153) (1981), 255–260.

[2] P. Fermat, *Oeuvres de Fermat*, vol. 2, 1894, o. 256.

[3] N. Koeblitz, *A Course in Number Theory and Cryptography*, Springer, 1994.

[4] D. H. Lehmer, R. E. Powers, *On factoring large numbers*, Bull. Amer. Math. Soc. 37 (10) (1991), 770-776.

11.3. Lánctört algoritmus

A faktorbázis módszernél [1] n-et szerettük volna faktorizálni, ehhez vettünk egy

$$B = \{-1\} \cup \{p : \ p \leq y\}$$

halmazt és B-sima számokat kerestünk.

Pontosabban: Vettünk egy $a \in \mathbb{Z}$ számot. Néztük $a^2 \bmod n$ abszolút legkisebb maradékát ($-\frac{n}{2}$ és $\frac{n}{2}$ közé esik), és néztük, B sima-e. Ezt a módszert Morrison és Brillhart [3] fejlesztette tovább lánctörtek segítségével.

A lánctört algoritmus [3]: Ebben az algoritmusban a-t úgy választjuk, hogy az $\sqrt{n}$ lánctört közelítő $\frac{p_i}{q_i}$ törtjeiből $a = p_i$. Ha $r_n(p_i^2)$ jelöli p_i^2 modulo n legkisebb abszolút maradékát, azaz

$$r_n(p_i^2) \equiv p_i^2 \pmod{n}, \text{ ahol } -\frac{n}{2} \leq r_n(p_i^2) \leq \frac{n}{2},$$

akkor hamarosan többet is látni fogunk, nevezetesen, hogy

$$\left|r_n({p_i}^2)\right| \leq 2\sqrt{n} \tag{11.4}$$

is teljesül. Azaz p_i^2 nagyobb eséllyel B-sima, mint egy véletlenül választott $a \in \mathbb{Z}_n$-re, hiszen lényegesen (gyökhatvánnyal) kisebb a várható cn körüli

értéknél. Mindehhez elég (11.4)-t belátni, amihez egyszerűen alkalmazzuk a 3.23. Tételt $x = \sqrt{n}$-re:

$$\left|{p_i}^2 - n{q_i}^2\right| < 2\sqrt{n},$$

azaz

$$\left|r_n({p_i}^2)\right| < 2\sqrt{n}.$$

Mivel $\sqrt{n}$ közelítő törtjeiben a p_i-kre fennáll

$$p_i = a_i p_{i-1} + p_{i-2}$$

rekurzió (lásd 3.22. Tétel), és ez modulo n is igaz, azaz

$$p_i \equiv a_i p_{i-1} + p_{i-2} \pmod{n},$$

így p_i kiszámításához csak a $\sqrt{n}$-nek az a_i lánctört jegyeire van szükség. Mivel $\sqrt{n}$ egy másodfokú egyenletnek a gyöke, ezért, a 3.20. Tétel alapján az $a_0, a_1, a_2, \ldots$ lánctört számjegyek sorozata előbb-utóbb periodikus lesz.

A szerzők javasolták még, hogy szükség esetén ismételjük meg az eljárást $\sqrt{kn}$-nel, ahol k kicsi természetes szám. Lehet akár $k = 2$ is. Pl. Morrison és Billhart cikkében [3] a 7. Fermat szám, $2^{2^7} + 1$ faktorizálása során $k = 257$ bizonyult a megfelelő választásnak.

A futási idő most is $\exp(c\sqrt{\log n \log\log n})$, de egy jobb c konstanssal mint az általános faktorbázis algoritmus esetében.

Hivatkozások

[1] J. D. Dixon, *Asymptotically fast factorization of integers*, Math. Comp. 36 (153) (1981), 255–260.

[2] N. Koeblitz, *A Course in Number Theory and Cryptography*, Springer, 1994.

[3] M. A. Morrison, J. Brillhart, *A method of factoring and the factorization of F7*. Mathematics of Computation. American Mathematical Society. 29 (129) (1975), 183–205.

11.4. A kvadratikus szita módszer

A kvadratikus szita módszer [2] a faktorbázis algoritmus egy ügyes variánsa, amely Pomerancetól származik. Ekkor a faktorbázisba olyan p prímeket veszünk be egy adott határig, amelyekre n kvadratikus maradék modulo p, pontosabban:

$$B = \{-1, 2\} \cup \{p : \ p \text{ prím} \ \ p \leq y, \ \left(\frac{n}{p}\right) = 1\}.$$

A B-sima számokat pedig egy

$$S = \{t^2 - n : \ [\sqrt{n}] + 1 \leq t \leq [\sqrt{n}] + A\}$$

halmazban keressük, ahol A (n függvényében) alkalmasan megválasztott konstans.

A futási idő most is $\exp(c\sqrt{\log n \log\log n})$, de egy az eddigieknél kisebb c konstanssal.

Hivatkozások

[1] N. Koeblitz, *A Course in Number Theory and Cryptography*, Springer, 1994.

[2] C. Pomerance, *Analysis and comparison of some integer factoring algorithms*, elérhető: Computational Methods in Number Theory, Part I, H. W. Lenstra, Jr., R. Tijdeman, szerk., Math. Centre Tract 154, Amsterdam, 1982, pp 89-139.

11.5. Számtest szita

Számelméletben az általános számtest szita [1] klasszikus algoritmus, amely 10^{100}-nál nagyobb egész számok faktorizálására ismert.

Ebben egy a faktorizálandó n szám ismeretében, ügyesen rögzített f irreducibilis polinomot használva, keresünk B-sima számokat egy, az f polinomon alapuló S halmaz értékei között.

Az algoritmus leírása megtalálható [1]-ben. Az algoritmus futási ideje $\exp(c(\log n)^{1/3}(\log\log n)^{2/3})$.

Hivatkozások

[1] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse, J. M. Pollard, *The number field sieve*, kiterjesztett absztrakt: Proc. 22nd Annual ACM Sympos. Theory of Computing (STOC) (Baltimore, May 14-16, 1990), 564-572.

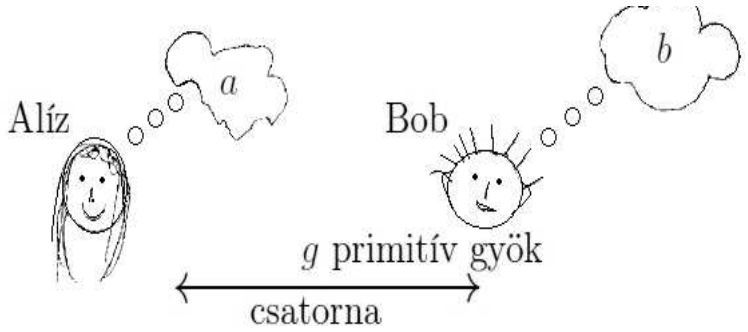
12. Diffie–Hellman kulcscsere

A Diffie-Hellman [1], [2] kulcscsere a nyilvános kulcsú kriptográfia egyik legfontosabb fejezete, melyben a felek úgy szeretnének megállapodni egy közös titkos kulcsban, hogy ha minden kommunikációjuk nyilvános, mások akkor se tudják kitalálni a közös titkos kulcsot. Az eljárás először [1]-ban jelent meg, így nevét a fenti cikk szerzőiről kapta, akik Merkle [2] egy ötletéből indultak ki. (Valóban, ezért 2002-ben Hellman javasolta, hogy az eljárást nevezzék át Diffie–Hellman–Merkle kulcscserére, és manapság ez az elnevezés is elterjedőben van ld. [3])

Tegyük fel, hogy a két fél, aki szeretne megállapodni egy közös titkos kulcsban Alíz és Bob, akik most egymástól távol vannak (más-más országban), és félnek attól, hogy a csatornát, amelyen kommunikálnak (telefon vagy email), lehallgatják. Hogyan állapodhatnak meg egy közös titkos kulcsban?

Tekintsük a legegyszerűbb esetet, amikor a közös kulcs $\mathbb{Z}_p^*$-nek egy (véletlen) eleme kell, hogy legyen. Alíz választ egy titkos $1 \leq a \leq p-1$, Bob választ egy titkos $1 \leq b \leq p-1$ egész számot, és ezt soha nem mondják ki, titokban tartják.

Megállapodnak egy közös g primitív gyökben $\bmod p$, ezt akár nyilvánosságra is hozhatják. Az se feltétlen szükséges, hogy g primitív gyök legyen (bár ideális esetben az), elég, hogy g rendje nagyon nagy.



Alíz kiszámolja	Bob kiszámolja
$g^a \pmod p$	$g^b \pmod p$

$$\xleftrightarrow{g^a,\, g^b}$$

Közös titkos kulcs: $g^{ab} \pmod p$

Alíz gyorsan ki tudja számolni g^{ab}-t, hiszen Bob elküldte neki $g^b \pmod p$-t, $a \in \mathbb{Z}$-t pedig ő találta ki, így $g^{ab} \pmod p$ gyorsan számolható egy egyszerű moduláris hatványozással:

$$\text{Alíz:} \quad g^{ab} \equiv (g^b)^a \pmod p.$$

Bob hasonlóan jár el: $g^{ab} \equiv (g^a)^b \pmod p$, tehát mindketten ki tudják számolni $g^{ab} \pmod p$-t.

Tegyük fel, hogy Éva lehallgatja a csatornát. Ekkor a-t és b-t ő nem ismeri, ezeket Alíz és Bob fejben tartotta, viszont esetleg megszerzi

$$g, g^a, g^b \qquad \pmod p$$

értékét. Évának ekkor $g^{ab} \pmod p$ kiszámolásához, ekkor az ún. Diffie–Hellman problémát kell megoldania. Ez:

Diffie–Hellman-probléma: A p prím, g primitív gyök, valamint g^a és $g^b \pmod p$ ismeretében számítsuk ki $g^{ab} \pmod p$-t. Rövidítése: DHP.

Sejtés. A Diffie–Hellman-probléma megoldására nincs gyors algoritmus.

Egy kapcsolódó probléma a diszkrét logaritmus probléma, melyet a következő alfejezetben tárgyalunk.

Az eljárás könnyen általánosítható $\mathbb{Z}_p^*$-ról, n-ed rendű ciklikus csoportokra, ahol a primitív gyök szerepét a csoport generátoreleme veszi át.

Hivatkozások

[1] W. Diffie, M. E. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory. 22 (6) (1976).

[2] R. C. Merkle, *Secure communications over insecure channels*, Communications of the ACM. 21 (4) (1978), 294–299.

[3] Wikipédia, *Diffie–Hellman key exchange*, https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

12.1. Diszkrét logaritmus probléma

A fejezetben megfogalmazott probléma a következő:

Diszkrét logaritmus probléma: Adott $c \in \mathbb{Z}_p{}^*$ és g primitív gyök esetén számítsuk ki azt az x-et, melyre

$$c \equiv g^x \pmod{p}.$$

Rövidítése: DLP.

