Eötvös Loránd Tudományegyetem

Természettudományi Kar

Espán Márton

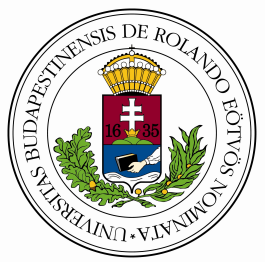
Az RSA titkosítás és a Diffie–Hellman kulcscsere matematikai alapjai

BSc alkalmazott matematikus szakdolgozat

Témavezető:

Seres István András

Komputeralgebra Tanszék



Budapest, 2023

Köszönetnyilvánítás

Szeretném megköszönni a segítséget elsősorban Seres István Andrásnak, aki a irányította a munkámat és rendelkezésemre bocsájtotta a szakirodalmat. Köszönöm Kiss Emil Tanár Úrnak, hogy belső konzulensként hozzájárult a szakdolgozat létrejöttéhez és készségesen segített a felmerülő kérdésekben, valamint köszönöm Somlai Gábor Tanár Úrnak a nyújtott szakmai segítséget. Ezen felül köszönöm Apagyi Dávidnak a latex szövegszerkesztő program használatával kapcsolatos meglátásait, tanácsait.

Tartalomjegyzék

Előszó

A szakdolgozatomban arra adok választ, hogy két elterjedt kriptográfiai módszernek, az RSA titkosításnak és a Diffie–Hellman kulcscserének mik a matematikai biztosítékai. Először bevezetjük az alapvető fogalmakat, tételeket, majd mutatunk néhány faktorizáló és diszkrét logaritmust számító algoritmust annak érdekében, hogy közelebb vigyük az olvasót ahhoz a feltevéshez, hogy a diszkrét logaritmus probléma, illetve a faktorizáció nehéz. Ezután megmutatjuk, hogyan történhet az RSA titkosításhoz nélkülözhetetlen prím generálás és a Diffie–Hellman kulcscseréhez szükséges ciklikus csoport generálás. Végül bonyolultságelméleti tételeket és feltevéseket mondunk ki és visszavezetjük a két vizsgált módszer nehézségét a faktorizációra és a diszkrét logaritmus problémára, melyek a modern kriptográfia legalapvetőbb feltevései, valamint érintjük ezek kapcsolatát is.

1. fejezet

Matematikai alapok

1.1. Számelmélet

1.1.1. Állítás. *Ha $a \in \mathbb{Z}$ és $b \in \mathbb{Z}^+$ akkor $\exists!$ q, r egészek, melyekre $0 \leq r < b$ és $a = q \cdot b + r$, továbbá q és r polinomiális futásidőidőben kiszámítható $\log a$ és $\log b$ szerint, azaz a és b hoszza szerint.*

1.1.2. Állítás. *Ha $a, b \in \mathbb{Z}^+$ akkor $\exists$ X, $Y \in \mathbb{Z}$ hogy $X \cdot a + Y \cdot b = \text{lnko}(a, b)$. Továbbá $\text{lnko}(a, b)$ a legkisebb ilyen pozitív szám.*

Bizonyítás. Legyen $I := \{\widehat{X} \cdot a + \widehat{Y} \cdot b \mid \widehat{X}, \widehat{Y} \in \mathbb{Z}\}$ és legyen d a legkisebb pozitív elem I-ben. Megmutatjuk, hogy $d = \text{lnko}(a, b)$. Legyen $c = X'a + Y'b \in I$ pozitív; $c = q \cdot d + r$; $0 \leq r < d$. Ekkor

$$r = c - qd = X'a + Y'b - q(Xa + Yb) = (X' - qX)a + (Y' - qY)b \in I,$$

de $r < d$ miatt $r = 0$. Tehát d oszt minden pozitív elemet I-ben, így $a, b \in I$ pozitív elemek esetén $d \mid a$ és $d \mid b$. Tegyük fel, hogy $\exists$ $d' > d$, melyre $d' \mid a$ és $d' \mid b$ amiből $d' \mid (Xa + Yb) = d$ következik, de ez ellentmond annak a feltételnek, hogy $d' > d$. □

Az X és Y számok valamint a legnagyobb közös osztó is kiszámolható a kiterjesztett Euklideszi algoritmus segítségével, melynek futásideje n bites számokra $\mathcal{O}(n^3)$.

Algoritmus 1: Kiterjesztett Euklideszi algoritmus

Input: A, B n bites pozitív egészek

Output: d, X, Y, ahol $d = XA + YB$ a legnagyobb közös osztó

```
1  Function LNKO(A, B):
2      if A>B then
3          return lnko(B, A, 0, 1, 1, 0)
4      else
5          return lnko(A, B, 1, 0, 0, 1)
   /* lnko hívásakor tudjuk, hogy a = u·A + v·B és b = w·A + z·B      */
6  Function lnko(a, b, u, v, w, z):
7      if a = 0 then
8          return (b, w, z)
9      b := c·a + r, ahol 0 ≤ r < a
10     lnko(r, a, (w − cu), (z − cv), u, v)
```

1.1.3. Állítás. *Ha $c \mid ab$ és $\text{lnko}(a, c) = 1$, akkor $c \mid b$, továbbá p prímre $p \mid ab$ esetén $p \mid a$ vagy $p \mid b$.*

Bizonyítás. Ha $c \mid ab$, akkor $nc = ab$; $n \in \mathbb{N}$. Továbbá ha $\text{lnko}(a, c) = 1$, akkor felírhtó $1 = Xa + Yc$ alakban. Innen $b = Xab + Ycb = Xnc + Ycb = c(Xn + Yb) \Rightarrow c \mid b$. □

1.1.4. Állítás. *Ha $a \mid N$; $b \mid N$ és $\text{lnko}(a, b) = 1$, akkor $ab \mid N$.*

Bizonyítás. Tudjuk, hogy $N = ac = bd$; $1 = Xa + Yb$ megfelelő c, d nemnegatív és X, Y egészek mellett. Ekkor $N = XaN + YbN = Xabd + Ybac = ab(Xd + Yc)$, így $ab \mid N$. □

1.1.5. Definíció (Moduláris inverz)**.** *Legyen $N \in \mathbb{N}^+$. Ha adott $b \in \mathbb{Z}$-hez létezik olyan $c \in \mathbb{Z}$, hogy $bc \equiv 1 \pmod N$, akkor b invertálható $\bmod N$ és c ekkor b-nek a multiplikatív inverze $\bmod N$, továbbá a b-vel való osztás $\bmod N$ megegyezik a b^{-1}-gyel való szorzással.*

1.1.6. Állítás. *Legyen b, $N \in \mathbb{Z}$; $b \geq 1$; $N > 1$. Ekkor b pontosan akkor invertálható $\bmod N$, ha $\text{lnko}(b, N) = 1$.*

Bizonyítás. $\rightarrow bc \equiv 1 \pmod N \Rightarrow bc - 1 = nN$; $n \in \mathbb{Z} \Rightarrow bc - nN = 1 \Rightarrow \text{lnko}(b, N) = 1$ (1.1.2 miatt).
$\leftarrow Xb + YN = 1$; $X, Y \in \mathbb{Z} \Rightarrow Xb \equiv 1 \pmod N \Rightarrow X$ a b multiplikatív inverze. □

1.1.7. Tétel (Kinai maradéktétel)**.** *Legyenek $m_1, m_2, \ldots, m_k > 0$ páronként relatív prímek, $\prod_{i=1}^{k} m_i = M$ és $c_1, c_2, .., c_k$ tetszőleges egészek. Ekkor az $x \equiv c_i \pmod{m_i}$ $i = 1, .., k$ kongruenciarendszer megoldható és a megoldás egyetlen maranékrendszer* mod M*.*

Bizonyítás. Egyértelműség: Tegyük fel, hogy x_1, x_2 megoldás. Ekkor $m_i|(x_1 - x_2)$ $\forall$ i-re $\Rightarrow M|(x_1 - x_2) \Rightarrow x_1 \equiv x_2 \pmod{M}$, ami ellentmondás.
Létezés: legyen $M_i = \dfrac{M}{m_i}$. Ekkor $(M_i, m_i) = 1$ miatt $\exists$ $z_i \in \mathbb{Z}$, hogy $M_i z_i \equiv 1 \pmod{m_i}$. Innen a megoldás $s = \sum_{i=1}^{k} M_i z_i c_i$, aminek belátásához igazolnunk kell, hogy $s \equiv c_i \pmod{m_i}$ $\forall$ i-re. Mivel $m_i \mid M_j$, ha $i \neq j$, ezért $s \equiv M_i z_i c_i \pmod{m_i}$, ebből pedig $s \equiv c_i \pmod{m_i}$, és készen vagyunk.

□

1.2. Csoportelméleti alapok

1.2.1. Definíció (Csoport)**.** *A G nem üres halmaz csoport a $\circ$ művelettel, ha a következők teljesülnek:*

1. *G zárt a csoportműveletre: $g, h \in G \Rightarrow g \circ h \in G$.*
2. *Létezik kétoldali neutrális elem: $\exists$ $e \in G$, hogy $\forall$ $g \in G$-re $e \circ g = g = g \circ e$.*
3. *Létezik kétoldali inverz: $\forall$ $g \in G$ elemhez $\exists$ $g^{-1} \in G$, hogy $g \circ g^{-1} = e = g^{-1} \circ g$.*
4. *A csoportművelet asszociatív: $g_1, g_2, g_3 \in G \Rightarrow (g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$.*

Könnyen belátható, hogy a neutrális elem, másnéven egységelem és az inverz is egyértelmű. Ha teljesül a kommutativitás is, azaz $g, h \in G \Rightarrow g \circ h = h \circ g$, akkor Abel-csoportról beszélünk. A csoport rendje a G halmaz számossága, jele $|G|$.
Multiplikatív csoportok esetén az egységelemet 1*-gyel, additív csoportok esetén* 0*-val jelöljük.*

1.2.2. Definíció (Részcsoport)**.** *Legyen G csoport a $\circ$ művelettel, Ekkor a $H \subseteq G$ részhalmaz részcsoport a $\circ$ művelettel, ha H maga is csoport a $\circ$ művelettel. Ha H részcsoportja G-nek annak jele $H \leq G$, ha valódi részcsoportja annak $H < G$.*

Ezentúl G-vel mindig csoportot fogunk jelölni.

1.2.3. Példa. *Főként az alábbi csoportokkal fogunk foglalkozni:*
$\mathbb{Z}_N = \{0, 1, \ldots, N-1\}$ *a művelet pedig:* $a, b \in \mathbb{Z}_N \Rightarrow a \circ b = [a + b \mod N]$.
$\mathbb{Z}_N^* = \{x \mid x \in \mathbb{Z}_N \wedge \text{lnko}(x, N) = 1\}$, *a művelet pedig:* $a, b \in \mathbb{Z}_N^* \Rightarrow a \circ b = [a \cdot b \mod N]$; $|\mathbb{Z}_N^*| = \phi(N)$, *ahol* ϕ *az Euler-féle fí-függvény. Ezek Abel-csoportok is, mivel a szorzás és az összeadás kommutatív..*

1.2.4. Állítás. *Ha* $a, b, c \in G$ *és* $ac = bc$, *akkor* $a = b$.

Bizonyítás. $ac = bc \Rightarrow (ac)c^{-1} = (bc)c^{-1} \Rightarrow a = b$. □

1.2.5. Állítás. *Legyen* $m = |G|$, *ekkor* $\forall\, g \in G$*-re* $g^m = 1$.

Bizonyítás. Abel-csoportokra bizonyítunk. Legyen $g \in G$ a csoport egy eleme, $g_1, .., g_m$ pedig a csoport összes eleme. Ekkor teljesül a következő egyenlőség:

$$g_1 \cdot g_2 \cdot \ldots \cdot g_m = (gg_1) \cdot (gg_2) \cdot \ldots \cdot (gg_m). \tag{1.1}$$

Az előző állítás szerint $gg_i = gg_j \Rightarrow g_i = g_j$. Így a bal oldal tényezői is mind különbözőek és a jobb oldaléi is, valamint mindkét oldalon fel van sorolva G összes eleme, így valóban fennál az egyenlőség. Az (1.1)-es egyenlőség mindkét oldalát $(g_1 \cdot g_2 \cdot \ldots \cdot g_m)^{-1}$-el megszorozva megkapjuk az állítást. □

1.2.6. Állítás. *Legyen* $m = |G|$, *bármely* $g \in G$*-re és* x *egészre:* $g^x = g^{(x \mod m)}$.

Bizonyítás. Legyen $x = qm + r;\ q, r \in \mathbb{Z};\ 0 \leq r < m$, ekkor $r = [x \mod m]$. Továbbá $g^x = g^{qm+r} = g^{qm}g^r = (g^m)^q g^r = 1^q g^r = g^r$, felhasználva az előző állítást. □

1.2.7. Állítás. *Legyen* $m = |G| > 1$, *továbbá tekintsük a következő függvényt:* $f_e : G \to G$, $f_e(g) = g^e\ \forall\, e > 0$*-ra. Ha* $\text{lnko}(e, m) = 1$, *akkor az* f_e *függvény bijekció és az inverze* f_d, *ahol* $d \equiv e^{-1} \pmod m$.

Bizonyítás. Mivel G véges ezért abból, hogy f_d az f_e inverze következik a bijekció is, így elegendő az előbbit bizonyítani.

$$f_d(f_e(g)) = f_d(g^e) = (g^e)^d = g^{ed} = g^{ed \mod m} = g^1 = g.$$

□

1.2.8. Tétel. *Legyen* $N = \prod_{i=1}^{n} p_i^{e_i}$, *ahol* $p_1, p_2 \ldots p_n$ *különböző prímek és* $\forall$ *i-re* $e_i \geq 1$. *Ekkor* $\phi(N) = \prod_{i=1}^{n} p_i^{e_i - 1}(p_i - 1)$.

A szakdolgozatban csak olyan N-el fogunk foglalkozni ami két prím szorzata, ekkor a bizonyítás triviális.

1.2.9. Tétel (Euler–Fermat tétel)**.** *Legyen $N > 1$ adott, $a \in \mathbb{Z}_N^*$. Ekkor $a^{\phi(N)} \equiv 1 \pmod N$, speciálisan ha $N = p$ prím, akkor $\forall\, a \in \{1, 2, .., p-1\}$-ra $a^{p-1} \equiv 1 \pmod p$.*

1.2.10. Definíció (Izomorfizmus)**.** *Legyenek G és H csoportok a $\circ_G$ és $\circ_H$ műveletekkel. Ekkor az $f : G \to H$ izomorfizmus, ha:*

1. *f bijekció.*
2. *$\forall\, g_1, g_2 \in G$-re $f(g_1 \circ_G g_2) = f(g_1) \circ_H f(g_2)$.*

Ha létezik ilyen f függvény akkor G és H izomorfak. Jel: $G \simeq H$.

1.2.11. Definíció (Direkt szorzat)**.** *Legyenek G és H véges csoportok, $|G| = n$, $|H| = n'$ renddel és $\circ_G$, $\circ_H$ művelettel. Ekkor G és H direkt szorzatára, melynek jele $G \times H$, a következők teljesülnek:*

1. *$(g, h) \in G \times H$, ahol $g \in G$, $h \in H$.*
2. *$|G \times H| = n \cdot n'$.*
3. *$(g, h) \circ (g', h') = (g \circ_G g', h \circ_H h')$, ahol $\circ$ a $G \times H$ csoportművelete.*

1.2.12. Tétel (Kínai maradéktétel, algebrai alak)**.** *Legyen $N = pq$, ahol $p, q > 1$ relatív prímek. Ekkor*

$$\mathbb{Z}_N \simeq \mathbb{Z}_p \times \mathbb{Z}_q \quad \text{és} \quad \mathbb{Z}_N^* \simeq \mathbb{Z}_p^* \times \mathbb{Z}_q^*.$$

Továbbá legyen az $f : \mathbb{Z}_N \to \mathbb{Z}_p \times \mathbb{Z}_q$ függvény a következő: $f(x) = ([x \mod p], [x \mod q])$. Ekkor f izomorfizmus $\mathbb{Z}_N$ és $\mathbb{Z}_p \times \mathbb{Z}_q$ között, ha pedig az $D_f = \mathbb{Z}_N^*$ és $R_f = \mathbb{Z}_p^* \times \mathbb{Z}_q^*$, akkor f izomorfizmus $\mathbb{Z}_N^*$ és $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$ között.

Bizonyítás. Legyen $x_p = [x \mod p]$ és $x_q = [x \mod q]$, ekkor minden $x \in \mathbb{Z}_N$-re $f(x) = (x_p, x_q)$, ahol $x_p \in \mathbb{Z}_p$ és $x_q \in \mathbb{Z}_q$. Azt állítjuk, hogy ha $x \in \mathbb{Z}_N^*$, akkor $(x_p, x_q) \in \mathbb{Z}_p^* \times \mathbb{Z}_q^*$. Tegyük fel, hogy $x_p \notin \mathbb{Z}_p^*$, ekkor $\text{lnko}([x \mod p], p) \neq 1$, ami azt jelenti, hogy $\text{lnko}(x, p) \neq 1$, ahonnan $\text{lnko}(x, N) \neq 1$ következik ($x_p \notin \mathbb{Z}_p^*$ esetén hasonlóan járunk el).

Most megmutatjuk, hogy f izomorfizmus $\mathbb{Z}_N$ és $\mathbb{Z}_p \times \mathbb{Z}_q$ között. (A bizonyítás hasonló $\mathbb{Z}_N^*$ és $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$ esetén.) Először belátjuk, hogy f bijekció. Tegyük fel, hogy $f(x) = f(x') = (x_p, x_q)$, ekkor $x \equiv x' \equiv x_p \pmod p$ és $x \equiv x' \equiv x_q \pmod q$, ami azt jelenti, hogy $(x - x')$

osztható p-vel és q-val is, ahonnan $N \mid (x - x')$ miatt $x \equiv x' \pmod{N}$ következik, ami $x, x' \in \mathbb{Z}_N$ esetén azt jelenti, hogy $x = x'$. Mivel $|\mathbb{Z}_N| = N = p \cdot q = |\mathbb{Z}_p| \cdot |\mathbb{Z}_q|$ is teljesül, ezért f bijektív.

Jelölje $+_N$ a mod N összeadást és $\circ$ a csoportműveletet a $\mathbb{Z}_p \times \mathbb{Z}_q$ csoportban (mod p összeadás az első tag szerint és mod q összeadás a második tag szerint). Ahhoz, hogy bizonyítsuk, hogy f izomorfizmus azt kell még belátnunk, hogy $\forall\, a, b \in \mathbb{Z}_N$-re $f(a +_N b) = f(a) \circ f(b)$.

$$\begin{aligned} f(a +_N b) &= ([a +_N b \mod p], [a +_N b \mod q]) \\ &= ([a + b \mod p], [a + b \mod q]) \\ &= ([a \mod p], [a \mod q]) \circ ([b \mod p], [b \mod q]) = f(a) \circ f(b), \end{aligned}$$

amivel teljes a bizonyítás. □

A tétel kiterjesztéseként elmondható, hogy ha $p_1, p_2, \ldots, p_l$ páronként relatív prímek és $N = \prod_{i=1}^{l} p_i$, akkor

$$\mathbb{Z}_N \simeq \mathbb{Z}_{p_1} \times \ldots \times \mathbb{Z}_{p_l} \text{ és } \mathbb{Z}_N^* \simeq \mathbb{Z}_{p_1}^* \times \ldots \times \mathbb{Z}_{p_l}^*.$$

Ami a jelölést illeti $x \leftrightarrow (x_p, x_q)$, ahol $x_p = [x \mod p]$ és $x_q = [x \mod q]$ pontosan akkor, ha $f(x) = (x_p, x_q)$. Ez azt jelenti, hogy $x \in \mathbb{Z}_N$ megfeleltethető $(x_p, x_q) \in \mathbb{Z}_p \times \mathbb{Z}_q$-nak.

Azt láttuk, hogy hogyan határozzuk meg x ismerete esetén (x_p, x_q)-t. Meg kell még mutatnunk, hogy (x_p, x_q) ismerete esetén, hogyan kapjuk meg x-et. Vegyük észre, hogy

$$(x_p, x_q) = x_p \cdot (1, 0) + x_q \cdot (0, 1).$$

Meg kell találnunk azokat az $1_p, 1_q \in \mathbb{Z}_N$ elemeket, melyekre $1_p \leftrightarrow (1, 0)$ és $1_q \leftrightarrow (0, 1)$. Mivel p és q relatív prímek, ezért a kiterjesztett Euklideszi algoritmus segítségével meg tudjuk találni azokat az X, Y egészeket, melyekre $Xp + Yq = 1$. Ekkor $1_p = [Yq \mod p]$ és $1_q = [Xp \mod q]$, innen pedig $x = [(1_p x_p + 1_q x_q) \mod N]$.

1.2.13. Lemma. *Legyen G véges csoport, $|G| = m$; $H \subseteq G$ nemüres halmaz. Ha $\forall\, a, b \in H$ esetén $ab \in H$, akkor H a G részcsoportja.*

Bizonyítás. Vizsgáljuk meg, hogy teljesül-e a csoport definíciója H-ra

1. H zárt a csoport műveletre.

2. $a^m = 1 \in H \Rightarrow$ van H-ban neutrális elem.

3. $a \in H \Rightarrow a^{m-1} = a^{-1} \in H \Rightarrow$ minden elemnek van inverze.

4. Az asszocivitást örökli.

Mind a 4 feltétel teljesül, ezért H valóban részcsoport. □

1.2.14. Lemma. *Ha H valós részcsoportja G-nek, akkor $|H| \leq \frac{|G|}{2}$.*

Bizonyítás. Legyen $\overline{h}$ olyan, hogy $\overline{h} \in G$, de $\overline{h} \notin H$, továbbá legyen $\overline{H} := \{\overline{h}h \mid h \in H\}$. Azt fogjuk megmutatni, hogy: 1) $|\overline{H}| = |H|$ és 2) $\overline{H}$ minden eleme H-n kívül van.
1) Legyen $h_1, h_2 \in H$, ekkor ha $\overline{h}h_1 = \overline{h}h_2$, akkor $\overline{h}^{-1}$-gyel való szorzás után $h_1 = h_2$-t kapunk, ami azt jelenti, hogy minden H-beli elem megfeleltethető egy $\overline{H}$-beli elemnek, és ez $\overline{H}$ definíciója miatt fordítva is igaz, így $|\overline{H}| = |H|$ teljesül.
2) Tegyük fel, hogy $\exists\, h$, melyre $\overline{h}h \in H$, ekkor $\overline{h}h = h' \Rightarrow \overline{h} = h'h^{-1} \Rightarrow \overline{h} \in H$ teljesülne, ami ellentmondás. Így $|H| + |H'| = 2|H| \leq |G|$, ezzel az állítás bizonyítást nyert.

□

1.3. Ciklikus csoportok

1.3.1. Definíció (Rend)**.** *Legyen G véges csoport, $g \in G$. Ekkor g rendje a legkisebb i pozitív egész, melyre $g^i = 1$. Jele:* $\mathrm{ord}(g)$.

1.3.2. Állítás. *Legyen G véges csoport, $g \in G$ rendje i. Ekkor $\forall\, x \in \mathbb{Z}$-re $g^x = g^{x \mod i}$.*

A bizonyítás triviálisan következik a definícióból.

1.3.3. Állítás. *Legyen G véges csoport, $g \in G$ rendje i. Ekkor $g^x = g^y$ pontosan akkor, ha $x \equiv y \pmod{i}$.*

Bizonyítás. $\leftarrow$: $x \equiv y \pmod{i} \Rightarrow x \mod i = y \mod i \Rightarrow g^x = g^{x \mod i} = g^{y \mod i} = g^y$.
$\rightarrow$: $g^x = g^y \Rightarrow 1 = g^{x-y} = g^{x-y \mod i} \Rightarrow x - y \mod i < i$, de i a legkisebb pozitív egész, melyre $g^i = 1 \Rightarrow x - y \mod i = 0 \Rightarrow x \equiv y \pmod{i}$. □

1.3.4. Definíció (Ciklikus csoport)**.** *G véges csoport ciklikus, ha $\exists\, g \in G$, melyre a g rendje megyegyezik a G rendjével. Ekkor g-t a G generátorelemének nevezzük.*

1.3.5. Állítás. *Ha a G csoport rendje m, akkor $\forall\, g \in G$-re ha g rendje i, akkor $i \mid m$.*

Bizonyítás. Az 1.2.5-ös állítás alapján $g^m = 1 = g^0$ és az 1.3.3-as állítás szerint ekkor $m \equiv 0 \pmod{i}$. □

1.3.6. Állítás. *Ha G rendje p prím, akkor G ciklikus csoport és az egységelemet leszámítva, minden elem generátorelem.*

Bizonyítás. Az előző állítás szerint minden elem rendje 1 vagy p. Az egységelem rendje 1, a többi elem rendje p. □

1.3.7. Tétel. *Legyen G egy ciklikus csoport $q > 1$ renddel és g generátorelemmel. Ekkor $\phi(q)$ generátoreleme van G-nek és ezek: $\{g^x \mid x \in \mathbb{Z}_q^*\}$.*

Bizonyítás. Ha g generátorelem, akkor $\mathrm{ord}(g) = q$ és $\mathrm{ord}(g^k) = \dfrac{q}{\mathrm{lnko}(q,k)}$. Ez akkor q, ha $\mathrm{lnko}(q,k) = 1$, pontosan $\phi(q)$ ilyen k van. □

1.3.8. Állítás. *G pontosan akkor ciklikus csoport, ha $G \simeq \mathbb{Z}^+$ vagy $G \simeq \mathbb{Z}_N$, valamilyen pozitív N-re.*

Bizonyítás. $\leftarrow$: Ez az irány triviális.
$\rightarrow$: Legyen g generátorelem és legyen $N = \mathrm{ord}(g)$.
Ha $N < \infty$, akkor $f : \mathbb{Z}_N \to G$, $f(x) = g^x$ izomorfizmus.
Ha $N = \infty$, akkor $f : \mathbb{Z}^+ \to G$, $f(x) = g^x$ izomorfizmus. □

1.3.9. Tétel. *Ha p prím, akkor $\mathbb{Z}_p^*$ egy ciklikus, $p-1$ rendű csoport.*

Ez nem az előző állítás következménye, mivel $p > 3$ esetén $p - 1$ nem prím.