Világos, hogy ha a DLP-re ismert gyors megoldás, akkor DHP-re is, hiszen

$$g^a, g^b \rightsquigarrow a, b \text{ adott } \rightsquigarrow (g^a)^b \equiv g^{ab} \pmod{p}$$

DLP-vel Moduláris hatványozás

Nem világos azonban, hogy ha DHP-re ismert gyors megoldás, akkor DLP-re is. Az az általános feltételezés, hogy ez a két probléma ekvivalens. De ez csak sejtés.

A következőkben a diszkrét logaritmus kiszámítására ismertetek egy-két (messze nem optimális, lassú) módszert. Az az általános hit, hogy a DLP nem oldható meg gyorsan. Ezt a feltételezést a kriptográfiában széles körben használják.

Hivatkozások

[1] Wikipédia, *Discrete logarithm*, https://en.wikipedia.org/wiki/Discrete_logarithm

12.2. Bébi–lépés–Óriás–lépés

Az alábbiakban Das könyve [1] alapján ismertetjük a **Bébi–lépés–Óriás–lépés** algoritmust.

Kissé talán régi módszer, de az az előnye, hogy nemcsak $\mathbb{Z}_p^*$-ban, hanem tetszőleges csoportban működik. Azonban általános csoportokban nem ismeretes ennél gyorsabb módszer, így például elliptikus görbéken alapuló csoportok esetén sem.

A következőkben $\mathcal{G}$ véges multiplikatív, ciklikus csoport, mérete n. Rögzítjük $\mathcal{G}$ egy generátorelemét: g-t.

Shanks [2] **Bébi lépés–Óriás lépés** algoritmusa:

Legyen m a $\sqrt{n}$ felső egészrésze, azaz $m = \lceil \sqrt{n} \rceil$. Kiszámítjuk $i = 1, 2, \ldots, m$ esetén g^i-t, és készítünk egy táblázatot. Pl.:

$$\mathcal{G} = {\mathbb{F}_{97}}^* \qquad g = 23 \qquad m = \left\lceil \sqrt{97} \right\rceil = 10$$

i	0	1	2	3	4	5	6	7	8	9
$g^i \pmod{97}$	1	23	44	42	93	5	18	26	16	77

.

Ez $O(\sqrt{n}(\log n)^2)$ bitoperáció, m-mel kifejezve az időigény: $O(m(\log m)^2)$. Ezután g^i értéke szerint rendezzük a táblázatot:

i	0	5	8	6	1	7	3	2	9	4
$g^i \pmod{97}$	1	5	16	18	23	26	42	44	77	93

.

Majd kiszámítjuk g^{-m}-t, ahol g^{-m} a g^m csoportelem inverzét jelöli, azaz

$$g^{-m}g^m = 1 \quad \mathcal{G} - \text{ben.}$$

Lagrange tétele szerint $g^n = 1$ minden $g \in \mathcal{G}$-re, ahol $n = |\mathcal{G}|$. Tehát

$$g^{-m} = g^{n-m} \qquad O((\log n)^3) \text{ bitoperációval számolhat,}$$

hiszen az ismételt négyzetre emelés algoritmust alkalmazva megkapjuk g^{-m}-et. A példában: $(\mathcal{G} = \mathbb{Z}_{97}^*,\ n = 96,\ g = 23,\ m = \lceil\sqrt{97}\rceil = 10)$ kapjuk:

$$g^{-m} = 66 \qquad (\bmod 97)$$

Tegyük fel, hogy a DLP, amelyet meg kell oldanunk a következő: Az $a = g^x$ egyenlet megoldását keressük, ahol a és g adott. Ahogy $j = 1, 2, \ldots, m-1$, kiszámítjuk

$$ag^{-jm}$$

értékét. Majd megnézzük, megegyezik-e valamelyik g^i-vel a táblázatban. Ez $O(\log n)$ darab összehasonlítás minden egyes j-re, így az időigény $O(\log n)^2$. Az összes j-re megvizsgálva, az időigény: $O(\sqrt{n}(\log n)^2)$.

Ha találtunk i-t, amelyre

$$ag^{-jm} = g^i \qquad \mathcal{G}\text{-ben,}$$

akkor

$$a = g^{jm+i},$$

és megoldottuk a diszkrét logaritmus problémát. Mivel minden x felírható $jm + i$ alakban, ez az algoritmus valóban mindig ad megoldást.

Hátránya: lassú $O(\sqrt{n}(\log n)^3)$ időigény, és nagy $O(\sqrt{n})$ tárhelyigény.

Hivatkozások

[1] A. Das, *Computational Number Theory*, CRC Press, 2013.

[2] D. Shanks, *Class number, a theory of factorization and genera*, Proceedings of the Symposia in Pure Mathematics 20 (1971), 415-440.

12.3. Pollard-féle ϱ-módszer

Ez a módszer a faktorizálásra vonatkozó ϱ módszer [3] adaptációja. A módszert Das könyve [1] alapján ismertetem, azonban Pollard a [4] cikkben publikálta a módszer első variánsát.

Tegyük fel, hogy az

$$a = g^x$$

DLP-t szeretnénk megoldani egy $\mathcal{G}$ ciklikus csoportban, ahol $|\mathcal{G}| = n$, és g egy generátoreleme $\mathcal{G}$-nek. Ehhez egy véletlen sétát konstruálunk $\mathcal{G}$-ben. Kezdő elem:

$$w_0 = g^{s_0} a^{t_0}.$$

Ezt követően $i = 1, 2, 3, \ldots$, tekintjük a séta i-edik elemét

$$w_i = g^{s_i} a^{t_i}$$

alakban. Ekkor $w_0, w_1, w_2, \ldots$ úgy viselkedik, mint egy véletlen séta $\mathcal{G}$-ben. A születésnap–paradoxon [2] szerint nagy valószínűséggel (99%) ebben a sétában lesz egy egybeesés $20\sqrt{n}$ lépés után. Ekkor

$$g^{s_i} a^{t_i} = g^{s_j} a^{t_j}.$$

Rendezve

$$a^{t_i - t_j} = g^{s_j - s_i}.$$

Mivel g egy generátoreleme $\mathcal{G}$-nek, ezért rendje n, és így:

$$(t_i - t_j) \operatorname{ind}_g a \equiv s_j - s_i \pmod{n}.$$

Ha $(t_i - t_j, n) = 1$, akkor

$$\mathrm{ind}_g a \equiv (s_j - s_i)(t_i - t_j)^{-1} \pmod{n}.$$

Ahhoz, hogy a módszer működjön, szükség van egy megfelelő $f : \mathcal{G} \to \mathcal{G}$ függvényre, amely w_{i-1}-hez hozzárendeli w_i-t.

Ehhez rögzítünk egy viszonylag kicsi pozitív egész r-et, és $\forall w \in \mathcal{G}$-hez hozzárendeljük a $\{0, 1, 2, \ldots, r-1\}$ halmaz egy elemét. Ezenkívül generálunk r darab „szorzót”,

$$M_j = g^{\sigma_j} a^{\tau_j}, \qquad j = 0, 1, 2, \ldots, r-1.$$

Ha w $(= g^{s_i} a^{t_i})$-hez $u \in \{0, 1, \ldots, r-1\}$-t rendeltük, akkor

$$f(w) = w \cdot M_u \quad \left(= g^{s_i} a^{t_i} \cdot g^{\sigma_u} a^{\tau_u}\right).$$

$$\uparrow$$

csoportbeli szorzás

Ez a módszer a gyakorlatban $r \approx 20$ értékkel jól működik. A módszer időigénye: $O(\sqrt{n}(\log n)^3)$. Egy ügyes ötlettel a módszer tárhely igényét minimálisra csökkenthetjük azon az áron, hogy az időigény kétszeresére változik. Tudjuk, hogy nagy valószínűséggel kb. $20\sqrt{n}$ lépés után lesz egy egybeesés, de ezután minden lépésnél lesz egy-egy újabb egybeesés, hiszen a séta a kapott hurokban megy körbe-körbe. Így tudjuk, hogy nagy valószínűséggel $20\sqrt{n}$ lépés után rögzítve egy elemet, az az elem már a sétának a hurok részében van. Elég ezt a rögzített elemet behelyezni a tartós tárba, és mindig az aktuális elemmel összehasonlítani, hiszen újabb $20\sqrt{n}$ lépés után (ami hosszabb mint a séta hurok része) ezzel a rögzített elemmel is lesz egy egybeesés.

Hivatkozások

[1] A. Das, *Computational Number Theory*, CRC Press, 2013.

[2] Wikipédia, *Birthday problem*, https://en.wikipedia.org/wiki/Birthday_problem

[3] J. M. Pollard, *A Monte Carlo method for factorization*, BIT Numerical Mathematics. 15 (3) (1975), 331–334.

[4] J. M. Pollard, *Monte Carlo methods for index computation (mod p)*, Mathematics of Computation. 32 (143) (1978), 918–924.

12.4. Pollard-λ-módszer

Pollard ezt a módszert ugyanabban a cikkben ismertette mint a ρ módszert, ld [2]. Ez a módszer a Pollard-féle ϱ-módszer kicsit módosított változata, amelyet Das [1] könyve alapján ismertetek. Az egyetlen különbség a ρ módszerhez képest, hogy most két séta van:

$$w_i = f(w_{i-1}) \quad \text{és} \quad w_i{}' = f(w_{i-1}{}').$$

Amikor a két véletlen séta találkozik, onnantól egybeesnek a séták további részei, s így a két séta együtt úgy néz ki, mint a görög λ betű. Ha

$$w_i = w_j{}',$$

akkor

$$g^{s_i}a^{t_i} \equiv g^{s_j{}'}a^{t_j{}'}, \quad g^{s_i}a^{t_i} = g^{s_j{}'}a^{t_j{}'},$$

így

$$a^{t_i - t_j} = g^{s_j' - s_i}.$$

Innen

$$(t_i - t_j{}')^{-1}(s_j{}' - s_i) \equiv \ \mathrm{ind}_g a \qquad (\mathrm{mod}\ n).$$

Ezt a módszert úgy is szokás nevezni, hogy a „vad és szelíd kenguruk módszere”. A szelíd kenguru sétája minden egyes lépése során ás egy csapdát, és amikor a vad kenguru eléri a séta egy pontját, beleesik a csapdába, és megfogták...

Hivatkozások

[1] A. Das, *Computational Number Theory*, CRC Press, 2013.

[2] J. M. Pollard, *Monte Carlo methods for index computation (mod p)*, Mathematics of Computation. 32 (143) (1978), 918–924.

[3] Pat Whelen fotója a Pexels oldaláról, https://www.pexels.com/hu-hu/foto/5615406/

12.5. Pohlig–Hellman-módszer

Most is legyen $|\mathcal{G}| = n$, ahol n prímtényezős felbontása

$$n = {p_1}^{\alpha_1} {p_2}^{\alpha_2} \dots {p_r}^{\alpha_r}.$$

Pohlig és Hellman [2] kitalált egy algoritmust, amely a diszkrét logaritmust

$$O\left(\sqrt{\max\{p_1, \dots, p_r\}}(\log n)^c\right)$$

időigénnyel számolja. Vagyis a módszer akkor hatékony, ha n legnagyobb prímosztója is kicsi. Úgy tűnik az algoritmust Roland Silver fedezte fel először, de eredményét nem publikálta, így időnként az algoritmust Silver-Pohlig-Hellman módszernek is szokás nevezni. A következő ismertető Das [1] könyve alapján készült.

Úgy próbáljuk meghatározni x-et, hogy minden egyes i-re meghatározzuk x-nek $p_i^{\alpha_i}$-vel vett osztási maradékát, majd a kínai maradéktétellel kiszámoljuk x-et.

Feladatunk így a következő: Legyen p prím és $p^\alpha \mid n$, de $p^{\alpha+1} \nmid n$. Meghatározandó $a = g^x$ esetén x-nek p^α-val vett osztási maradéka. Ekkor:

$$\begin{aligned} a &= g^x, && \backslash\hat{}\{n/p^\alpha\} \\ a^{n/p^\alpha} &= \left(g^{n/p^\alpha}\right)^x, \end{aligned}$$

de itt g^{n/p^α} rendje p^α, így a $\mathcal{G}'$ csoportban kell megoldanunk az

$$a' = (g')^x \tag{12.1}$$

DLP-t, ahol $a' = a^{n/p^\alpha}$ és $g' = g^{n/p^\alpha}$. Ez megadja x-nek p^α-val vett osztási maradékát: $r_{p^\alpha}(x)$-t. Nézzük tehát (12.1) DLP probléma megoldását. Ehhez a következőkben írjuk fel $r_{p^\alpha}(x)$-et

$$r_{p^\alpha}(x) = x_0 + x_1 p + x_2 p^2 + \cdots + x_{\alpha-1} p^{\alpha-1} \tag{12.2}$$

alakban. Először meghatározzuk x_0-t, majd x_1-t, majd x_2-t és így tovább. Nézzük ennek az algoritmusnak egy lépését. Tegyük fel, hogy $x_0, x_1, x_2, \ldots, x_{i-1}$ adott és keressük x_i-t. Legyen

$$\lambda = x_0 + x_1 p + \cdots + x_{i-1} p^{i-1}.$$

Tudjuk:

$$a' = (g')^x \qquad \mathcal{G}' - ben,$$

így

$$a'(g')^{-\lambda} = (g')^{x_i p^i + x_{i+1} p^{i+1} + \cdots + x_{\alpha-1} p^{\alpha-1}} \qquad \mathcal{G}' - ben.$$

Felemelve $p^{\alpha-i-1}$-dik hatványra

$$\left(a'(g')^{-\lambda}\right)^{p^{\alpha-i-1}} = (g')^{x_i p^{\alpha-1}+x_{i+1}p^{\alpha}+\cdots+x_{\alpha-1}p^{2\alpha-i-2}} \qquad \mathcal{G}' - ben.$$

De g' rendje p^{α}, így

$$\left(a'(g')^{-\lambda}\right)^{p^{\alpha-i-1}} = (g')^{x_i p^{\alpha-1}} = \left((g')^{p^{\alpha-1}}\right)^{x_i} \qquad \mathcal{G}' - ben.$$

Azaz tulajdonképpen az

$$a'' = (g'')^{x_i}$$

DLP-t kell megoldanunk a $\mathcal{G}'' =< (g')^{p^{\alpha-1}} >$ csoportban, ahol $a'' = \left(a'(g')^{-\lambda}\right)^{p^{\alpha-i-1}}$ és $g'' = (g')^{p^{\alpha-1}}$. Ekkor $\mathcal{G}''$ csoport rendje p.

A fenti algoritmussal megadjuk az $x_0, x_1, \ldots, x_{\alpha-1}$ számjegyeket, és így megkapjuk x-nek p^{α}-nal vett osztási maradékát. Végigfuttatva az algoritmust n összes prímhatvány osztójára, majd a kapott maradékokra alkalmazva a kínai maradéktételt, megkapjuk x-et.

Hivatkozások

[1] A. Das, *Computational Number Theory*, CRC Press, 2013.

[2] S. Pohlig, M. Hellman, *An improved algorithm for computing logarithms over GF(p) and its cryptographic significance*, IEEE Transactions on Information Theory 24 (1978), 106-110.

12.6. Index-kalkulus módszer

A továbbiakban Das [1] könyve alapján az index-kalkulus módszert ismertetem. Ez a faktorbázis algoritmus adaptációja, habár történelmileg előbbre nyúlik mint az. Az algoritmus Western és Millertől [3] származik, akik cikkükben felhasználták Kraitchik [2] ötleteit.

A faktorbázis t darab kicsi prímet tartalmaz (t értéke később meghatározandó):

$$B = \{p_1, p_2, \ldots, p_t\}.$$

Az első lépés során olyan u-kat keresünk, amelyekre g^u hatványnak p-vel vett osztási maradéka B-szám. Azaz

$$g^u \equiv {p_1}^{\gamma_1} \ldots {p_t}^{\gamma_t} \pmod{p}.$$

Ilyen u-ból kell $s \gg t$ darab (ahol $\gg$ azt jelenti, hogy körülbelül $s \geq 2t$). Az indexekre nézve a következőt kapjuk:

$$\begin{aligned} \gamma_{11} \ \mathrm{ind}_g\,(p_1) + \ldots + \gamma_{1t} \ \mathrm{ind}_g(p_t) &\equiv u_1 \pmod{p-1} \\ &\vdots \\ \gamma_{s1} \ \mathrm{ind}_g\,(p_1) + \ldots + \gamma_{st} \ \mathrm{ind}_g(p_t) &\equiv u_s \pmod{p-1} \end{aligned}$$

Itt γ_{ij}-k adottak, és a lineáris egyenletrendszert megoldva, megkapjuk $\mathrm{ind}_g(p_1), \ldots, \ \mathrm{ind}_g(p_t)$ értékeit.

Most $g^x = a$. A második lépés során olyan α-t keresünk, amire ag^α egy B-szám, azaz

$$ag^\alpha \equiv {p_1}^{\gamma_1} \ldots {p_t}^{\gamma_t} \pmod{p}.$$

Ebből $\mathrm{ind}\, a$ számolható:

$$\mathrm{ind}\, a \equiv -\alpha + \gamma_1 \ \mathrm{ind}\, p_1 + \ldots + \gamma_s \ \mathrm{ind}\, p_s \pmod{p-1}.$$

Hivatkozások

[1] A. Das, *Computational Number Theory*, CRC Press, 2013.

[2] M. Kraitchik, *Théorie des nombres*, Gauthier–Villards, 1922.

[3] A. E. Western, J. Miller, *Tables of indices and primitive roots*, Royal Society Mathematical Tables, vol 9 (1968), Cambridge University Press.

12.7. Nullaismeretű protokoll

A Nullaismeretű protokoll (angolul „Zero-knowledge protocol”) olyan kriptográfiai módszer neve, amelyet az 1980-as évek elején találtak ki, a diszkrét logaritmus problémához kapcsolódóan [2]. A módszer célja az, hogy ha valaki szeretné bebizonyítani másoknak, hogy meg tud oldani egy kriptográfiai problémát, de anélkül, hogy a megoldás módszeréről bármit elárulna. Erre remek példa a diszkrét logaritmus probléma.

Tegyük fel, hogy Picara szeretné bebizonyítani a többieknek, hogy be tudja bizonyítani az

$$a = g^x$$

diszkrét logaritmus problémát, de mindezt úgy szeretné megtenni, hogy a többieknek semmit sem árul el x-ről. (Picara nevének kezdőbetűje az angol „prover” szó kezdőbetűjével egyezik meg.)

Tehát a és g adott, Picara azt állítja, kiszámolta x-et, de senkinek nem árulja el x értékét. Meggyőzhet-e minket, hogy ismeri x értékét?

A hitelesítő személy (angolul „verifier”), hívjuk őt Vivalenek, a következőképp ellenőrzi:

1. lépés: Picara generál egy véletlen $y < p - 1$ számot. Elküldi Vivalenak $a' = g^y$-t.

2. lépés: Vivale feldob egy pénzérmét.
   Ha fej: Picara elküldi Vivalenak y-t. Vivale ellenőrzi $a' = g^y$ tényleg?
   Ha írás: Picara elküldi Vivalenak $x + y \pmod p$. Vivale ellenőrzi, $a'a = g^{x+y}$?

Ez a két lépés addig ismétlődik, amíg Vivale meggyőződik, hogy Picara tényleg tudja a megoldást, vagy Picara lebukik, hogy blöfföl.

Valóban, ha Picara nem csal, azaz mind a' mind a hozzátartozó y ismeretében van, akkor az $a'a = g^{x+y}$ diszkrét logaritmus probléma megoldása ekvivalens az $a = g^x$ diszkrét logaritmus probléma megoldásával.

Ha Vivale fejt dob, akkor ő ellenőrizheti, hogy valóban $a'a = g^{x+y}$ teljesüle, azaz Picara tudja-e a megoldását a vele ekvivalens $a = g^x$ diszkrét logaritmus problémának is. Ugyanakkor ebben az esetben Vivale (ellentétben Picarával) csak a'-t ismeri, y számára titok, így ő nem jut közelebb az $a = g^x$ megoldásához (kivéve, ha megtudja oldani az $a' = g^y$ diszkrét logaritmus problémát, viszont számára ez nehéz, hiszen ez egy másik nehéz diszkrét logaritmus probléma és Vivale nincs olyan módszer birtokában, amellyel ez megoldható).

Elvileg elképzelhető, hogy Picara csal, azaz valójában nem is ismeri y-t, csak generál egy véletlen z-t, majd utána kiszámolja azt az a'-t, amelyre $a' = g^z a^{-1}$, azaz $a'a = g^z$, és tervezi, hogy elküldi Vivalenak z-t mint $x+y$-t. Ekkor, ha Vivale írást dob a pénzérmével Picara valóban nem bukik le. Igen ám, de ha Vivale fejet dob, akkor Vivale y-ra kérdez rá, és erre Picara nem tud válaszolni, csak akkor, ha ő tudja $a' = g^y$ megoldását. Viszont az $a' = g^y$ egyenlet ismeretével Vivale semmivel nem jut közelebb az $a = g^x$ diszkrét logaritmus probléma megoldásához.