2. fejezet

Faktorizáló és diszkrét logaritmus kiszámító algoritmusok

2.1. Faktorizációs algoritmusok

Adott egy N pozitív egész, ami két n bites prím, p és q szorzata. A kérdés az, hogy N ismeretében, hogyan tudjuk megkapni p-t és q-t. Erre a problémára nem ismert $\|N\| = \log_2 N$ szerint polinomiális futásidejű algoritmus, de ebben a fejezetben mutatunk 3 említésre méltó faktorizáló algoritmust. A triviális megoldás a problémára a brute force, azaz végigmenni a számokon 1-től $\sqrt{N}$-ig, és megnézni, hogy osztói-e N-nek. Ennek a futásideje $\mathcal{O}(\sqrt{N}) = \mathcal{O}(2^{\|N\|/2})$, amely exponenciális és megfelelően nagy N esetén esélytelen.

2.1.1. Pollard $p-1$ algoritmusa

Legyen $N = pq$, ahol p és q is n bites. Ez az algoritmus csak akkor működik, ha $p-1$-nek csak "kicsi" prímosztói vannak. Legyen B olyan, hogy $(p-1) \mid B$, de $(q-1) \nmid B$ (B választásáról később). Ekkor $B = c \cdot (p-1)$; valamely $c \in \mathbb{N}^+$-re és legyen $x \in_R \mathbb{Z}_N^*$. A továbbiakban $\in_R$-rel jelölöm, ha egy véletlen elemet veszünk egy halmazból. Legyen $y := [x^B - 1 \mod N]$. Ekkor a Kínai maradéktétel szerint, mivel $1 \leftrightarrow (1,1)$, ezért

$$\begin{aligned} y = [x^B - 1 \mod N] &\leftrightarrow (x_p,\, x_q)^B - (1,\, 1) = ([x_p^B - 1 \mod p],\, [x_q^B - 1 \mod q]) \\ &= \left([\left(x_p^{p-1}\right)^c - 1 \mod p],\, [x_q^B - 1 \mod q]\right) = (0,\, [x_q^B - 1 \mod q]), \end{aligned}$$

felhasználva az Euler–Fermat tételt.

Ezzel találunk egy olyan y-t, melyre $p \mid y$, de $q \nmid y$, innen $\text{lnko}(y, N) = p$.

Algoritmus 2: Pollard $p-1$

Input: N, amely két n bites prím szorzata

Output: N egy prímosztója

```
x ← Z*_N
y := [x^B − 1 mod N]                    // B választása lentebb
p := lnko(y, N)
if p ∉ {1, N} then
    return p
```

2.1.1. Tétel. *Egy $N \geq 3$ n bites egészre* $\dfrac{N}{\phi(N)} < 2n$.

Bizonyítás. Csak két esetben bizonyítunk, ha N prím vagy ha $N = pq$, ahol p és q ugyanolyan hosszú páratlan prímek. Ha N prím, akkor

$$\frac{N}{\phi(N)} \leq \frac{2^n}{\phi(N)} = \frac{2^n}{N-1} < \frac{2^n}{2^{n-1}} = 2.$$

Ha $N = pq$, akkor

$$\frac{N}{\phi(N)} = \frac{p}{p-1} \cdot \frac{q}{q-1} \leq \frac{3}{2} \cdot \frac{5}{4} < 2.$$

□

Vizsgáljuk meg az algoritmus helyességét. Amikor x-et egyenletesen választjuk $\mathbb{Z}_N^*$-ból, akkor x választása $\mathbb{Z}_q^*$-ból is egyenletes. Ha x a $\mathbb{Z}_q^*$ generátoreleme, akkor az 1.3.2-es állítás miatt nem teljesülhet $x_q^B \equiv 1 \pmod{q}$, mivel $|\mathbb{Z}_q^*| = q-1$ és $\mathbb{Z}_q^*$ ciklikus csoport. Továbbá $\mathbb{Z}_q^*$ generátorelemeinek száma $\phi(q-1)$, az 1.3.7-as tétel szerint. Tudnunk kell, hogy mennyi a valószínűsége annak, hogy $x_q \in_R \mathbb{Z}_q^*$ generátorelem.

$$P(x_q \text{ generátorelem}) = \frac{\phi(q-1)}{q-1} > \frac{1}{2n},$$

ami azt jelenti, hogy ha $2n^2$-szer veszünk egy $x_q \in_R \mathbb{Z}_q^*$ elemet, akkor annak az esélye, hogy nem találunk egy generátorelemet sem és ezáltal nem tudunk faktorizálni:

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

Adósok vagyunk még B választásával. Legyen $B := \prod_{i=1}^{k} p_i^{\lfloor n/\log p_i \rfloor}$, ahol p_i jelöli az i. prímet. Ha p_i osztja $p-1$-et akkor $\lfloor n/\log p_i \rfloor$ a p_i legnagyobb hatványa, amely szintén. Ha $p-1 =$

$\prod_{i=1}^{k} p_i^{e_i}$, ahol $e_i = \lfloor n/\log p_i \rfloor$ $\forall$ i-re akkor $q-1$-nek van p_k-nál nagyobb prímosztója, mivel p és q is n bitesek, így $(p-1) \mid B$, valamint $(q-1) \nmid B$. Ha $p-1$-nek csak kicsi prímosztói vannak, akkor könnyű faktorizálni, ezért tehetjük fel, hogy ismerjük $p-1$ prím tényezős felbontását.

Nagyobb k választása növeli a futásidőt, $(p-1) \mid B$ esélyét, de $(q-1) \mid B$ esélyét is. A futásidő n szerint polinomiális, mivel minden lépés polinomiális és az algoritmust $2n^2$-szer futtattuk le. Ezen algoritmus miatt érdemes a kriptográfiában az erős prímeket használni.

2.1.2. Definíció. *Egy p prím erős, ha $\frac{p-1}{2}$ is prím.*

2.1.2. Pollard rho algoritmusa

Pollard rho algoritmusának a futásideje $\mathcal{O}(N^{\frac{1}{4}})$, ami már lényegesebb jobb, mint a brute force, de még mindig nem túl hatékony. Alapötlete az az, hogy ha találunk két $x, x' \in \mathbb{Z}_N^*$ elemet, melyekre $x \equiv x' \pmod{p}$, (ezket jó pároknak fogjuk hívni), akkor $\text{lnko}(x-x', N) = p$, persze csak ha $x \not\equiv x' \pmod{N}$.

Hogyan találunk jó párt? Vegyük az $x^{(1)}, \ldots, x^{(k)}$ random elemeket $\mathbb{Z}_N^*$-ból, ahol $k = 2^{n/2} = \mathcal{O}(\sqrt{p})$. A Kínai maradéktétel szerint $x^{(i)}$ megfeleltethető $(x_p^{(i)}, x_q^{(i)})$-nek, ahol $x_p^{(i)} = [x^{(i)} \mod p]$ és $x_q^{(i)} = [x^{(i)} \mod q]$.

Algoritmus 3: Pollard rho

Input: $N \in \mathbb{Z}_N^*$, két n bites prím szorzata

Output: N egy prímosztója

```
x^(0) ∈_R Z*_N;  x' := x := x^(0)
for i = 1 ... 2^{n/2} do
    x := F(x)
    x' := F(F(x'))
    p := lnko(x − x', N)
    if p ∉ {1, N} then
        return p
```

Kérdés az, hogy mi az esélye annak, hogy az $x^{(1)}, \ldots, x^{(k)}$ elemek között van jó pár. Erre ad választ a következő lemma.

2.1.3. Lemma. *Legyen N egy pozitív egész és $q \leq \sqrt{2N}$. Ha az $y_1, \ldots y_q$ elemeket véletlen választjuk egy N elemű halmazból,akkor annak az esélye, hogy az elemek között van két azonos, legalább* $\dfrac{q(q-1)}{4N}$.

Bizonyítás. Coll jelölje azt az eseményt ha van két azonos elem, $\mathrm{NoColl_i}$ pedig azt az eseményt, ha nincs két azonos $y_1, y_2, \ldots, y_i$ között. Nyilván $\overline{\mathrm{Coll}} = \mathrm{NoColl_q}$. Ekkor

$$P(\mathrm{NoColl_q}) = P(\mathrm{NoColl_1}) \cdot P(\mathrm{NoColl_2} \mid \mathrm{NoColl_1}) \cdot \ldots \cdot P(\mathrm{NoColl_q} \mid \mathrm{NoColl_{q-1}}),$$

valamint $P(\mathrm{NoColl_1}) = 1$, mivel ekkor csak 1 elem van. Továbbá ha $\mathrm{NoColl_i}$ már teljesült, azaz az első i elem között nincs ütközés akkor annak az esélye, hogy a következő elemet hozzávéve se lesz két azonos az $1 - \dfrac{i}{N} = P(\mathrm{NoColl_{i+1}} \mid \mathrm{NoColl_i})$, ezért

$$P(\mathrm{NoColl_q}) = \prod_{i=1}^{q-1} \left(1 - \frac{i}{N}\right) \leq \prod_{i=1}^{q-1} e^{-i/N} = e^{-\sum_{i=1}^{q-1}(i/N)} = e^{-q(q-1)/2N}.$$

Ezért $P(\mathrm{Coll}) = 1 - P(\mathrm{NoColl_q}) \geq 1 - e^{-q(q-1)/2N} \geq \dfrac{q(q-1)}{4N}$. □

Jelen esetben ez azt jelenti, hogy az ütközés esélye alulról becsülhető kb. $\dfrac{1}{4}$-del. Visszatérve, mind a k elemre az egyenlőség letesztelése $\binom{k}{2}$ összehasonlítás, ami $\mathcal{O}(N^{1/2})$ lépés, ez túl sok. Az ötlet az, hogy konstruáljunk egy $F : \mathrm{Z}_N^* \to \mathrm{Z}_N^*$ függvényt amire:

- $x^{(i)} := F(x^{(i-1)})$,
- $x \equiv x' \pmod{p} \Rightarrow F(x) \equiv F(x') \pmod{p}$.

Erre a standard választás $F(x) = [x^2 + 1 \mod N]$.

2.1.3. Négyzetes szitálási algoritmus

A négyzetes szitálási algoritmus volt az 1990-es évekig a leggyorsabb faktorizáló algoritmus, futásideje szubexponenciális és egészen 300 bites számokig használható a gyakorlatban faktroizálásra. Az algoritmus ötlete a következő: keressünk olyan x-et és y-t, hogy $x^2 \equiv y^2 \pmod{N}$, de $x \not\equiv \pm y \pmod{N}$, mert ekkor $0 \equiv x^2 - y^2 \pmod{N} \Rightarrow N \mid (x-y)(x+y)$, de $N \nmid (x-y)$, $N \nmid (x+y) \Rightarrow \mathrm{lnko}(x-y, N)$ egy prímosztó.

2.1.4. Definíció. *Az $y \in \mathbb{Z}_N^*$ kvadratikus maradék* $\bmod N$*, ha $\exists\, x \in \mathbb{Z}_N^*$,amire $x^2 \equiv y \pmod{N}$.*

2.1.5. Állítás. *Legyen $N = pq$, ahol p és q különböző prímek és $y \in \mathbb{Z}_N^*, y \leftrightarrow (y_p, y_q)$. Az y pontosan akkor kvadratikus maradék* mod N *ha y_p kvadratikus maradék* mod p *és y_q kvadratikus maradék* mod q.