Ha Vivale elég sokszor ismétli a fenti ellenőrző procedúrát, akkor Picara ha csal, nagyon nagy valószínűséggel előbb-utóbb lebukik. Viszont ha Picara soha nem bukik le, akkor gyakorlatilag biztos, hogy ismeri az $a = g^x$ diszkrét logaritmus probléma megoldását. Ugyanakkor Vivale, akár fejet dob, akár írást, ő semmivel sem jut közelebb az $a = g^x$ diszkrét logaritmus probléma megoldásához. Így a fenti valóban egy nullaismeretű protokoll.

Nullaismeretű protokollok további leírása megtalálható pl. [1], [4] is. Számelmélethez kapcsolódó nullaismeretű protokollok megtalálhatóak pl. [3]-ben.

Hivatkozások

[1] M. Blum, P. Feldman, S. Micali, *Non-interactive zero-knowledge and its applications*, Proceedings of the Twentieth Annual ACM Symposium on

Theory of Computing (STOC 1988), 103–112.

[2] D. Chaum, J.-H. Evertse; J. van de Graaf, *An improved protocol for demonstrating possession of discrete logarithms and some generalizations*, Advances in Cryptology – EuroCrypt '87, in: Proceedings. Lecture Notes in Computer Science. Vol. 304. (1987), 127–141.

[3] N. Koeblitz, *A Course in Number Theory and Cryptography*, Springer, 1994.

[4] H. Wu, F. Wang, *A survey of non interactive zero knowledge proof system and its applications*, The Scientific World Journal. 2014.

13. Elliptikus görbéken alapuló kriptográfia

A számelmélet kriptográfiai alkalmazásai során sokszor egy $\mathbb{F}_p$ véges testben számolunk (ahol p prím), esetleg annak kiterjesztésében $\mathbb{F}_q$-ban (ahol q prímhatvány). Gyakran azonban jól jön, ha a csoport, amelyben a műveleteket végezzük, más jellegű, a fentieknél „véletlenszerűbb" struktúrával rendelkezik, ezáltal nagyobb biztonságot garantálva az esetleges feltörési kísérletek ellen. Erre a célra tökéletesen megfelelnek az elliptikus görbéken definiált csoportok.

Magának az elliptikus görbéknek a története Diofantoszig nyúlik vissza, aki természetesen a mai jelölések, azonosságok, csoportösszeadás nélkül, elemileg vizsgálta az $y(a-y) = x^3 - x$ görbét [3]. Erről és az elliptikus görbék további történetéről rövid érdekes összefoglalót olvashatunk a [1] cikkben.

Az elliptikus görbék kriptográfiai használatát egymástól függetlenül Koeblitz [4] és Miller [5] javasolta 1985-ben. A széles körű kriptográfiai alkalmazások azonban csak 2004-től vannak elterjedve.

Az téma iránt mélyebben érdeklődő olvasóknak javaslom az angol nyelvű „Handbook of Elliptic and Hyperelliptic Curve Cryptography" [2] című könyv tanulmányozását.

Az elliptikus görbék definiálása során olyan algebrai konstrukciókat adunk meg, mely rengeteg ($\mathbb{F}_q{}^*$-oknál „sokkal több"), jól számolható Abel-csoportot ad meg. Ezek:

13.1. Definíció. a) *Legyen K test, többnyire $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{C}$ vagy $\mathbb{F}_q$ valamely q-ra, melynek karakterisztikája 3-nál nagyobb, és legyen $x^3 + ax + b$ $(a, b \in K)$ harmadfokú polinom, melynek nem létezik többszörös gyöke. Egy* elliptikus görbe *K felett azon (x, y) $(x, y \in K$-val$)$ pontok halmaza, melyekre*

$$y^2 = x^3 + ax + b \tag{13.1}$$

plusz még egy egyetlen 0*-val jelölt, a „végtelenben levőnek" hívott pont.*

b) *Ha K 2 karakterisztikájú test, akkor egy K feletti elliptikus görbe azon (x, y) pontok halmaza, melyekre vagy*

$$y^2 + cy = x^3 + ax + b, \tag{13.2}$$

vagy

$$y^2 + xy = x^3 + ax^2 + b \tag{13.3}$$

teljesül adott $a, b, c \in K$-val (most a jobb oldali harmadfokú polinomnak lehet többszörös gyöke is) + „a végtelenben levő” 0 pont.

c) *Ha K karakterisztikája 3, akkor egy K feletti elliptikus görbe*

$$y^2 = x^3 + ax^2 + bx + c, \tag{13.4}$$

ahol a jobb oldali polinomnak nem létezik többszörös gyöke.

Ahhoz, hogy az elliptikus görbén jól értelmezett csoportot tudjunk definiálni, a görbének nem szinguláris nak kell lennie, ami azt jelenti, hogy a görbének „sima”, vagyis nincsenek csúcspontjai vagy metszéspontjai. Ez ekvivalens azzal, hogy a görbének minden pontja nem szinguláris, amit a következőképp definiálunk:

13.2. Definíció. *Írjuk* (13.1)*-et (vagy hasonlóan* (13.2), (13.3), (13.4)*-t) $F(x, y) = 0$ alakban:*

$$F(x, y) \stackrel{\text{def}}{=} y^2 - (x^3 + ax + b) = 0.$$

Azt mondjuk, hogy egy (x, y) pont a görbén „nem szinguláris” (vagy „sima”), ha $\dfrac{\partial F}{\partial x} \neq 0$, $\dfrac{\partial F}{\partial y} \neq 0$ közül legalább az egyik teljesül az (x, y) pontban.

A jegyzetben nem bizonyítjuk, de belátható a feltétel, hogy az (13.1), illetve (13.4) jobb oldalán álló polinomnak nem létezik többszörös gyöke, akkor a görbén minden pont nem szinguláris. Diszkriminánst használva azt is tudjuk, hogy pl. az (13.1) esetben ez azzal ekvivalens, hogy $4a^3 + 27b^2 \neq 0$.

A fejezet további részeiben legtöbbször olyan testekre szorítkozunk, amelynek karakterisztikája nagyobb mint 3, azaz elliptikus görbénk (13.1) **alakú**:

$$\mathbf{y^2 = x^3 + ax + b}.$$

A legtöbb esetben az elliptikus görbe test felett definiált, de olyan alkalmazás is van, ahol egy $\mathbb{Z}_n$ csoport felett. Itt az elliptikus görbén alapuló faktorizáció módszerére gondolunk, amikor is nem test felett definiáljuk az elliptikus görbét, hanem egy $\mathbb{Z}_n$ csoport felett, ahol (általában) n összetett szám.

Az elliptikus görbék alkalmazhatósága azon múlik, hogy a görbe pontjai Abel-csoportot alkotnak alkalmas műveletre nézve. Ehhez műveletet kell definiálni a görbe pontjai között. A szemléletesség kedvéért először a $K = \mathbb{R}$-re szorítkozunk (más testekre ugyanígy megy, csak ott nem működik a geometriai szemléltetés), ez lesz a következő alfejezetünk témája.

Hivatkozások

[1] M. W. Barsagade, Dr. S. Meshram, *Overview of history of elliptic curves and its use in cryptography*, International Journal of Scientific & Engineering Research 5 (4) (2014), https://www.ijser.org/researchpaper/Overview-of-History-of-Elliptic-Curves-and-its-use-in-cryptography.pdf.

[2] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, Chapman and Hall/CRC 2005, https://www.hyperelliptic.org/HEHCC/

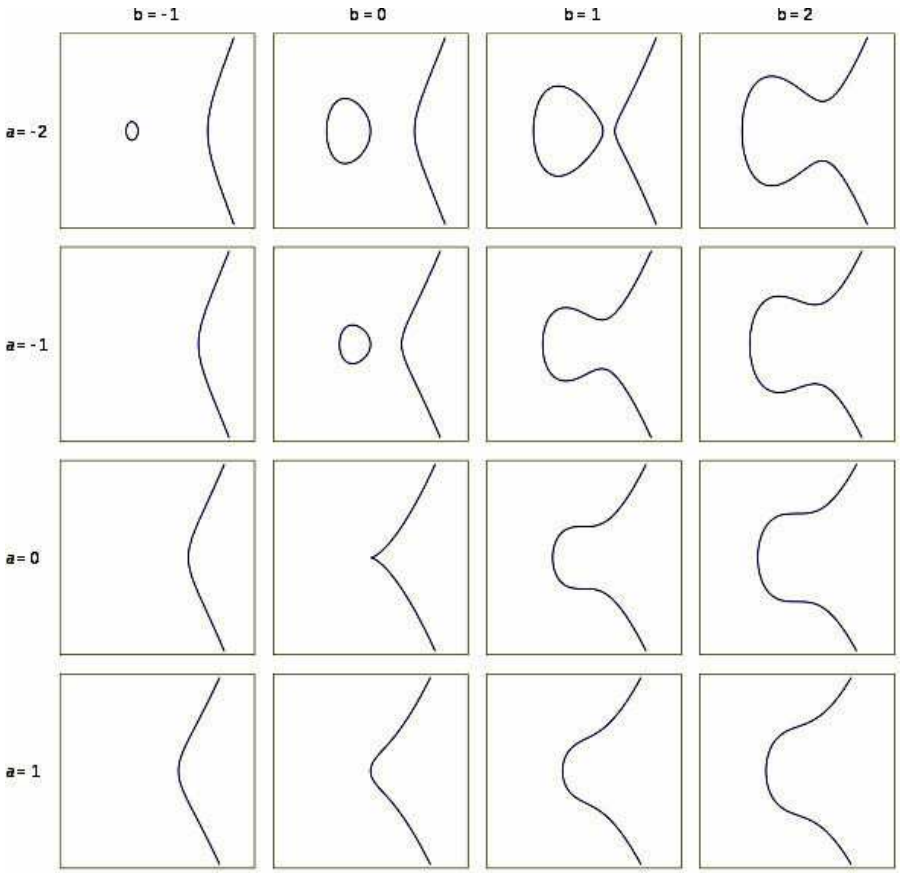
[3] Heath, Thomas Little, Sir, 1861-1940; Euler, Leonhard, 1707-1783, *Diophantus of Alexandria; a study in the history of Greek algebra*, https://archive.org/details/diophantusofalex00heatiala/page/n5/mode/2up.

[4] N. Koeblitz, *Elliptic curve cryptosystems*, Mathematics of Computation. 48 (177) (1985), 203–209.

[5] V. S. Miller, *Use of elliptic curves in cryptography*, Advances in Cryptology — CRYPTO '85, Proceedings. CRYPTO, Lecture Notes in Computer Science. Vol. 85 (1985), 417–426.

13.1. Elliptikus görbék $\mathbb{R}$ felett

Ebben a fejezetben $\mathbb{R}$ felett értelmezett elliptikus görbék pontjain definiálunk egy összeadást. Mielőtt azonban rátérnénk az összeadás definíciójára, mutatunk egy ábrát $\mathbb{R}$ feletti elliptikus görbék lehetséges formáiról.



Ezután definiálhatjuk az összeadást. Először a szemléletes geometriára

alapozott definíciót adjuk meg.

13.3. Definíció. *Legyen E elliptikus görbe $\mathbb{R}$ felett, P, Q pedig E-nek két pontja. A végtelen távoli pontot továbbra is 0-val jelöljük. Ekkor:*

1. $P + 0 = 0 + P = P$.

2. *Ha $P = (x_P, y_P) \neq 0$ és $Q = (x_Q, y_Q) \neq 0$, akkor*

a) *$x_P \neq x_Q$ esetén a P, Q-t összekötő egyenes egy $R = (x_R, y_R)$ pontban metszi a görbét. Legyen $P+Q$ az R tükörképe az x tengelyre, azaz:*

$$P + Q \stackrel{\text{def}}{=} (x_R, -y_R).$$

b) *$x_P = x_Q$ esetén vagy*

b_1) *$y_P = y_Q$, azaz $P = Q$, ekkor az érintő P-ben egyetlen más $R = (x_R, y_R)$ pontban metszi a görbét, legyen megint*

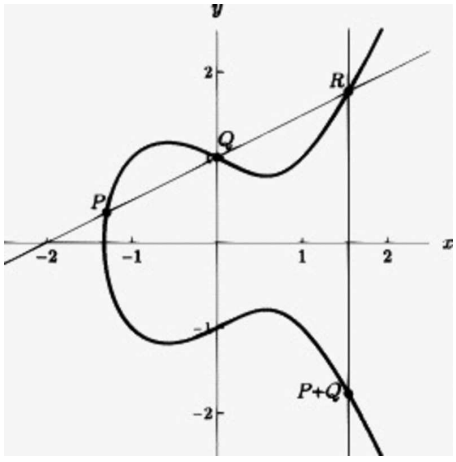
$$P + Q = 2P \stackrel{def}{=} (x_R, -y_R)$$

(ha P inflexiós pont: $R = (x_R, y_R) = P$).

b_2) *$y_P = -y_Q$: ekkor*

$$P + Q \stackrel{\text{def}}{=} 0$$

.



Annak bizonyítása, hogy a 13.3. Definícióban megadott összeadás jól definiált, valamint ez az összeadás E pontjain egy Ábel csoportot alkot, megtalálható pl. Silverman könyvében [1]. Világos az is, hogy ebben a csoportban az egységelem a 0, a végtelen távoli pont hiszen

$$P + 0 = 0 + P.$$

A $P = (x, y)$ elem inverze pedig $-P = (x, -y)$.

Elemi geometriával kiszámolhatók a következő képletek (ld. [1]):

Ha $P = (x_P, y_P)$ és $Q = (x_Q, y_Q)$, ahol $P \neq \pm Q$, akkor a PQ egyenes meredeksége:

$$m = \frac{y_p - y_Q}{x_P - x_Q}.$$

A PQ egyenes $R = (x_R, y_R)$ pontban metszi a görbét, itt:

$$\begin{aligned} x_R &= m^2 - x_P - x_Q \\ y_R &= y_p + m(x_R - x_P), \end{aligned} \tag{13.5}$$

vagy ezzel ekvivalensen

$$y_R = y_Q + m(x_R - x_Q).$$

Lássuk (13.5) bizonyítását. A görbénk $y^2 = x^3 + ax + b$ alakú, míg a PQ egyenes $y = mx + d$ alakú. A PQ egyenesnek is és a görbének is P, Q és R pontja, tehát x_P, x_R és x_Q is megoldása az

$$(mx + d)^2 = x^3 + ax + b$$

egyenletnek. Ezt rendezve

$$x^3 - m^2x^2 - 2mdx + ax + b - d^2 = 0.$$

A fenti egyenletnek x_P, x_R és x_Q is gyökei, tehát a gyökök és együtthatók közötti összefüggés szerint:

$$x_P + x_Q + x_R = m^2,$$

ami igazolja (13.5)-t.

Mivel $S = P + Q$, ezért az S pont az R pont x tengelyre vett tükörképe, így $S = P + Q = (x_S, y_S)$ koordinátáira:

$$x_S = m^2 - x_P - x_Q,$$
$$y_S = -y_p + m(x_P - x_R) = -y_Q + m(x_Q - x_R)$$

teljesül.

A P és Q pontok összeadása során, amennyiben $P = Q$, akkor a görbének a P pontbeli meredekségét kell felírni, ez

$$m = \frac{3x_p^2 + a}{2y_p}.$$

Ekkor az $S = P + Q = (x_S, y_S)$ koordinátáira:

$$x_S = m^2 - x_p - y_p,$$
$$y_S = -y_p + m(x_P - x_R) = -y_Q + m(x_Q - x_R)$$

adódik.

Végül $P = -Q$ esetén $P + Q = 0$.

Hivatkozások

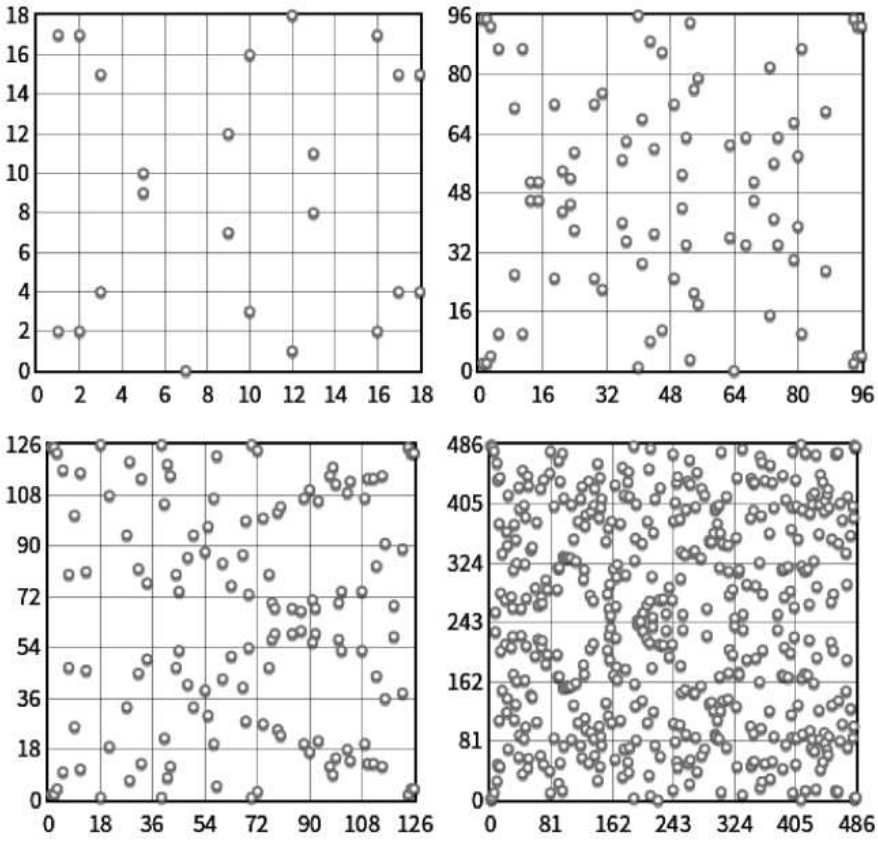
[1] J. H. Silverman, *The Arithmetic of Elliptic Curves*, Graduate Texts in Mathematics, Vol. 106, Springer-Verlag, 1986.

[2] Ábra, Wikipédia, *Elliptikus görbék lehetséges alakjai*, https://en.wikipedia.org/wiki/Elliptic_curve

[3] Ábra, *Csoport összeadás elliptikus görbéken*, https://www.researchgate.net/figure/The-group-law-for-an-elliptic-curve-P-Q-R-The-points-P-and-Q-sum-to-the-point-R_fig1_23552588

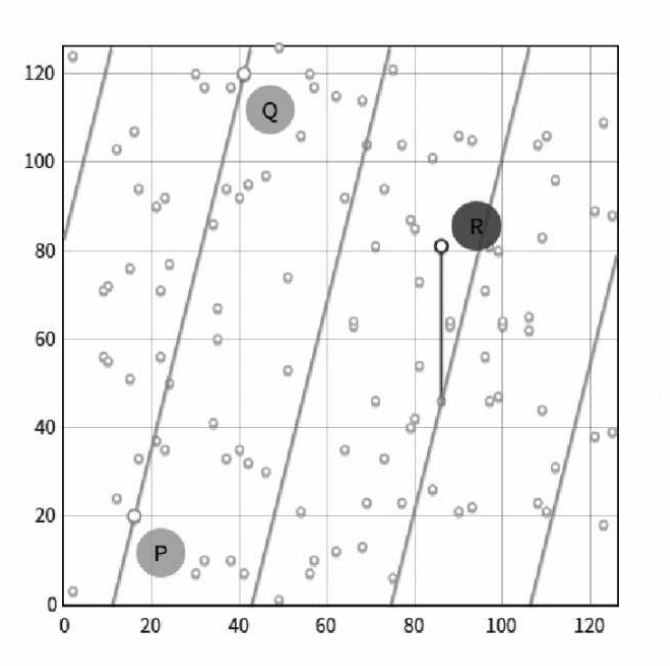
13.2. Elliptikus görbék $\mathbb{F}_p$ felett

Kriptográfiai alkalmazások során legtöbbször az elliptikus görbék egy $\mathbb{F}_p$ test felett értelmezettek. Sajnos, ekkor a valós felett ábrázolt szép folytonos görbe eltűnik, helyette egy diszkrét pontokból álló ábrát kapunk. Az alfejezetben Corbellini [1] ismeretterjesztő írására támaszkodunk, a fejezetben lévő ábrák is tőle származnak. A következő ábra az $y^2 = x^3 - 7x + 10$ elliptikus görbe pontjait mutatja meg az $\mathbb{F}_p$ testek felett, ahol p rendre $19, 97, 127$ és 487.



Vegyük észre, hogy a görbén adott x koordinátával legfeljebb 2 pont létezik, és az ábra mindig szimmetrikus az $y = p/2$ egyenesre.

Jól látható, hogy a görbe egy P és Q pontján átmenő egyenes képe más lesz mint a valós esetben. Azonban szerencsére az egyeneseknek van egy koordináta geometriai képlete, $ax+by = c$, amellyel $\mathbb{F}_p$ felett is dolgozhatunk. A P, Q pontokon átmenő egyenes most is egy harmadik R pontban metszi a görbét, amelynek az $y = p/2$ egyenesre vett tükörképe lesz a $P + Q$. Ezt szemlélteti a következő ábra, mely az $y^2 = x^3 - x + 3 \pmod{127}$ elliptikus görbén ábrázolja a $P = (16, 20)$ és $Q = (41, 120)$ pontok összegét.