Bizonyítás. Ha y kvadratikus maradék mod N, akkor létezik x, hogy $x^2 \equiv y \bmod \mathrm{N}$ és $x \leftrightarrow (x_p, x_q)$. Ekkor

$$(y_p, y_q) \leftrightarrow y \equiv x^2 \pmod{N} \leftrightarrow (x_p, x_q)^2 = (x_p^2 \bmod p, x_q^2 \bmod q), \tag{2.1}$$

ahonnan azt kapjuk, hogy $y_p = [x_p^2 \bmod p]$ és $y_q = [x_q^2 \bmod q]$.

A 2.1-es egyenlet során ekvivalens átalakításokat hajtottunk végre, így ha abból indulunk ki, hogy y_p és y_q kvadratikus maradékok akkor visszafelé haladva eljutunk addig, hogy y is kvadratikus maradék. □

Így ekkor y-nak 4 db négyzetgyöke van: (x_p, x_q), $(-x_p, x_q)$, $(x_p, -x_q)$, $(-x_p, -x_q)$. Kérdés az, hogy hogyan találnuk megfelelő x-et és y-t. Ha random választanánk őket akkor exponenciális sok futtatásra lenne szükség, helyette az alábbi módszert követjük.

Legyen B rögzített. Egy egész szám B-sima, ha minden prímosztója $\leq B$. Először keresünk B-sima számokat majd faktorzáljuk őket, ami megfelelő B esetén nem nehéz. A számokat $q_i = [x_i^2 \bmod \mathrm{N}]$ alakban keressük, ahol $x = \lfloor\sqrt{N}\rfloor + 1, \lfloor\sqrt{N}\rfloor + 2 \ldots$ Így $q = [x^2 \bmod \mathrm{N}]$ kicsi lesz és nagyobb eséllyel B-sima. Legyenek $p_1, \ldots, p_k$ a B-nél nem nagyobb prímek. Ekkor a $\{q_1, \ldots, q_l\}$ halmaz elemeire fennállnak:

$$\begin{aligned} q_1 &= [x_1^2 \bmod N] = \prod_{i=1}^{k} p_i^{e_{1,i}} \\ q_2 &= [x_2^2 \bmod N] = \prod_{i=1}^{k} p_i^{e_{2,i}} \\ &\vdots \\ q_l &= [x_l^2 \bmod N] = \prod_{i=1}^{k} p_i^{e_{l,i}}. \end{aligned}$$

Szükség van a $\{q_1, \ldots, q_l\}$ halmaznak egy olyan részhalmazára, ahol a részhalmaz elemeinek szorzata négyzetszám. Legyen S egy részhalmaz, ekkor

$$z = \prod_{j \in S} q_j = \prod_{i=1}^{k} p_i^{\sum_{j \in S} e_{j,i}},$$

ami akkor négyzetszám, ha a $\sum_{j\in S} e_{j,i}$ kitevő páros $\forall$ i-re. Készítsünk egy A mátrixot, ahol $a_{i,j} = [e_{i,j} \mod 2]$.

$$A = \begin{bmatrix} a_{1,1} & \dots & a_{1,k} \\ \vdots & \ddots & \vdots \\ a_{l,1} & \dots & a_{l,k} \end{bmatrix} = \begin{bmatrix} [e_{1,1} \mod 2] & \dots & [e_{1,k} \mod 2] \\ \vdots & \ddots & \vdots \\ [e_{l,1} \mod 2] & \dots & [e_{l,k} \mod 2] \end{bmatrix}.$$

Az A mátrixnak l sora és k oszlopa van. Ha $l > k$ akkor A-nak több sora van, mint oszlopa, így a mátrix rangja legfeljebb k, ami azt jelenti, hogy a sorok lineárisan összefüggenek mod 2. Így a lineárisan összefüggő soroknak megfelelő $\{q_1, .., q_l\}$-beli elemek által alkotott részcsoport megfelelő lesz, hiszen abban az elemek szorzata négyzetszám.

Tehát egyfelől

$$z = \prod_{j\in S} q_j = \prod_{i=1}^{k} p_i^{\sum_{j\in S} e_{j,i}} = \left(\prod_{i=1}^{k} p_i^{(\sum_{j\in S} e_{j,i})/2}\right)^2,$$

másfelől

$$z = \prod_{j\in S} q_j = \prod_{j\in S} [x_j^2 \mod N] \equiv \left(\prod_{j\in S} x_j\right)^2 \pmod N.$$

Így találtunk megfelelő x, y párt, ha $x \not\equiv \pm y \pmod N$, akkor kapunk egy prímosztót. Nagyobb B választása esetén nagyon eséllyel találunk B-sima q-kat, de a faktorizációjuk nehezebb és több B-sima q kell, hogy $l > k$ teljesüljön, valamint akkor az A mátrix is nagyobb és a lineáris algebrás rész is lassabb. Megfelelő B választása mellett az algoritmus futásideje $2^{\mathcal{O}\left(\sqrt{\log N \log(\log N)}\right)}$.

2.1.6. Példa. $N = 377753$; $B = 29$; $\{q_i\}$ *releváns elemei:*

$$\begin{aligned} q_1 &= [620^2 \mod N] = 17^2 \cdot 23; \quad x_1 = 620 \\ q_2 &= [621^2 \mod N] = 2^4 \cdot 17 \cdot 29; \quad x_2 = 621 \\ q_3 &= [645^2 \mod N] = 2^7 \cdot 13 \cdot 23; \quad x_3 = 645 \\ q_4 &= [655^2 \mod N] = 2^3 \cdot 13 \cdot 17 \cdot 29; \quad x_4 = 655. \end{aligned}$$

Továbbá $q_1 \cdot q_2 \cdot q_3 \cdot q_4 = 2^{14} \cdot 13^2 \cdot 17^4 \cdot 23^2 \cdot 29^2$, *ami négyzetszám, így* $q_1q_2q_3q_4 \equiv (x_1x_2x_3x_4)^2 \pmod N \Rightarrow 45335^2 \equiv 127194^2 \pmod N$. *Végül* $\text{lnko}(127194 - 45335, 377753) = 751 \mid N$.

2.2. Diszkrét logaritmus kiszámító algoritmusok

2.2.1. Definíció (Diszkrét logaritmus)**.** *Legyen G ciklikus csoport q renddel és g generátorelemmel. Ekkor $G := \{g^0, g^1, \ldots, g^{q-1}\}$ és $\forall\, h \in G$-hez $\exists!\, x$, melyre $0 \leq x \leq q-1$, és $g^x = h$. Ekkor x-et a h g alapú diszkrét logaritmusának nevezzk. Jelölés: $x = \log_g h$. Az, hogy mi a G csoport a kontextusból derül ki.*

A diszkrét logaritmus tulajdonságai:

- $g^x = h \Leftrightarrow x \mod q = \log_g h$,
- $\log_g 1 = 0$,
- $\log_g(h^r) \equiv r \cdot \log_g h \pmod q$,
- $\log_g(h_1 h_2) \equiv \log_g h_1 + \log_g h_2, \pmod q$.

A következő algoritmusok célja mind ugyanaz, adott G csoport esetén, melynek rendje q és egy eleme g, valamint $h \in \langle g \rangle$ esetén, megkeresni azt az x-et, melyre $g^x = h$, tehát g-nek nem kell feltétlenül generátorelemnek lenni. A triviális brute force algoritmus futásideje $|\langle g \rangle|$, ennél fogunk jobb algoritmusokat mutatni.

2.2.1. Pohlig–Hellman algoritmus

Ez az algoritmus akkor használható, ha ismert q-nak egy nem triviális osztója.

2.2.2. Lemma. *Legyen* $\operatorname{ord}(g) = q$ *és* $p \mid q$*, ekkor* $\operatorname{ord}(g^p) = \dfrac{q}{p}$.

Bizonyítás. Mivel $(g^p)^{q/p} = g^q = 1$, ezért $\operatorname{ord}(g^p) \leq \dfrac{q}{p}$. Ha $(g^p)^i = 1$, akkor mivel a g rendje q, ezért $pi \geq q \Rightarrow i \geq \dfrac{q}{p}$. Így $\operatorname{ord}(g^p) = \dfrac{q}{p}$. □

Tegyük fel, hogy q egy felbontása ismert, azaz $q = \prod_{i=1}^{k} q_i$, ahol $q_1, q_2, \ldots, q_k$ páronként relatív prímek, de ez nem feltétlenül q prím tényezős felbontása. Ekkor

$$(g^{q/q_i})^x = (g^x)^{q/q_i} = h^{q/q_i} \quad i = 1, \ldots, k,$$

valamint a Kínai maradéktétel szerint

$$\mathbb{Z}_q \simeq \mathbb{Z}_{q_1} \times \ldots \times \mathbb{Z}_{q_k} \quad \text{és} \quad \mathbb{Z}_q^* \simeq \mathbb{Z}_{q_1}^* \times \ldots \times \mathbb{Z}_{q_k}^*.$$

Legyen $g_i = g^{q/q_i}$ és $h_i = h^{q/q_i}$. A problémát innentől k kisebb részcsoportban kell megoldani: $g_i^x = h_i;\ \ i = 1, \ldots, k$. Ezekben a csoportokban diszkrét logaritmus probléma gyorsabban megoldható. A megoldások: $g_i^{x_i} = h_i,\ i = 1, \ldots, k$. Ez a 1.3.3-as állítás szerint pontosan akkor egyenlő g^x-nel, ha $x \equiv x_i \mod q_i\ i = 1, \ldots, k$, azaz ha a következő k egyenlet teljesül x-re:

$$\begin{aligned} x &\equiv x_1 \mod q_1 \\ x &\equiv x_2 \mod q_2 \\ &\vdots \\ x &\equiv x_k \mod q_k. \end{aligned}$$

Ez a Kínai maradéktétel szerint megoldható és a megoldás egyetlen maradékrendszer mod q, ez a keresett x.

2.2.2. Baby-Step/Giant-Step algoritmus

Legyen $g \in G$ generátorelem, $|G| = q$. Ekkor g hatványai a következőképp néznek ki:

$$1 = g^0,\ g^1,\ g^2,\ \ldots,\ g^{q-2},\ g^{q-1},\ g^q = 1.$$

Tudjuk, hogy h valahol ezen a "körön" helyezkedik el. Legyen $t := \lfloor \sqrt{q} \rfloor$, jelöljünk ki legfeljebb t hosszú intervallumokat ezen a körön, azaz számoljuk ki a következő $\lfloor q/t \rfloor + 1 = \mathcal{O}(\sqrt{q})$ értéket: $g^0,\ g^t,\ g^{2t}, \ldots,\ g^{\lfloor q/t \rfloor \cdot t}$. Ekkor a körön bármely 2 kijelölt pont között a távolság legfeljebb t, valamint h is valamely 2 szomszédos kijelölt pont között van, ha nem azt azt jelenti, hogy már meg is találtuk. Számoljuk ki a következő értékeket: $g^1 \cdot h,\ g^2 \cdot h,\ \ldots,\ g^t \cdot h$. Ezek szintén egymást követő elemek lesznek a körön csak valamennyivel eltolva. Tehát van t db szomszédos elemünk valahol a körön. Ezek közül 1 biztosan megegyezik valamelyik kijelölt ponttal, hiszen 2 kijelölt pont távolsága legfeljebb t. Tegyük fel, hogy $h \cdot g^i = g^{kt}$, innen $\log_g h = [kt - i \mod q]$.

Ami az algoritmus futásidejét illeti, a g_i-k kiszámítása $\mathcal{O}(\sqrt{q})$(lásd az algoritmus pszeudokódját); a rendezés $\mathcal{O}(\sqrt{q} \log q)$; eldönteni, hogy van-e olyan g_k, hogy $h_i = g_k$, minden i-re bináris kereséssel pedig $\mathcal{O}(\log q)$. Összesen a futásidő $\mathcal{O}(\sqrt{q} \cdot \text{polylog}(q))$.