Látható, hogy ha az elliptikus görbe feletti test véges, akkor a geometriai definíció csak nehezen kezelhető. Az algebrai képletekkel definiált összeadás azonban minden további nélkül átvihető, erre az esetre:

13.4. Definíció. *Ha $P = (x_P, y_P)$ és $Q = (x_Q, y_Q)$, ahol $P \neq \pm Q$, akkor a PQ egyenes meredeksége:*

$$m = \frac{y_p - y_Q}{x_P - x_Q}.$$

Amennyiben $S = P + Q$, akkor az S pont az R pont x tengelyre vett tükör-

képe, így $S = P + Q = (x_S, y_S)$ *koordinátáira:*

$$x_S = m^2 - x_P - x_Q,$$
$$y_S = -y_p + m(x_P - x_R) = -y_Q + m(x_Q - x_R)$$

teljesül.

Amennyiben $P = Q$*, akkor a görbének a* P *pontbeli meredekségét kell felírni, ez*

$$m = \frac{3x_p^2 + a}{2y_p}.$$

Ekkor az $S = P + Q = (x_S, y_S)$ *koordinátáira:*

$$x_S = m^2 - x_p - y_p,$$
$$y_S = -y_p + m(x_P - x_R) = -y_Q + m(x_Q - x_R)$$

adódik.

Végül $P = -Q$ *esetén* $P + Q = 0$.

Nagyon fontos kérdés, hogy egy $\mathbb{F}_q$ véges test felett definiált E elliptikus görbének hány pontja lehet. Hasse tétele szerint [2] ez az érték $q+1$-től csak $2\sqrt{q}$-val térhet el:

13.5. Tétel. (Hasse)

$$|\text{card } E(\mathbb{F}_q) - (q+1)| \leq 2q^{1/2}.$$

Sejtésként a tételt Artin fogalmazta meg 1924-ben szakdolgozatában. Csak 12 évvel később sikerült igazolnia Hasse-nek, bizonyítását egy cikk sorozatban ismertette [2]. Weil [5] tovább általánosított a becslést elliptikus görbéknél általánosabb görbékre.

Ismert az is, hogy az $E(\mathbb{F}_q)$ csoport szerkezete vagy ciklikus, vagy két ciklikus csoport direkt szorzata.

A kriptográfiai alkalmazások során gyakran szükségünk van az elliptikus görbe pontos rendjére (azaz pontos elemszámára). Schoof [3], [4] polinomiális

algoritmust adott erre, azonban az algoritmus leírása és bizonyítása túl megy a jelen jegyzet keretein.

Az elliptikus görbéken alapuló kriptográfiáról (ECC) rövid ismertetőt olvashatunk a kapcsolódó Wikipédia oldalon [6]. A jegyzetben a teljesség igénye nélkül betekintünk az ECC néhány fejezetébe. Így szó lesz a Diffie-Hellman kulcscsere analogonjáról, elliptikus görbékre alapuló digitális aláírásról és Lenstra faktorizációs algoritmusáról.

Hivatkozások

[1] A. Corbellini, *Elliptic Curve Cryptography: finite fields and discrete logarithms*, https://andrea.corbellini.name/2015/05/23/elliptic-curve-cryptography-finite-fields-and-discrete-logarithms/

[2] H. Hasse, *Zur theorie der abstrakten elliptischen funktionenkörper.* I, II & III, Crelle's Journal, 1936 (175), 55-62, 69-88, 293-208.

[3] R. Schoof, *Elliptic curves over finite fields and the computation of square roots mod p.* Math. Comp., 44 (170), 483–494, 1985, http://www.mat.uniroma2.it/~schoof/ctpts.pdf.

[4] R. Schoof, *Counting points on elliptic curves over finite fields*, J. Theor. Nombres Bordeaux 7, 219–254, 1995, http://www.mat.uniroma2.it/~schoof/ctg.pdf.

[5] A. Weil, *Numbers of solutions of equations in finite fields*, Bulletin of the American Mathematical Society, 55 (5) (1949).

[6] Wikipédia, *Elliptic-curve cryptography*, https://en.wikipedia.org/wiki/Elliptic-curve_cryptography.

[7] Ábra, A. Corbellini, *Elliptic Curve Cryptography: finite fields and discrete logarithms*, https://andrea.corbellini.name/2015/05/

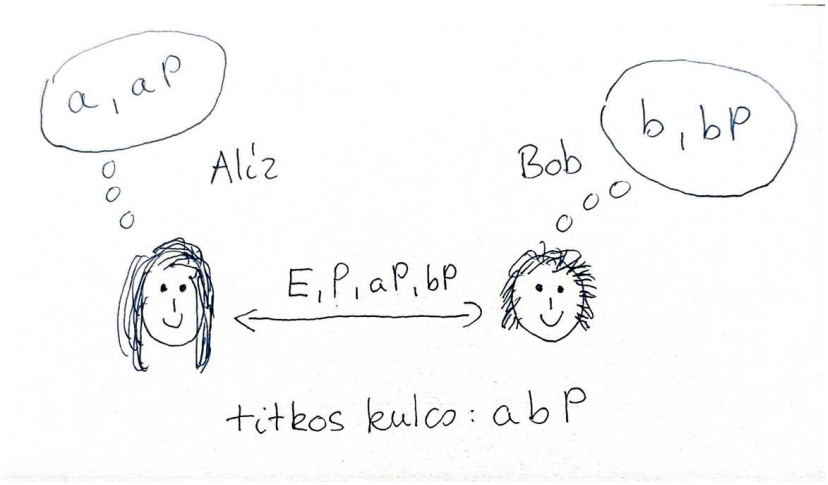
23/elliptic-curve-cryptography-finite-fields-and-discrete-logarithms/

13.3. Diffie-Hellman kulcscsere elliptikus görbéken

Tegyük fel, hogy Alíz és Bob szeretne megállapodni egy közös titkos kulcsban, amelyet most egy $\mathbb{F}_p$ feletti elliptikus görbe egy S pontja szimbolizál. (S-nek könnyen megfeleltethetjük $\mathbb{F}_p$ egy pontját, pl. vesszük S-nek az x vagy y koordinátáját.) Éva figyeli a csatornát, amelyen Alíz és Bob kommunikál. Kérdés, hogy vajon Éva ki tudja-e találni a közös titkos kulcsot S-et abból, amit a csatornán lehallgat.

A Diffie-Hellman kulcscsere elliptikus görbékre vonatkozó analogonja a következő:

Alíz és Bob megállapodik egy nyilvános E elliptikus görbében $\mathbb{F}_p$ felett, és annak egy P pontjában, amelynek rendje egy nagy prím. Alíz gondol egy a természetes számot, Bob gondol egy b természetes számot. A gondolt számot mindketten titokban tartják, senkinek nem árulják el, még egymásnak sem. Alíz kiszámolja aP-t, Bob kiszámolja bP-t. A csatornán elküldik egymásnak aP és bP értékét, a közös titkos kulcs pedig abP.



De hogyan számolható ki gyorsan az elliptikus görbe egy tetszőleges P pontjának többszöröse, mondjuk cP? Ez a duplázás és összeadás algoritmussal történik. Írjuk fel c-t kettőhatványok összegeként:

$$c = 2^{a_1} + 2^{a_2} + \cdots + 2^{a_r},$$

ahol 2^{a_1} a legnagyobb kettőhatvány. Először a $P_i = 2^i P$ alakú többszöröseit számoljuk ki, ahol $i = 1, 2, \ldots, a_1$, a következő rekurzióval:

$$\begin{aligned} P_0 &= P, \\ P_i &= P_{i-1} + P_{i-1} \qquad \text{ha } i \geq 1. \end{aligned}$$

Majd:

$$cP = P_{a_1} + P_{a_2} + \cdots + P_{a_r}.$$

Ez az algoritmus nagyon gyors polinomiális idejű. (Itt megjegyezzük, hogy Morain és Olivos [7] észrevette, hogy ha c-t a kettes számrendszerben, de $\{0, 1, -1\}$ számjegyekkel írjuk fel és törekszünk arra, hogy a felírás minél rövidebb és minél kevesebb nem 0 számjegyet tartalmazzon, akkor $25-30\%$-os gyorsaság növelést lehet elérni. Ennek oka, hogy elliptikus görbéken az összeadás és a kivonás lényegében ugyanannyi idő alatt elvégezhető.)

Tehát a ismeretében Alíz gyorsan ki tudja számolni aP-t, Bob pedig b ismeretében bP-t. Mindketten gyorsan ki tudják számolni a közös titkos kulcsot, hiszen Alíz b-szer összeadja aP-t, Bob pedig a-szor összeadja bP-t.

Általános a feltétezés azonban, hogy Évának ahhoz hogy ki tudja találni a közös titkos kulcsot, meg kell tudnia határozni a-t vagy b-t, de ő ehhez csak a P, aP és bP pontjait ismeri a görbének. Ez egy diszkrét logaritmus probléma, amelyet bővebben a 12. fejezetben tárgyaltunk.

Van az algoritmusnak még néhány kérdéses pontja. Például, Alíz és Bob, hogyan talál egy nagy rendű közös P pontot? Ez fordítva történik mint ahogy gondolnánk. Tehát nem az van, hogy veszünk egy véletlen P pontot, kiszámítjuk a rendjét, és ha ez a rend nagy akkor megtartjuk P-t, ha kicsi új

véletlen P ponttal próbálkozunk. Schoof algoritmusa alapján [9], [10] ugyan a görbe rendjét meg tudjuk határozni, de ez a görbe tetszőleges pontjának a rendjének a meghatározására alkalmatlan. Ehhez képest fordítva haladunk. Kiszámítjuk a görbe rendjét, legyen ez N. Ennek az N-nek vesszük egy nagy méretű n prímosztóját. Ezután keresünk egy n-ed rendű P pontot. Ehhez választunk egy véletlen R-t. Kiszámoljuk:

$$P = \frac{N}{n}R.$$

Ha ez 0, akkor új véletlen R-t választunk, ha nem 0, akkor viszont P rendje n.

Könnyen gondolhatunk arra, hogy elliptikus görbék kriptográfiai alkalmazására bármilyen véletlen görbe jó. Ez koránt sincs így. Így például Smart támadása [11] alapján, ha az $\mathbb{F}_p$ feletti görbe rendje pont p, akkor a diszkrét logaritmus lineáris időben megoldható. Gondolhatunk még itt a nevezetes MOV támadásra is [6] (pl. amikor a görbe rendje pont $p + 1$). Szerencsére az ilyen, anomálisnak nevezet görbék száma viszonylag kevés. A fenti és hozzá hasonló jellegű támadások kizárására 1999-ben a NIST közzétett egy publikációt [8] az akkor biztonságosnak ítélt elliptikus görbékről. Elliptikus görbékre alapuló kriptográfia elleni támadásokról, illetve bizonyos elliptikus görbék gyenge kriptográfiai tulajdonságairól olvashatunk még pl. [1], [4], [5] és [3]-ban is. Az utolsó referencia kvantum-számítógépes támadásokhoz is kapcsolódik.