Algoritmus 4: Baby-Step/Giant-Step

Input: $g, h \in G$; g generátorelem; $q = |G|$

Output: $\log_g h$

```
1 t := ⌊√q⌋
2 for i = 0 ... ⌊q/t⌋ do
3     g_i := g^{it} kiszámítása
4 (i, g_i) rendezése a 2. tag szerint
5 for i = 1 do
6     h_i := h · g^i
7     if h_i = g_k valamely k-ra then
8         return kt − i  mod q
```

2.2.3. Az index kalkulus módszer

Ez az algoritmus adott $g, h \in \mathbb{Z}_N^*$ elemek esetén, ahol g generátorelem, kiszámítja azt az $x \in \mathbb{Z}_p^*$-t, amelyre $g^x \equiv h \mod N$, futásideje szubexponenciális. Legyen B fix és legyenek $p_1, \ldots, p_k$ azon prímek melyek nem nagyobbak B-nél. Először keressünk $l \geq k$ különböző $x_1, \ldots, x_l \in \mathbb{Z}_{p-1}$ értéket, melyekre $g_i = [g^{x_i} \mod p]$ B-sima. Ezt véletlen $\mathbb{Z}_{p-1}$-beli elemek választásával tesszük. Faktorizálva a kapott g_i elemeket kapjuk:

$$[g^{x_1} \mod p] = \prod_{i=1}^{k} p_i^{e_{1,i}}$$

$$\vdots$$

$$[g^{x_l} \mod p] = \prod_{i=1}^{k} p_i^{e_{l,i}}.$$

Diszkrét logaritmust véve kapjuk a következő lineáris egyenletrendszert:

$$
\begin{aligned}
x_1 &= \left[\sum_{i=1}^{k} e_{1,i} \cdot \log_g p_i \mod p-1 \right] \\
&\vdots \\
x_l &= \left[\sum_{i=1}^{k} e_{l,i} \cdot \log_g p_i \mod p-1 \right],
\end{aligned}
$$

ahol csak a $\log_g p_i$-k ismeretlenek. Ha adott h-ra ki akarjuk számítani $\log_g h$-t akkor vegyünk olyan $x \in_R \mathbb{Z}_{p-1}$-t, melyre $[g^x \cdot h \mod p]$ B-sima. Ekkor kapjuk a következő 2 egyenletet:

$$
[g^x \cdot h \mod p] = \prod_{i=1}^{k} p_i^{e_i} \implies x + log_g h \equiv \sum_{i=1}^{k} e_i \cdot \log_g p_i \pmod{p-1},
$$

ahol x és az e_i-k ismertek. Hozzáadva ezt az előző egyenletrendszerhez kapunk egy $l+1$ egyenletből álló, $k+1$ ismeretlent tartalmazó egyenletrendszert, ahol $l \geq k$. Lineáris algebrai módszerekkel ez megolható, amivel megkapjuk $\log_g h$-t.

Ami a futásidőt illeti, nagyobb B választása esetén egy random $\mathbb{Z}_N^*$-beli elem nagyobb eséllyel lesz B sima, de a faktorizációjuk nehezebb, többet is kell keresni belőlük, valamint az egyenletrendszer megoldása is hasszasabb. Optimális B választása esetén a futásidő $\mathbb{Z}_p^*$-ben $2^{\mathcal{O}(\sqrt{\log p \cdot \log\log p})}$.

3. fejezet

Prím generálás és ciklikus csoportok generálása

3.1. Prím generálás és prímtesztelés

Szeretnénk n bites prímszámokat véletlenszerűen generálni. Ezt úgy fogjuk tenni, hogy random generálunk egy n bites számot majd megnézzük, hogy az prím-e. ha prím akkor készen is vagyunk ha nem, akkor újrapróbáljuk. Ehhez tudnunk kell mennyi a prímek aránya az n bites számok között.

3.1.1. Tétel (Prímszámtétel)**.** *Jelölje $\pi(x)$ az 1-től x-ig terjedő prímek számát. Ekkor*

$$\lim_{x\to\infty} \frac{\pi(x)}{\frac{x}{\log x}} = 1 \qquad \Longrightarrow \qquad \pi(x) \sim \frac{x}{\log x}.$$

Ebből könnyen következik az alábbi tétel:

3.1.2. Tétel (Bertrand-féle posztolátum)**.** *Annak az esélye, hogy egy n bites szám prím $\forall n$-re legalább $\dfrac{1}{3n}$.*

Ez azt jelenti, hogy ha választunk $3n^2$ db random n bites számot, akkor annak az esélye, hogy nincs közte prím legfeljebb:

$$\left(1-\frac{1}{3n}\right)^{3n^2} = \left(\left(1-\frac{1}{3n}\right)^{3n}\right)^{n} \leq \left(e^{-1}\right)^{n} = e^{-n}.$$

3.1.3. Tétel. *A Miller–Rabin prímteszt futásideje $\|p\|$ és t szerint polinomiális, ahol p prím, t pedig biztonsági paraméter. Továbbá ha p prím azt a teszt helyesen megállapítja, ha pedig p összetett akkor a hiba esélye legfeljebb 2^{-t}.*

Algoritmus 5: Random prím generálása

Input: n

Output: random n bites prím

```
for i = 1...3n^2 do
    p' := {0,1}^{n-1}
    p := 1||p'
    Miller–Rabin prímteszt futtatása p inputtal és 1^n biztonsági paraméterrel
    if A Miller–Rabin prímteszt outputja "prím" then
        return p
return "fail"
```

A tétel bizonyítása meglehetősen hosszas, a következőkben az ehhez szügséges állítások, definíciók következnek.

Ha N prím, akkor $|\mathbb{Z}_N^*| = N-1$ és az Euler–Fermat tétel szerint $a^{N-1} \equiv 1 \pmod{N}$ $\forall$ $a \in \{1, \ldots, N-1\}$ esetén. Tehát ha egy $N \in \mathbb{N}^+$-hoz találunk egy olyan a-t, melyre $a^{N-1} \not\equiv 1 \pmod{N}$, akkor N biztosan nem prím.

3.1.4. Definíció. *Ha $a \in \{1, \ldots, N-1\}$ olyan, hogy $a^{N-1} \not\equiv 1 \pmod{N}$, akkor a tanúsítja, hogy N összetett, ezért a-t tanúnak nevezzük.*

3.1.5. Állítás. *Ha $\exists$ a, ami tanúsítja, hogy N összetett, akkor $\mathbb{Z}_N^*$ elemeinek legalább a fele is tanú.*

Bizonyítás. Definíció szerint legyen Bad $:= \{a \mid a \in \mathbb{Z}_N^* \wedge a^{N-1} \equiv 1 \pmod{N}\}$. Ekkor $1 \in$ Bad, valamint $a, b \in$ Bad $\Rightarrow (ab)^{N-1} \equiv a^{N-1}b^{N-1} \equiv 1 \cdot 1 \pmod{N} \Rightarrow ab \in$ Bad. Ekkor az 1.2.13-as lemma szerint Bad a $\mathbb{Z}_N^*$ részcsoportja, az 1.2.14-es lemma szerint pedig $|\text{Bad}| \leq \dfrac{|\mathbb{Z}_N^*|}{2}$. □

Tehát ha van tanú, akkor akkor $\mathbb{Z}_N^*$ elemeinek legalább a fele tanú, ezért t db véletlen elemet választva a hiba esélye 2^{-t} alatt lenne.

Sajnos léteznek olyan összetett számok, melyeknek nincs tanúja, ezeket Carmichael számoknak nevezzük, ilyen szám pl. az 561. Ezért erősebb definícióra lesz szükségünk.

3.1.6. Definíció. *Legyen $N-1 = 2^r u$, ahol $r \geq 1$; u páratlan. Az $a \in \mathbb{Z}_N^*$ erős tanú, ha $a^u \not\equiv \pm 1 \pmod{N}$ és $a^{2^i u} \not\equiv -1 \pmod{N}$ $\forall$ $i \in \{1, \ldots, r-1\}$ esetén.*

Ha a nem erős tanú, akkor $(a^u, a^{2u}, \ldots, a^{2^r u})$ a következőképp nézhet ki:

$$(\pm 1, 1, \ldots, 1) \ \textit{vagy} \ (*, *, \ldots, *, -1, 1, \ldots, 1).$$

Tehát ha a nem erős tanú akkor $a^{2^{r-1}u} \equiv \pm 1 \pmod{N}$ és $a^{N-1} = a^{2^r u} \equiv 1 \pmod{N}$. Ebből következőleg, ha a nem erős tanú, akkor nem is tanú, így több az erős tanú, mint a tanú. Most azt fogjuk belátni, hogy ha N prím akkor nincs erős tanúja.

3.1.7. Lemma. *Azt mondjuk, hogy $x \in \mathbb{Z}_N^*$ négyzetgyöke 1-nek $\bmod N$, ha $x^2 \equiv 1 \pmod{N}$. Ha N páratlan prím, akkor 1 négyzetgyöke $\bmod N$ az 1, vagy -1.*

Bizonyítás. Legyen $x^2 \equiv 1 \pmod{N}$; $x \in \{1, 2, \ldots, N-1\}$. Ekkor

$$x^2 - 1 \equiv (x+1)(x-1) \equiv 0 \pmod{N} \Rightarrow N \mid (x+1)(x-1) \Rightarrow N \mid (x+1) \vee N \mid (x-1).$$

Ha $N \mid (x+1)$, akkor $x \equiv -1 \pmod{N}$, ha $N \mid (x-1)$, akkor $x \equiv 1 \pmod{N}$. □

Legyen N páratlan prím, $a \in \mathbb{Z}_N^*$ tetszőleges. Legyen k a minimális olyan nemnegatív egész, melyre $a^{2^k u} \equiv 1 \pmod{N}$. Mivel $a^{2^r u} \equiv a^{N-1} \equiv 1 \pmod{N}$, ezért $\exists$ ilyen $k \leq r$. Ha $k = 0$, akkor $a^u \equiv 1 \pmod{N}$, így a nem erős tanú.

Különben $\left(a^{2^{k-1}u}\right)^2 \equiv a^{2^k u} \equiv 1 \pmod{N}$, így $a^{2^{k-1}u}$ négyzetgyöke 1-nek $\bmod N$, de k választása miatt csak $a^{2^{k-1}u} \equiv -1 \pmod{N}$ lehetséges, így a nem erős tanú. Tehát ha N páratlan prím akkor nincs erős tanúja.

Most megmutatjuk, hogy ha N páratlan, összetett és nem teljes hatvány ($N \neq p^r$, ahol p-nek nem is kell jelen esetben prímnek lenni), akkor több erős tanúja is van.

3.1.8. Tétel. *Legyen N páratlan, összetett és nem teljes hatvány, ekkor $\mathbb{Z}_N^*$ elemeinek több, mint fele erős tanú.*

Bizonyítás. Először definiálunk két halmazt. Legyen $\text{Bad} \subseteq \mathbb{Z}_N^*$ azon elemek halmaza, melyek nem erős tanúk. Legyen $i \in \{0, 1, \ldots, r-1\}$ a legnagyobb olyan, amelyhez létezik olyan a nem erős tanú, hogy $a^{2^i u} \equiv -1 \pmod{N}$. Mivel $-1^{2^0 u} \equiv -1 \pmod{N}$, ezért létezik ilyen i. A Bad' halmazt a következőképp definiáljuk: $\text{Bad}' := \{a \mid a^{2^i u} \equiv \pm 1 \pmod{N}\}$.

Azt fogjuk belátni, hogy 1) : a Bad részhalmaza Bad′-nek és 2) : Bad′ valós részcsoportja $\mathbb{Z}_N^*$-nak. Ha ezek teljesülnek akkor a az 1.2.14-es lemma miatt a tétel bizonyítást nyer.

1) $\text{Bad} \subseteq \text{Bad}'$: Ha $a \in \text{Bad}$, akkor $a^u \equiv 1 \pmod{N}$ vagy $\exists\, j : a^{2^j u} \equiv -1 \pmod{N}$ $j \in$

$\{0, \ldots, r-1\}$. Ha $a^u \equiv 1 \pmod N \Rightarrow a^{2^i u} = 1 \pmod N \Rightarrow a \in \text{Bad}'$, ha valamely $j \leq i$-re $a^{2^j u} \equiv -1 \pmod N \Rightarrow a^{2^i u} \equiv \pm 1 \pmod N \Rightarrow a \in \text{Bad}'$.
2) $\text{Bad}' < \mathbb{Z}_N^* : 1 \in \text{Bad}'$; $a, b \in \text{Bad}' \Rightarrow (ab)^{2^i u} \equiv a^{2^i u} b^{2^i u} \equiv (\pm 1)(\pm 1) \equiv \pm 1 \pmod N \Rightarrow ab \in \text{Bad}'$. Ekkor a 1.2.13-as lemma szerint $\text{Bad}' \leq \mathbb{Z}_N^*$.

Azt már beláttuk, hogy Bad′ részcsoportja $\mathbb{Z}_N^*$-nak, ahhoz, hogy belássuk, hogy valós részcsoportja találnunk kell egy elemet $\mathbb{Z}_N^*$-ból, ami nincs benne Bad′-ben.

Legyen $N = N_1 N_2$, ahol $\text{lnko}(N_1, N_2) = 1$; $N_1, N_2 > 1$ páratlan számok. Ekkor a Kínai maradéktétel szerint $\mathbb{Z}_N^* \simeq \mathbb{Z}_{N_1}^* \times \mathbb{Z}_{N_2}^*$ és $a \in \mathbb{Z}_N^*$ megfeleltethető $(a_1, a_2) \in \mathbb{Z}_{N_1}^* \times \mathbb{Z}_{N_2}^*$-nek. Mivel $-1 \leftrightarrow (-1, -1)$, ezért az i definíciója szerint $a^{2^i u} \mod N = -1 \leftrightarrow (-1, -1) = (a_1^{2^i u} \mod N_1, a_2^{2^i u} \mod N_2)$. Legyen $b \in \mathbb{Z}_N^*$ olyan, hogy $b \leftrightarrow (a_1, 1)$. Ekkor $b^{2^i u} \leftrightarrow (a_1^{2^i u} \mod N_1, 1 \mod N_2) = (-1, 1) \not\leftrightarrow \pm 1 \Rightarrow (a_1, 1) \leftrightarrow b \notin \text{Bad}'$.

Találtunk olyan b elemet ami benne van $\mathbb{Z}_N^*$-ban, de nincs benne Bad′-ban, ezzel a 2) állítást is igazoltuk. Így $|\text{Bad}| \leq |\text{Bad}'| \leq \frac{|\mathbb{Z}_N^*|}{2}$, amivel beláttuk a tételt.

□

Algoritmus 6: Miller–Rabin prímteszt

Input: $N > 2 \in \mathbb{Z}$ és 1^t paraméter

Output: Döntés n-ről, hogy prím-e

```
1  if N páros then
2      return "összetett"
3  if N teljes hatvány then
4      return "összetett"
5  N − 1 := 2^r u, ahol u páratlan és r ≥ 1
6  for j = 1 do
7      a ← {1, −1}
8      if a^u ≢ ±1 (mod N) then
9          for i = 1 − 1 do
10             if a^{2^i u} ≢ −1 (mod N) then
11                 return "összetett"
12 return "prím"
```

Ezzel minden adott, hogy a 3.1.3-as tételt bizonyítsuk.

Bizonyítás. Ha N páratlan prím, akkor nincs erős tanúja és a Miller–Rabin teszt "prím" választ fog adni. Ha N páros vagy teljes hatvány akkor a teszt automatikusan "összetett" választ ad. Ezek közül mindkettő eldönthető polinomiális időben könnyedén. Az érdekes eset amikor N páratlan, összetett nem teljes hatvány. Ekkor annak az esálye, hogy egy véletlen $a \in \{1, \ldots, N-1\}$ erős tanú vagy egy nem $\mathbb{Z}_N^*$-beli elem (ami szintén igazolja, hogy N összetett) legalább $\frac{1}{2}$. Tehát ha t-szer választunk véletlen elemet az $\{1, \ldots, N-1\}$ halmazból, akkor annak az esélye, hogy mind a t alkalommal olyan a elemet választunk ami nem igazolja, hogy N öszetett, miközben N mégis az, és így az algoritmus tévesen ír "összetett" választ legfeljebb 2^{-t}. □

3.2. Ciklikus csoportok generálása

Adott egy q prím, mi pedig szeretnénk egy q rendű ciklikus csoportot generálni. Ez azért nem triviális, mert $\mathbb{Z}_p^*$ rendje $p-1$, ami nem prím. A prímrendű ciklikus csoportokra a 4.5-ös alfejezetben tárgyalt Diffie–Hellman problémák miatt van szükség.

3.2.1. Tétel. *Legyen $p = rq + 1$, ahol p, q prímek. Ekkor $G := \{[h^r \mod p] \mid h \in \mathbb{Z}_p^*\}$ egy q rendű ciklikus részcsoportja $\mathbb{Z}_p^*$-nak.*

Bizonyítás. Az, hogy G részcsoport az nyilvánvaló. A továbbiakban azt fogjuk belátni, hogy az $f_r : \mathbb{Z}_p^* \to G$, $f_r(g) = [g^r \mod p]$ függvény minden értéket r alkalommal vesz fel. Ebből $\dfrac{|\mathbb{Z}_p^*|}{r} = \dfrac{p-1}{r} = q$ miatt következik a tétel. Legyen $\mathbb{Z}_p^*$ egy generátoreleme g. Ekkor $\mathbb{Z}_p^* := \{g^0, g^1, \ldots, g^{p-2}\}$. Az 1.3.3-as állítás szerint

$$(g^i)^r = (g^j)^r \Leftrightarrow ir \equiv jr \pmod{p-1} \Leftrightarrow p-1 \mid (i-j)r \Leftrightarrow q \mid i-j.$$

Ez azt jelenti, hogy minden fix $j \in \{0, 1, \ldots, p-2\}$ esetén azok az $i \in \{0, 1, \ldots, p-2\}$ elemek, melyekre $(g^i)^r = (g^j)^r$, azok a következők: j, $j+q$, $j+2q, \ldots, j+(p-1)q$, ahol minden értéket $\mod p-1$ kell venni. Ez összesen r ilyen elem és mivel ez $\forall\ j \in \mathbb{Z}_p^*$-re igaz, így az állítás igazolást nyert. □

A tételből adódóan az is igaz, hogy könnyen tudunk generálni véletlen G-beli elemet, valamint egy $\mathbb{Z}_p^*$-beli véletlen elemről könnyen el tudjuk dönteni, hogy G-beli-e. Véletlen G-beli elemet úgy kapunk, hogy veszünk egy $x \in_R \mathbb{Z}_p^*$-ot és kiszámoljuk $x^r \mod p$-t. Mivel G prímrendű, ezért az egységelemet leszámítva minden elem generátorelem (1.3.6).

Algoritmus 7: A csoport generáló $\mathcal{G}$ algoritmus

Input: 1^n és $l = l(n)$ paraméter

Output: A G ciklikus csoport q prímrenddel és g generátorelemmel

1 n bites random q prím generálása

2 l bites p prím generálása, melyre $q \mid (p-1)$

3 $h \in_R \mathbb{Z}_p^*$ választása $h \neq 1$ feltétellel

4 $g := [h^{(p-1)/g} \mod p]$ meghatározása

5 **return** p, q, g

Egy $h \in \mathbb{Z}_p^*$-re $h \in G$ pontosan akkor teljesül, ha $h^q \equiv 1 \pmod{p}$ is. Ez azért igaz, mert ha $h = g^i$, ahol $g \in \mathbb{Z}_p^*$ egy generátorelem, akkor

$$h^q \equiv 1 \pmod{p} \Leftrightarrow g^{iq} \equiv 1 \pmod{p} \Leftrightarrow iq \equiv 0 \pmod{p-1} \Leftrightarrow rq \mid iq \Leftrightarrow r \mid i,$$

felhasználva az 1.3.3-as állítást. Tehát $h = g^i = g^{cr} = (g^c)^r$, megfelelő c-re, így $h \in G$.

Az algoritmusban szereplő $l = \|p\|$ paraméter választásához vissza kell emlékeznünk a diszkrét logaritmus számító algoritmusok futásidejére. $\mathbb{Z}_p^*$-nek egy q rendű részcsoportjában a diszkrét logaritmus kiszámítása a 2.2-es alfejezetben leírtak szerint a legjobb futásidő $\mathcal{O}(\sqrt{q}) = \mathcal{O}(2^{n/2})$ és $2^{\mathcal{O}((\log p)^{1/3} \cdot (\log(\log p))^{2/3})} = 2^{\mathcal{O}(l^{1/3} \cdot (\log l)^{2/3})}$. Fix n esetén l választásánál törekedünk kell arra, hogy a futásidők egyensúlyba legyenek. Kisebb l kevesebb biztonságot, míg nagyobb l lassabb műveleteket jelent.

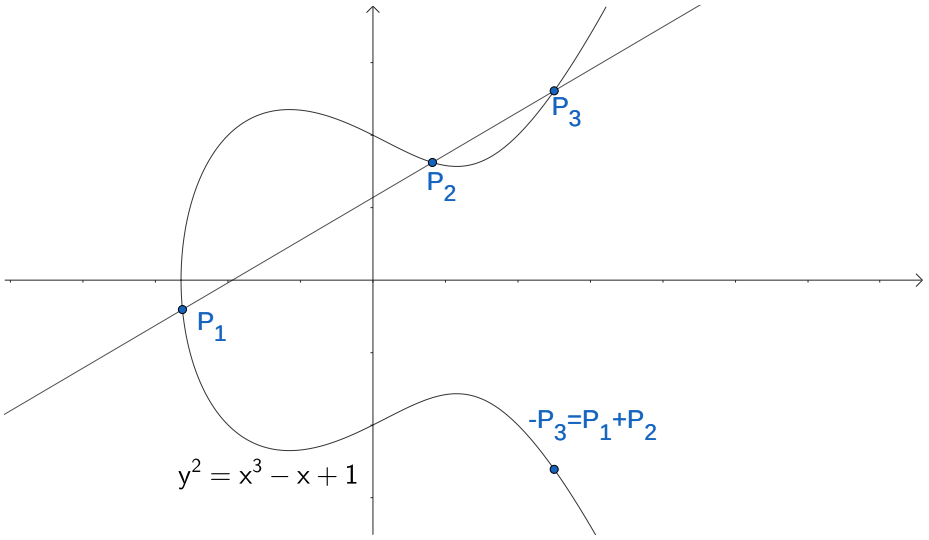
3.2.1. Elliptikus görbék

Az elliptikus görbéken nem ismert szubexponenciális futásidejű algoritmus sem a diszkrét logaritmus kiszámítására, így azok a kritográfiai rendszerek, melyek elliptikus görbéken vannak implementálva a $\mathbb{Z}_p^*$ prímrendű részcsoportjai helyett, azok hatékonyabbak.

Legyen $p \geq 5$ prím. Tekintsük a következő egyenletet, ahol x és y ismeretlenek:

$$y^2 \equiv x^3 + Ax + B \pmod{p}, \tag{3.1}$$

és az A, $B \in \mathbb{Z}_p$ konstansokra $4A^3 + 27B^2 \neq 0 \pmod{p}$ teljesül, ezzel biztosítva, hogy $x^3 + Ax + B$-nek ne legyen többszörös gyöke. Jelölje $E(Z_p)$ az $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ pontpárok azon halmazát, melyek kielégítik a 3.1-es egyenletet, kiegészülve egy $\mathcal{O}$ ponttal. Ekkor a 3.1-es egyenlet szerint definiált E elliptikus görbe pontjait $E(\mathbb{Z}_p)$ jelöli az $\mathcal{O}$ pontot pedig az elliptikus görbe végtelen távoli pontjának nevezzük.



3.1. ábra. Az $y^2 = x^3 - x + 1$ egyenlet a valós számsíkon értelmezve.

Érdesmes az elliptikus görbe pontjaira a 3.1-es ábra szerint gondolnunk. Nyilván az ábra nem felel meg $E(\mathbb{Z}_p)$-nek, hiszen még előbbinek végtelen, utóbbinak véges sok pontja van, de rendkívül szemléletes és hasznos. A végtelen távol pont az $x = 0$ egyenes végtelen távoli pontja és rajta van minden függőleges egyenesen. Látható, hogy ha egy egyenes metszi $E(\mathbb{Z}_p)$-t két pontban, akkor azt pontosan 3 pontban metszi, abban az esetben, ha egy érintési pontot kétszer számoljnk, és függőleges egyenes esetén számoljuk a végtelen távoli pontot is.

Definiáljunk $E(\mathbb{Z}_p)$ pontjain egy $+$-szal jelölt műveletet a következőképp:

- Az $\mathcal{O}$ pont az identitás, azaz, $\forall\ P \in E(\mathbb{Z}_p)$-re $P + \mathcal{O} = \mathcal{O} + P = P$.
- A $P_1, P_2 \neq \mathcal{O}$ pontokra $P_1 + P_2$-t úgy kapjuk, hogy vesszük a P_1P_2 egyenes harmadik metszéspontját az $E(\mathbb{Z}_p)$-t leíró görbén, legyen ez P_3. Ha $P_3 = (x, y) \neq \mathcal{O}$, akkor $P_1 + P_2 = (x, -y)$, vagyis P_3 tükörképe az x tengelyre. Ha $P_3 = \mathcal{O}$, akkor $P_1 + P_2 = \mathcal{O}$.

Ha $P = (x, y)$ az $E(\mathbb{Z}_p)$ egy végtelen távolitól különböző pontja, akkor definíció szerint legyen P egyértelmű inverze $-P = (x, -y)$, ami nyiván szintén eleme $E(\mathbb{Z}_p)$-nek. Mivel az $(x, -y)$, (x, y) pontokon átmenő egyenes függőleges, így $-P + P = \mathcal{O}$ teljesül. Ha $y = 0$, akkor $P = -P$ egy érintő érintési pontja, és az érintő harmadik metszéspontja $\mathcal{O}$. Továbbá $\mathcal{O} = -\mathcal{O}$ természetesen.