Fontos kérdés, hogy miért jobb vagy megbízhatóbb egy elliptikus görbére alapozott ECC titkosítási rendszer mint egy hagyományos titkosítási rendszer? AZ ECC titkosítási kulcsok mérete jóval kisebb mint az előtte használtaké. Ezt szemlélteti a NIST által közzétett táblázat:

Symmetric Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	Elliptic Curve Key Size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Table 1: NIST Recommended Key Sizes

Látható a kulcsok mérete közötti nagyságrendbeli különbség. Így kisebb méretű elektronikus eszközök pl. mobiltelefonok esetén kézenfekvőbb az *ECC* alapú titkosítás.

Hivatkozások

[1] , I. Biehl, B. Meyer, V. Müller, *Differential fault attacks on elliptic curve cryptosystems*, Advances in Cryptology – CRYPTO 2000. Lecture Notes in Computer Science Vol. 1880. (2000), 131–146, https://www.iacr.org/archive/crypto2000/18800131/18800131.pdf.

[2] A. Corbellini, *Elliptic Curve Cryptography: finite fields and discrete logarithms*, https://andrea.corbellini.name/2015/05/23/elliptic-curve-cryptography-finite-fields-and-discrete-logarithms/

[3] L. De Feo, P. Jao, Plut, *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies*, Cryptology ePrint Archive, Report

2011/506. IACR,
https://eprint.iacr.org/2011/506.

[4] M. Hedabou, P. Pinel, L. Beneteau, *A comb method to render ECC resistant against Side Channel Attacks*,
https://eprint.iacr.org/2004/342.pdf.

[5] *How to design an elliptic-curve signature system*, Cr.yp.to: 2014.03.23,
http://blog.cr.yp.to/20140323-ecdsa.html

[6] A. Menezes, T. Okamoto, S. A. Vanstone, *Reducing elliptic curve logarithms to logarithms in a finite field*, IEEE Transactions on Information Theory. 39 (5) (1993), 1639–1646.

[7] F. Morain, J. Olivos, *Speeding up the computations on an elliptic curve using addition-subtraction chains*, RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications 24.6 (1990), 531-543, http://eudml.org/doc/92374

[8] National Institute of Standards and Technology, *Recommended Elliptic Curves for Federal Government Use*, July 1999.
http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.doc

[9] R. Schoof, *Elliptic curves over finite fields and the computation of square Roots mod p*. Math. Comp., 44 (170), 483–494, 1985,
http://www.mat.uniroma2.it/~schoof/ctpts.pdf.

[10] R. Schoof, *Counting points on elliptic curves over finite fields*, J. Theor. Nombres Bordeaux 7, 219–254, 1995,
http://www.mat.uniroma2.it/~schoof/ctg.pdf.

[11] N. P. Smart, *The discrete logarithm problem on elliptic curves of trace one*, 1997,
http://www.hpl.hp.com/techreports/97/HPL-97-128.html.

13.4. Elliptikus görbén alapuló digitális aláírás

A digitális aláírásoknak a mai informatika világában rendkívül fontos jelentősége van. A kézzel írt hagyományos aláírások mellett ma már gyakran elektronikusan is szükséges igazolni, hogy egy adott dokumentum tőlünk származik.

A digitális aláírás ötletét Diffie és Hellman az [2] cikkben vetették fel, de csak később vált megvalósíthatóvá, amikor az RSA-t publikálták. Évtizedekig tartott, amíg a digitális aláírást jogszabályokban is a hagyományossal egyenértékűnek ismerték el. Azóta digitális aláírásokra sokféle algoritmust találtak ki a matematikusok. A fejezet témájának megfelelően most az elliptikus görbéken alapuló digitális aláírásról (ECDSA) lesz szó.

Az ECDSA használatát először Vanstone [4] javasolta 1992-ben, amelyet az ISO (International Standard Organization) 1998-ban fogadott el, az ANSI (American National Standard Institute) pedig 1999-ben. Az ECDSA főbb kriptográfiai tulajdonságairól 2001-ben összefoglaló cikk jelent meg Johnson, Menezes és Vanstone [3] tollából. Ebben a fejezetben mi csak röviden elemezzük az ECDSA-t, Corbellini [1] munkája alapján.

Tegyük fel, hogy Alíz szeretne elektronikusan aláírni egy üzenetet. Az ECDSA esetében ehhez szüksége van egy nyilvános p prímre, egy E elliptikus görbére $\mathbb{F}_p$ felett, abban egy prímrendű G pontra. Ezek mind nyilvánosak. Jelöljük G rendjét n-nel. Ekkor n prímszám. (Itt megjegyezzük, hogy a legtöbb standard használatban lévő elliptikus görbe rendje prímszám, azaz n megegyezik az E elliptikus görbe rendjével, így nincs szükség E rendjének faktorizálására.) Alíznak van még egy titkos kulcsa még, amely egy természetes szám 1 és n között, ezt jelölje d_A. Alíz nyilvános kulcsa az elliptikus

görbe

$$H_A = d_A G$$

pontja.

Fontos, hogy bárki a világon meggyőződhessen arról, hogy valóban Alíz írta alá az üzenetet.

Például, hívjuk történetünk másik szereplőjét Bobnak, aki szeretne megbizonyosodni arról, hogy az adott üzenetet valóban Alíz írta alá. Ehhez használhatja az E elliptikus görbét, G-t, az aláírt dokumentumot és Alíz nyilvános kulcsát H_A-t.

Legtöbbször Alíz nem az eredeti üzenetet írja alá (hiszen az általában túl hosszú), hanem annak egy **hash függvénnyel** rövidített változatát. A kriptográfiailag biztonságos hash függvényeknek sok követelménynek kell eleget tennie, ezekről pl. [5]-ben olvashatunk bővebben. Az üzenet hash függvénnyel rövidített változata egy egész szám, amely bináris hossza nem lehet nagyobb mint n bináris hossza (ahol n a G pont által generált részcsoport rendje). Jelölje z az üzenet Hash függvénnyel rövidített változatát. Ekkor $z \in \mathbb{N}$ és z lehet nagyobb mint n, de kettes számrendszerbeli alakja nem lehet hosszabb n-nél.

Alíz algoritmusa az aláírás során a következő:

1. Alíz választ egy véletlen k-t az $\{1, 2, 3, \ldots, n-1\}$ halmazból.

2. Kiszámolja az E elliptikus görbe $P = kG$ pontját.

3. Aliz a P pont x-koordinátáját r-rel jelöli ($r = x_p$).

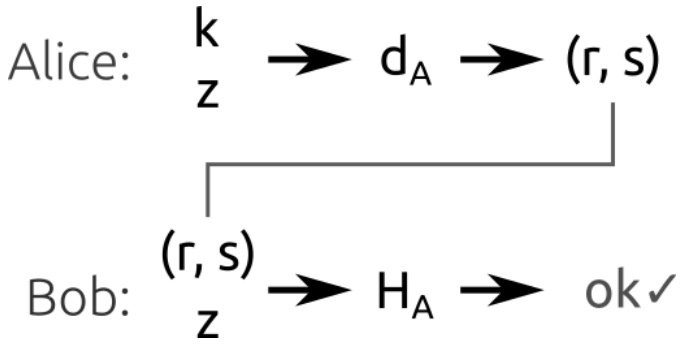
4. Amennyiben $r = 0$, másik véletlen k-t választ, és újra próbálkozik, azaz visszamegy az 1. lépéshez.

5. Kiszámolja $s \equiv k^{-1}(z + rd_A) \pmod{n}$ (ahol d_A Alíz titkos kulcsa, k^{-1} pedig k multiplikatív inverze modulo n.)

6. Amennyiben $s \equiv 0 \pmod{n}$, Alíz másik véletlen k-t választ, és újra próbálkozik, azaz visszamegy a 1. lépéshez.

A digitális aláírás az (r, s) pár.

(Megjegyezzük, hogy itt az 5. lépésben a műveletek a szokásos $\mathbb{Z}_n$-beli összeadás és szorzás. Ezzel szemben a 2. lépésben az elliptikus görbe pontjain értelmezzük az összeadást, és $kG = \underbrace{G + G + \cdots + G}_{k \text{ darab } G}$-t jelöli.)



Ezután rátérhetünk arra, hogy Bob hogyan ellenőrizheti az aláírás hitelességét.

1. Bob kiszámolja az $u_1 = s^{-1}z \pmod{n}$ számot.

2. Meghatározza a $u_2 = s^{-1}r \pmod{n}$ számot.

3. Kiszámolja az elliptikus görbe $P = u_1G + u_2H_A$ pontját.

A digitális aláírás akkor érvényes, ha $r = x_P$.

Első ránézésre nem nyilvánvaló miért működik ez az ellenőrző módszer, de ha összerakjuk az egyenleteket, akkor minden világossá válik:

Tudjuk, hogy $P = u_1G + u_2H_A$ és $H_A = d_AG$, így

$$P = u_1G + u_2H_A$$

$$= u_1 G + (u_2 d_A) G$$
$$= (u_1 + u_2 d_A) G.$$

Ha visszaemlékezünk u_1 és u_2 definíciójára, azaz $u_1 = s^{-1} z \pmod{n}$ és $u_2 = s^{-1} r \pmod{n}$, akkor pedig

$$\begin{aligned} P &= (u_1 + u_2 d_A) G \\ &= (s^{-1} z + s^{-1} r d_A) G \\ &= s^{-1} (z + r d_A) G. \end{aligned}$$

Előzőleg s-et $k^{-1}(z + r d_A)$-val adtuk meg, így $ks = z + r d_A$, amiből

$$P = kG.$$

Ezzel az ellenőrző módszer helyességét igazoltuk.

Ma már ECDSA-t használnak pl. az interneten keresztüli kommunikációhoz védelmet biztosító TLS protokol során. Biztonsága azon múlik, hogy az elliptikus görbéken való diszkrét logaritmus probléma megoldására nincs gyors algoritmus. (Vagyis a privát kulcs ismerete nélkül a $H_A = d_A G$ egyenletből, H_A és G ismeretében nincs gyors algoritmus d_A privát kulcs kiszámolására.) Így amennyiben az elliptikus görbe paraméterei elég nagyok, még számítógépek segítségével sem oldható meg az adott elliptikus görbén megadott diszkrét logaritmus problémák.