Most az eddig leírtak szerint definiáljuk a $+$ műveletet $E(\mathbb{Z}_p)$-n, ezúttal precízen. Legyen $P_1 = (x_1, y_1)$ és $P_2 = (x_2, y_2)$ az $E(\mathbb{Z}_p)$ elliptikus görbe két, végtelen távolitól

különböző pontja. Egyszerűség kedvéért tegyük fel, hogy $x_1 \neq x_2$, egyenlőség esetén is nyilvánvaló a definció. Ekkor a P_1P_2 egyenes meredeksége $m = \left[\frac{y_2 - y_1}{x_2 - x_1} \mod p\right]$, valamint, mivel feltettük, hogy $x_1 \neq x_2$, ezért $(x_2 - x_1)$ inverze létezik $\mathbb{Z}_p$-ben, így a P_1P_2 egyenes egyenlete

$$y \equiv m \cdot (x - x_1) + y_1 \pmod{p}. \tag{3.2}$$

A harmadik metszéspontot a következő egyenlet megoldásával kapjuk:

$$(m \cdot (x - x_1) + y_1))^2 \equiv x^3 + Ax + B \pmod{p}.$$

Azon x-ek melyek kielégítik az egyenletet x_1, x_2 és $x_3 = [m^2 - x_1 - x_2 \mod p]$. Behelyettesítve x_3-at a 3.2-es egyenletbe, azt kapjuk, hogy $y_3 = [m(x_3 - x_1) + y_1 \mod p]$. Ahhoz, hogy megkapjuk $P_1 + P_2$-t, csak meg kell változtatnunk y_3 előjelét:

$$(x_1, y_1) + (x_2, y_2) = ([m^2 - x_1 - x_2 \mod p], [m(x_1 - x_3) - y_1 \mod p]).$$

Az eddigieket összefoglalja valamint kiterjeszti a következő állítás.

3.2.2. Állítás. *Legyen $p \geq 5$ prím és E az $y^2 \equiv x^3 + Ax + B \pmod{p}$ egyenlet által meghatározott elliptikus görbe, ahol $4A^3 + 27B^2 \not\equiv 0 \pmod{p}$. Legyenek P_1, $P_2 \neq \mathcal{O}$ az E pontjai, melyekre $P_1 = (x_1, y_1)$ és $P_2 = (x_2, y_2)$.*

1. *Ha $x_1 \neq x_2$, akkor $P_1 + P_2 = (x_3, y_3)$, ahol $x_3 = [m^2 - x_1 - x_2 \mod p]$ és $y_3 = [m \cdot (x_1 - x_3) - y_1 \mod p]$, ahol $m = \left[\frac{y_2 - y_1}{x_2 - x_1} \mod p\right]$.*
2. *Ha $x_1 = x_2$, de $y_1 \neq y_2$, akkor $P_1 = -P_2$, és így $P_1 + P_2 = \mathcal{O}$.*
3. *Ha $P_1 = P_2$ és $y_1 = 0$ akkor $P_1 + P_2 = 2P_1 = \mathcal{O}$.*
4. *Ha $P_1 = P_2$ és $y_1 \neq 0$, akkor $P_1 + P_2 = 2P_1 = (x_3, y_3)$, ahol $x_3 = [m^2 - 2x_1 \mod p]$, $y_3 = [m \cdot (x_1 - x_3) - y_1 \mod p]$ és $m = \left[\frac{3x_1^2 + A}{2y_1} \mod p\right]$.*

Talán meglepő lehet, de $E(\mathbb{Z}_p)$ pontjai a fent definiált összeadással Abel-csoportot alkotnak. A kommutativitás a definícióból adódik, az identitás az $\mathcal{O}$, azt, hogy minden pontnak van inverze pedig már korábban láttuk. Az asszocivitás is igazolható hosszas számolás árán.

Kérdés még, hogy hány pontból állhat $E(\mathbb{Z}_p)$. Az $y^2 \equiv f(x) \pmod{p}$ egyenletnek 2 megoldása van ha $f(x)$ valamely $y \in \mathbb{Z}_p^*$-nek a négyzete mod p és 1 ha $f(x) \equiv 0 \pmod{p}$,

valamint $\mathbb{Z}_p^*$ elemeinek fele négyzete valamely elemnek mod p. Ez alapján gondolhatjuk, hogy a görbe pontjainak száma $2 \cdot \frac{p-1}{2} + 1$, ami kiegészülve a végtelen távoli ponttal összesen $p+1$ pont. A Hasse becslés szerint ezzel nem is járunk olyan messze az igazságtól.

3.2.3. Tétel (Hasse becslés)**.** *Legyen p prím és E a $\mathbb{Z}_p$ feletti elliptikus görbe. Ekkor* $p + 1 - 2\sqrt{p} \leq |E(\mathbb{Z}_p)| \leq p + 1 + 2\sqrt{p}$.

Továbbá az is igaz, hogy ha $A, B \in_R \mathbb{Z}_N$, feltéve, hogy $4A^3 + 27B^2 \not\equiv 0 \pmod{p}$, akkor $|E(\mathbb{Z}_p)|$ eloszlása a Hasse becslés által meghatározott intervallumon közel van az egyenleteshez.

4. fejezet

Kriptográfiai feltevések

4.1. A faktorizációs feltevés

4.1.1. Definíció. *Az $f : \mathbb{N} \to \mathbb{R}_0^+$ függvény elhanyagolható, ha minden p pozitív polinomhoz létezik, olyan N, hogy $\forall\, n > N$-re $f(n) < \frac{1}{p(n)}$.*

Legyen GenModulus egy polinom idejű algoritmus, melynek inputja 1^n, outputja pedig p, q, N, ahol $p \cdot q = N$ és p, q n bites prímek elhanyagolható valószínűségtől eltekintve. Erre egy kézenfekvő módszer n bites prímeket generálni, majd összeszorozni őket. Tekintsük a következő kísérletet egy adott $\mathcal{A}$ algoritmus esetén:

A faktorizációs kísérlet $\mathrm{Factor}_{\mathcal{A},\mathrm{GenModulus}}(n)$:

1. GenModulus futattása, hogy megkapjuk p, q, N-t.
2. $\mathcal{A}$ inputként megkapja N-et, outputként visszaadja p', $q' > 1$-et.
3. A kísérlet eredménye 1, ha $p' \cdot q' = N$, 0 különben.

Jegyezzük meg, hogy az, hogy a kísérlet eredménye 1 az azt jelenti, hogy $\{p', g'\} = \{p, q\}$, kivéve ha p és q összetettek, aminek elhanyagolható az esélye. Most már kimondhatjuk a faktorizációs feltevést.

4.1.2. Definíció. *A faktorizáció nehéz* GenModulus*-hoz képest, ha minden $\mathcal{A}$ probabilisztikus polinom idejű algoritmus esetén létezik egy $\boldsymbol{e}$ elhanyagolható függvény, hogy*

$$\mathrm{P}(\mathrm{Factor}_{\mathcal{A},\mathrm{GenModulus}}(n) = 1) \leq \boldsymbol{e}(n).$$

Tehát a feltevés szerint létezik olyan GenModulus, amihez képest a faktorizáció nehéz. Az, hogy GenModulus egy olyan N-et ad amit nehéz faktorizálni az nem triviális, erre példa Pollard $p-1$ algoritmusa, ahol azt látjuk, hogy ha p-nek csak kicsi prímosztói vannak akkor a faktorizáció nem nehéz.

4.2. Az RSA feltevés

A faktorizációs problémát évszázadokon keresztül tanulmányozták, de hatékony algoritmus nem született annak megoldására. Ezért próbáltak olyan problémákat létrehozni melyek nehézsége visszavezethető a faktorizációra. A leghíresebb ezek közül az RSA probléma.

Adott egy N modulus és egy $e > 2$, amely relatív prím $\phi(N)$-hez. Ekkor $[y^{1/e} \mod N] = x$ pontosan akkor, ha $x^e = [y \mod N]$, ahol minden $y \in \mathbb{Z}_N^*$ esetén $x \in \mathbb{Z}_N^*$ egyértelmű, ez következik a 1.2.7-es állításból. Az RSA probléma informálisan az, hogy meghatározzuk $[y^{1/e} \mod N]$-et olyan N esetén melynek nem ismert a faktorizációja.

Legyen GenRSA egy probabilisztikus polinom idejű algoritmus, melynek inputja 1^n, outputja egy N modulus, amely két n bites prím szorzata, valamint az $e, d > 0$ egészek, melyekre $\text{lnko}(e, \phi N) = 1$, illetve $ed \equiv 1 \pmod{\phi(N)}$, ilyen d létezik, mert e invertálható mod $\phi(N)$. Tekintsük a következő kísérletet adott $\mathcal{A}$ algoritmus és n paraméter esetén:

Algoritmus 8: GenRSA

Input: 1^n biztonsági paraméter

Output: N, e, d a leírtak szerint

1 $(p, q, N) \longleftarrow \text{GenModulus}(1^n)$

2 $\phi(N) = (p-1) \cdot (q-1)$

3 $e > 2$ választása, melyre $\text{lnko}(e, \phi(N)) = 1$

4 $d := [e^{-1} \mod \phi(N)]$ kiszámítása

5 **return** N, e, d

Az RSA kísérlet $\text{RSA} - \text{inv}_{\mathcal{A},\text{GenRSA}}(n)$:

1. GenRSA futattása, hogy megkapjuk N, e, d-t.
2. $y \in_R \mathbb{Z}_N^*$ választása.

3. $\mathcal{A}$ inputként megkapja N, e, y-t, outputként visszaad egy $x \in \mathbb{Z}_N^*$ elemet.

4. A kísérlet eredménye 1, ha $x^e \equiv y \pmod{N}$, 0 különben.

4.2.1. Definíció. *Az RSA probléma nehéz* GenRSA*-hoz képest, ha minden $\mathcal{A}$ probabilisztikus polinom idejű algoritmus esetén létezik $\boldsymbol{e}$ elhanyagolható függvény, hogy*

$$\mathrm{P}(\mathrm{RSA} - \mathrm{inv}_{\mathcal{A},\mathrm{GenRSA}}(n) = 1) \leq \boldsymbol{e}(n).$$

Tehát a feltevés szerint létezik olyan GenRSA, amelyhez képest az RSA probléma nehéz.

Az e választása nem befolyásolja számottevően az RSA probléma nehézségét, standard választás e-re a $2^{16}+1 = 65537$, egy prím alacsony Hamming súllyal. Egy szám Hamming súlya a bináris reprezentációjában az 1-esek száma. Azért célszerű ilyen e-t választani, mert a hatékony algoritmus a moduláris hatványozásra az alábbi:

Algoritmus 9: Hatékony algoritmus moduláris hatványozásra

Input: N modulus, $a \in \mathbb{Z}_N$ alap és $b > 0$ egész kitevő

Output: $[a^b \mod N]$

```
1 x := a
2 t := 1
3 while b > 0 do
4     if b páratlan then
5         t := [t · x mod N]
6         b := b − 1
7     x := [x^2 mod N]
8     b := b/2
9 return t
```

Az algoritmus során az $[a^b \mod N] = [t \cdot x^b \mod N]$ egyenlőség végig fennáll, és addig fut az algoritmus, amíg $x^b = 1$ nem teljesül. Látható, hogy pontosan annyiszor lépünk be az **if** ágba ahányszor a b bináris reprezentációjában 1-es van.

A d kiszámítása a következőképp történik: Tudjuk, hogy $\mathrm{lnko}(e, \phi(N) = 1$, az 1.1.2-es állítás szerint, pedig léteznek X, Y egészek, hogy $eX + \phi(N)Y = 1$, az Euklideszi algoritmussal pedig ki is tudjuk őket polinomiális futásidőben számítani. Innen $eX - 1 = -\phi(N)Y$, amiből pedig $eX \equiv 1 \pmod{\phi(N)}$ következik, tehát $d := X$. Ha d ismert akkor

az RSA probléma megoldható. Ha d kicsi, akkor brute force-szal meg tudjuk találni ezért olyan d használata biztonságos, melyre $N^{1/4} < d$.

A teljesebb kép érdekében definiáljuk még az RSA titkosítást, amely egy nyílt kulcsú titkosítási séma.