Fontos, hogy az algoritmus során használt véletlen k-t csak egyszer szabad használni, ismételt alkalmazás esetén Alíz privát kulcsa gyorsan számolhatóvá válik. Ugyanígy az is nagyon fontos, hogy k-t valóban véletlen módon válasszuk, ugyanis, ha k valamilyen módon előre megjósolható, akkor d_A könnyen meghatározható. Erről bővebben Corbellini jegyzetében [1] és az ECDSA-hoz kapcsolódó Wikipédia [6] oldalon olvashatunk.

Hivatkozások

[1] A. Corbellini, *Elliptic Curve Cryptography: ECDH and ECDSA*, https://andrea.corbellini.name/2015/05/30/elliptic-curve-cryptography-ecdh-and-ecdsa/

[2] W. Diffie, M. E. Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory. 22 (6) (1976), 644-654.

[3] D. Johnson, A. Menezes, S. Vanstone *The Elliptic Curve Digital Signature Algorithm (ECDSA)*, International Journal of Information Security 1, 36–63 (2001).

[4] S. Vanstone, *Responses to NIST's Proposal*, Communications of the ACM 35 (1992), 50-52.

[5] Wikipédia, *Cryptographic bash function*, http://en.wikipedia.org/wiki/Cryptographic_hash_function

[6] Wikipédia, *Elliptic Curve Digital Signature Algorithm*, https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm.

[7] Ábra, A. Corbellini, *Elliptic Curve Cryptography: ECDH and ECDSA*, https://andrea.corbellini.name/2015/05/30/elliptic-curve-cryptography-ecdh-and-ecdsa/

13.5. Lenstra elliptikus görbéken alapuló faktorizációja

Az eddigiekben az elliptikus görbéket mindig egy test felett definiáltuk. Van egy fontos eset, amikor az elliptikus görbe nem test felett értelmezett, hanem egy $\mathbb{Z}_n$ csoport felett. Ez Lenstra [3] faktorizációs algoritmusa. Amennyiben n összetett az elliptikus görbének lesznek pontjai, amelyeket

nem tudunk összeadni a szokásos összeadással, ugyanis a két ponton átmenő egyenes meredekségében a nevező nem lesz invertálható modulo n. Ez az első ránézésre kellemetlen tulajdonság azonban valójában nagyon is hasznos: n egy valódi osztójának megtalálásához vezethet, ugyanis a nevező és n legnagyobb közös osztóját kiszámítva megkapjuk n egy osztóját.

A faktorizációs módszer kiindulópontja Pollard $p-1$ módszere [4]. Ezt az algoritmust csak nagyon röviden ismertetjük, hiszen ennél sok gyorsabb módszer létezik manapság. Viszont ez az a módszer, amely kiindulópontja Lenstra [3] algoritmusának, ami a mai napig a leggyorsabb faktorizációs algoritmusok között tartandó számon.

Az algoritmus akkor működik, ha az n egy p prímosztójára $p-1$ egy B-hatványsima szám, azaz $p-1$ minden prímhatvány osztója $\leq B$. Másképpen megfogalmazva $p-1 \mid [1,2,3,\ldots,B]$, ahol a szögletes zárójel a legkisebb közös többszöröst jelenti. A továbbiakban legyen

$$L_B = [1,2,3,\ldots,B].$$

A kis-Fermat tétel szerint, ha p prím, és $(a,p)=1$ akkor

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

Amennyiben $p-1$ egy B hatványsima szám, azaz $p-1 \mid L_B$, akkor a fenti kongruenciát $L_B/(p-1)$-edik hatványra emelve kapjuk

$$a^{L_B} \equiv 1 \pmod{p}.$$

Azaz $p \mid n$ esetén

$$p \mid (n, a^{L_B}-1).$$

Így amennyiben az Eukleideszi algoritmussal kiszámoljuk n és $a^{L_B}-1$ legnagyobb közös osztóját, amennyiben az n-nél kisebb, megkapjuk n egy valódi osztóját. Ha nem járunk sikerrel, más a-t választunk. Az algoritmusban B

értékét folyamatosan növeljük: $B = 1, 2, 3, \ldots$. Pollard cikkének [4] érdekessége, hogy az algoritmus időigényét Turing gépek működési ideje alapján határozza meg.

Pollard $p-1$ módszere akkor működik gyorsan, ha $p-1$ egy B hatványisima szám, viszonylag kicsi B-re. Itt a $p-1$ a $\mathbb{Z}_p^*$ multiplikatív csoport rendje. Lenstra erről a csoportól áttért $\mathbb{Z}_p$ felett definiált elliptikus görbék csoportjaira. Ez azért lesz jó, mert egy ilyen csoport rendje Hasse tétele [2] alapján

$$p + 1 \pm k$$

alakú, ahol $k \leq 2\sqrt{p}$. Amennyiben az a $p + 1 \pm k$ rend egy B hatványisima szám, az algoritmus jó eséllyel meg fogja adni az n összetett szám egy valódi osztóját.

A következőkben ismertetjük Lenstra algoritmusának leegyszerűsített változatát. Legyen n egy összetett szám, amelyet faktorizálni szeretnénk, B pedig egy n függvényében alkalmasan választott konstans. Először vesszük egy $P = (x_P, y_P)$ pontot $\mathbb{Z}^2$-ből, majd olyan $a \neq 0$ és b egész számokat választunk, amire

$$6(4a^3 + 27b^2)$$

relatív prím n-hez, továbbá a $\mathbb{Z}_n$ felett definiált

$$y^2 = x^3 + ax + b$$

elliptikus görbének P pontja, azaz

$$b = y_p^2 - x_p^3 - ax_P.$$

Ekkor ismételt összeadással tekintjük az

$$P,\ 2!P,\ 3!P, \cdots, B!P$$

pontjait az elliptikus görbének. Ha ugyanezt a görbét és ugyanezen pontokat $\mathbb{Z}_p$ felett vizsgáljuk, akkor amennyiben a görbe rendjére, s-re

$$s \mid B!,$$

akkor Lagrange tétele alapján $B!P$ a 0 pontot adja, amikor a görbét $\mathbb{Z}_p$ felett vizsgáljuk. Mit jelent ez? Az ismételt összeadás során volt egy olyan R és S pontja a görbének (ahol nyilván $R = aP$ és $S = bP$ alakú), amikor R és S-en átmenő egyenes meredekségének a nevezője p-vel osztható volt, azaz $p \mid x_R - x_S$. Vagyis, ha R és S-et a $\mathbb{Z}_n$ feletti görbén adjuk össze, akkor az összeadás eredménye vagy a 0, vagy egyszerűen nem tudjuk összeadni a két pontot, hiszen a rajtuk átmenő egyenes képletében $x_R - x_S$ nem invertálható. Ha pedig nem tudjuk összeadni a két pontot, akkor

$$1 < (x_R - x_S, n) < n$$

kiszámításával megkapjuk n egy valódi osztóját. A $P,\ 2!P,\ 3!P, \cdots, B!P$ pontokat ismételt összeadással számoljuk ki. Valóban, ha $P_i = i!P$, akkor $P_i = iP_{i-1}$ gyorsan számolható, pl. a duplázás és összeadás algoritmussal. Ha nem tudunk az eljárás során két pontot összeadni, mondjuk R-et és S-et, akkor

$$1 < (x_R - x_S, n) < n$$

kiszámításával megkapjuk n egy valódi osztóját.

Térjünk vissza az elliptikus görbe $\mathbb{Z}_p$ feletti rendjére, s-re. Hasse tétele [2] szerint $s = p + 1 \pm k$, ahol $k \leq 2\sqrt{p}$. Amennyiben elegendően nagy B természetes számra

$$s \mid B!$$

bekövetkezik, akkor $\mathbb{Z}_p$ feletti összeadásként $B!P$ a 0 elem, ilyenkor a $\mathbb{Z}_n$ feletti is vagy a 0 elem, vagy pedig nem értelmezhető az összeadás, amikor is megkapjuk n egy valódi osztóját.

Az eljárás során az is előfordulhat, hogy a $\mathbb{Z}_n$ feletti összeadások során a

$$P,\ 2!P,\ 3!P, \cdots, B!P$$

pontok mind az elliptikus görbe elemei. Ilyenkor új P pontot és új elliptikus görbét választunk, egészen addig, amíg meg nem kapjuk n egy valódi osztóját.

Az algoritmus időigénye $\exp\left((\sqrt{2}+o(1))(\log p)^{1/2}\log\log p\right)$, ahol p az n legkisebb prímtényezője.

Híres verseny volt az RSA Faktorizációs Felhívás (ld. [1], [7]), amelyet az RSA Laboratórium tett közzé 1991-ben, ahol az általuk megadott félprímek (két nagy prím szorzata) faktorizációja volt a feladat, jelentős díjazás fejében. Egy-egy nagy szám faktorizálásáért, akár 20 000 $-t is adtak akkoriban. Az utolsó kettőt az RSA-240-et és RSA-250-t (12-13 évvel a határidő lejárta után) 2019-ben és 2020-ban faktorizálták az általános számtest szita és az elliptikus görbéken alapuló faktorizáció kombinált alkalmazásával [5], [6]. Az eredmények megjelenése után az alkalmazott területen dolgozó szakértők azt javasolták, hogy az RSA alkalmazása során legalább 2048 bit hosszú modulust biztonságos használni (bár az első ezirányú javaslat már a XX. század végén is megjelent).

Sajnos a verseny díjazása 2007-ben befejeződött, de reméljük, lesz még hasonló felhívás a jövőben is...

Köszönöm a figyelmet!

Hivatkozások

[1] K. Burt (18 Mar 1991), *Announcement of "RSA Factoring Challenge"*, Retrieved 8 March 2021, https://groups.google.com/u/0/g/sci.crypt/c/AA7M9qWWx3w/m/EkrsR69CDqIJ?pli=1.

[2] H. Hasse, *Zur theorie der abstrakten elliptischen funktionenkörper.* I, II & III, Crelle's Journal, 1936 (175), 55-62, 69-88, 293-208.

[3] H. W. Lenstra, Jr., *Factoring integers with elliptic curves*, Annals of Mathematics Second Series, 126, (3) (1987), 649-673, https://www.jstor.org/stable/1971363.

[4] J. M. Pollard, *Theorems of factorization and primality testing*. Proceedings of the Cambridge Philosophical Society. 76 (3) 521–528.

[5] E. Thomé et al. (December 2, 2019), *795-bit factoring and discrete logarithms*, cado-nfs-discuss (Mailing list), https://sympa.inria.fr/sympa/arc/cado-nfs/2019-12/msg00000.html.

[6] P. Zimmermann et al. (28 February 2020), *Factorization of RSA-250*, cado-nfs-discuss (Mailing list), https://sympa.inria.fr/sympa/arc/cado-nfs/2020-02/msg00001.html.

[7] Wikipédia, *RSA Factoring Challenge*, https://en.wikipedia.org/wiki/RSA_Factoring_Challenge.