4.2.2. Definíció. *Egy nyilvános kulcsú titkosítási séma egy* $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ *hármas melyekre:*

- Gen*: A kulcsgenerálás, valószínűségi algoritmus, melynek inputja* 1^n*, outputja* $k = (pk, sk) \in K$*, melyekre* $|pk|, |sk| \geq n$*, ahol* pk *a nyilvános kulcs,* sk*, pedig a titkos kulcs.*
- Enc*: A titkos üzenet legyártása, valószínűségi algoritmus, melynek inputja* pk *nyilvános kulcs és* $m \in M$*, outputja egy* $c = \mathrm{Enc}_{pk}(m) \in C$ *titkos üzenet.*
- Dec*: A visszafejtés, determinisztikus algoritmus, melynek inputja* sk *titkos kulcs és* c *titkos üzenet, outputja egy* $\mathrm{Dec}_{sk}(c) \in M$ *üzenet.*

M *az üzenettér,* C *a titkos üzenetek tere,* K *pedig a kulcstér. A titkosítási séma korrekt, ha* $\forall\, n, pk, sk$*-ra és* $\forall\, m \in M$ *esetén* $\mathrm{Dec}_{sk}(\mathrm{Enc}_{pk}(m)) = m$.

Az RSA algoritmus esetén az $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ hármas az alábbi:

- Gen: Lefuttatjuk GenRSA-t a 8. algoritmusban leírtak szerint, $pk = (N, e)$, $sk = (p, q, d)$.
- Enc: Egy $m < N$ üzenet esetén a titkos üzenet $\mathrm{Enc}_{pk}(m) = [m^e \mod N] = c$.
- Dec: A visszafejtés pedig adott c esetén $\mathrm{Dec}_{sk}(c) = [c^d \mod N]$.

Az RSA egy korrekt titkosítási séma, mivel $\mathrm{Dec}_{sk}(\mathrm{Enc}_{pk}(m)) = [m^{ed} \mod N] = m$. A gyakorlatban ezt sosem így használják, mert az nem lenne biztosnágos. Gyakori, hogy m elé egy random bitsorozatot konkatenálnak és azt titkosítják el.

4.3. A faktorizációs feltevés és az RSA feltevés kapcsolata

Legyen GenRSA a 8. algoritmusban leírtak szerinti. Ha N-t tudjuk faktorizálni, akkor ki tudjuk számolni $\phi(N)$-et és ezáltal $d = [e^{-1} \mod \phi(N)]$-et is. Tehát ahhoz, hogy az

RSA probléma nehéz legyen GenRSA-hoz képest, a faktorizációnak is nehéznek kell lenni GenModulus-hoz képest, azaz az RSA probléma nem lehet nehezebb a faktorizációnál.

Ami a másik irányt illeti, az nyitott kérdés. A legtöbb amit tudunk mutatni, hogy d kiszámítása ismert N, e esetén olyan nehéz, mint a faktorizáció.

4.3.1. Tétel. *Létezik probabilisztikus polinom idejű algoritmus, amely inputként megkapja N-et, valamint e, d-t melyekre $ed \equiv 1 \pmod{\phi(N)}$ és outputként megadja N egy nemtriviális osztóját, elhanyagolható valószínűségtől eltekintve.*

Bizonyítás. A tétel minden összetett N-re igaz, de mi csak abban az esetben bizonyítjuk ha N két páratlan prím szorzata, mivel ez használatos leginkább a kriptográfiában.

A 2.1.5-ös állítás alapján, az 1-nek 4 db négyzetgyöke van $\bmod N$, ha N két páratlan prím szorzata. Ebből kettő a ± 1, a másik kettő viszont nemtriviális. Ha y egy nemtriviális gyök, akkor

$$0 \equiv y^2 - 1 \equiv (y-1)(y+1) \pmod{N} \Rightarrow N \mid (y-1)(y+1),$$

de $N \nmid (y-1)$ és $N \nmid (y+1)$, így ekkor $\text{lnko}(y-1, N)$ az N egy prímosztója.

Legyen $k = ed - 1$. Ekkor $\phi(N) \mid k$, és az Euler–Fermat tétel szerint $x^k \equiv 1 \pmod{N}$ $\forall\, x \in \mathbb{Z}_N^*$-re. Legyen $k = 2^r u$, ahol u páratlan és $r \geq 1$.

A stratégiánk a következő lesz: vegyünk $x \in_R \mathbb{Z}_N^*$ értékeket és számoljuk ki a

$$x^u,\, x^{2u}, \ldots,\, x^{2^r u}$$

sorozat tagjait, mindegyiket $\bmod N$. Ebben a sorozatban minden elem az előtte állónak a négyzete $\bmod N$. Legyen h a legnagyobb olyan, amelyre $y = [x^{2^h u} \bmod N] \neq 1$. Ekkor $y^2 \equiv 1 \pmod{N}$ és ha $y \not\equiv -1 \pmod{N}$, akkor találtunk egy nem triviális négyzetgyököt mod N. Az összes felsorolt lépés polinomiális idejű, az egyetlen nyitott kérdés az, hogy $x \in_R \mathbb{Z}_N^*$ esetén mennyi az esélye annak, hogy y nemtriviális négyzetgyöke 1-nek $\bmod N$.

Legyen $i \in \{0, 1, .., r-1\}$ a legnagyobb olyan, melyhez létezik $x \in \mathbb{Z}_N^*$, melyre $x^{2^i u} \not\equiv 1 \pmod{N}$. Mivel u páratlan, biztosan van ilyen i. Ekkor $\forall\, x \in \mathbb{Z}_N^*$-re $x^{2^{i+1} u} \equiv 1 \pmod{N}$. Definiáljuk a Bad halmazt a következőképp: $\text{Bad} := \{x \mid x^{2^i u} \equiv \pm 1 \pmod{N}\}$. Ha az algoritmus olyan x-et választ mely nem eleme Bad-nek akkor megtaláljuk egy nemtriviális négyzetgyökét 1-nek $\bmod N$. Megmutatjuk, hogy Bad szigorú részcsoportja $\mathbb{Z}_N^*$-nak. Ekkor az 1.2.14-es lemma miatt $|\text{Bad}| \leq \dfrac{|\mathbb{Z}_N^*|}{2}$, és az algoritmust t-szer elismételve annak az esélye, hogy nem találunk nemtriviális négyzetgyököt kisebb, mint 2^{-t}.

Mivel $1 \in \text{Bad}$ és ha $x,\ x' \in \text{Bad}$, akkor

$$(xx')^{2^i u} \equiv x^{2^i u} \left(x'\right)^{2^i u} \equiv \pm 1 \cdot \pm 1 \equiv \pm 1 \pmod{N}$$

miatt $xx' \in \text{Bad}$ is teljesül, így az 1.2.13-as állítás szerint Bad részcsoportja $\mathbb{Z}_N^*$-nek. Ahhoz, hogy belássuk, hogy Bad valós részcsoport, keresnünk kell egy x elemet amelyre, $x \in \mathbb{Z}_N^*$, és $x^{2^i u} \not\equiv 1 \pmod{N}$ teljesül. Ekkor ha $x^{2^i u} \not\equiv -1 \pmod{N}$, akkor már meg is vagyünk. Ellenkező esetben legyen $N = pq$, és $x \leftrightarrow (x_q, x_p)$ a Kínai maradéktétel szerinti reprezentáció. Ekkor

$$x^{2^i u} \leftrightarrow (x_q, x_p)^{2^i u} = (x_q^{2^i u}, x_p^{2^i u}) \equiv (-1, -1) \leftrightarrow -1.$$

De akkor pl. $(x_p, 1) \notin \text{Bad}$, mivel

$$(x_p, 1)^{2^i u} = ([x_p^{2^i u} \mod p], 1) = (-1, 1) \not\leftrightarrow \pm 1.$$

Ezzel beláttuk, hogy $\text{Bad} < \mathbb{Z}_N^*$, amivel teljes a bizonyítás. □

Feltéve, hogy a faktorizáció nehéz, a fenti eredmény azt jelenti, hogy nincs hatékony megoldás az RSA problémára, ha először ki akarjuk számolni d-t.

4.4. A diszkrét logaritmus feltevés

Bár azonos rendű ciklikus csoportok izomorfak, ez nem jelenti, azt, hogy az számolás ezekben a csoportokban egyformán nehéz. Pl. hiába $\mathbb{Z}_p^* \simeq \mathbb{Z}_{p-1}$, utóbbiban a $\bmod p-1$ összeadások egymásutánja helyettesíthető a $\bmod p-1$ szorzással, amit el tudunk hatékonyan végezni, így $\mathbb{Z}_{p-1}$-ben nem nehéz a diszkrét logaritmus probléma, viszont $\mathbb{Z}_p^*$-ban igen. Ahhoz azonban, hogy a következő problémákról beszélhessünk, szükéges az, hogy a csoportműveletet, illetve annak eldöntését, hogy egy adott elem benne van-e a csoportban el tudjuk végezni polinomiális időben.

Legyen $\mathcal{G}$ egy polinom idejű algoritmus amelynek inputja 1^n, outputja pedig egy G ciklikus csoportnak a q rendje ($\|q\| = n$) és egy g generátoreleme. Tekintsük a következő kísérletet $\mathcal{G}$-vel, adott $\mathcal{A}$ algoritmus és n paraméter esetén.

A diszkrét logaritmus kísérlet $\text{DLog}_{\mathcal{A},\mathcal{G}}(n)$:

1. $\mathcal{G}$ futattása, hogy megkapjuk G, g, q-t, ahol G ciklikus, q rendű csoport ($\|q\| = n$) g generátorelemmel.

2. $h \in_R G$ választása.

3. $\mathcal{A}$ inputként megkapja G, g, q, h-t, outputként visszaad egy $x \in \mathbb{Z}_q$ elemet.

4. A kísérlet eredménye 1, ha $g^x = h$, 0 különben.

4.4.1. Definíció. *A diszkrét logaritmus probléma nehéz $\mathcal{G}$-hez képest, ha minden probabilisztikus polinom idejű $\mathcal{A}$ algoritmushoz létezik egy $\boldsymbol{e}$ elhanyagolható függvény, hogy*

$$\Pr[\mathrm{DLog}_{\mathcal{A},\mathcal{G}}(n) = 1] \leq \boldsymbol{e}(n).$$

Tehát a diszkrét logaritmus feltevés az az, hogy van olyan $\mathcal{G}$, amihez képest a diszkrét logaritmus probléma nehéz. Arról, hogy egy ilyen $\mathcal{G}$ hogyan nézhet ki már írtunk a 3.2-es alfejezetben.

Külön szekcióban nem tárgyaljuk, de itt szót ejtünk a diszkrét logaritmus és a faktorizáció kapcsolatáról. Tudjuk, hogy $\mathbb{Z}_p^*$-ban mindkét problémára a leggyorsabb algoritmus szubexponenciális futásidejű. Kissé meglepő módon viszont nem egyformán nehezek, a következőket tudjuk elmondani a két problémáról:

- Ha meg tudjuk oldani $a^x \equiv b \mod N$-et polinomiális időben akkor nagy eséllyel meg tudjuk találni egy prímosztóját N-nek.

- Az $a^x \equiv b$ modulo egy p prím probléma nehézsége pontosan ugyanaz, mint az $a^x \equiv b$ modulo egy p^e prímhatvány problémáé. Ha létezik megoldás modulo p^e, akkor azt polinom időben megkaphatjuk a modulo p megoldásból.

- Ahhoz, hogy megoldjuk $a^x \equiv b \mod N$-et ahhoz elegendő megoldani a problémát modulo N prímosztói.

Összefoglalva, megoldani a diszkrét logaritmus problémát modulo egy N öszzetett egyész pontosan olyan nehéz, mint a faktorizálni N-et és megoldani a problémát modulo N prímosztói.

4.5. A Diffie–Hellman problémák és kulcscsere

Beszélhetünk számítási (CDH) vagy eldöntési (DDH) Diffie–Hellman problémáról. Legyen G ciklikus csoport g generátorelemmel. Adottak a h_1, $h_2 \in G$ elemek és definíció szerint legyen $\mathrm{DH_g}(h_1, h_2) := g^{\log_g h_1 \log_g h_2}$. Ekkor ha $h_1 = g^{x_1}$ és $h_2 = g^{x_2}$, akkor

$$\mathrm{DH_g}(h_1, h_2) := g^{x_1 \cdot x_2} = h_1^{x_2} = h_2^{x_1}.$$

A CDH probléma kiszámolni $\mathrm{DH_g}(h_1, h_2)$-t adott h_1 és h_2 esetén.

A számítási Diffie–Hellman kísérlet $\mathrm{CDH}_{\mathcal{A},\mathcal{G}}(n)$:

1. $\mathcal{G}$ futattása, hogy megkapjuk G, g, q-t, ahol G ciklikus, q rendű csoport ($\|q\| = n$) g generátorelemmel.
2. x_1, $x_2 \in_R G$ választása.
3. $\mathcal{A}$ inputként megkapja G, g, q, g^{x_1}, g^{x_2}-t, outputként visszaad egy $y \in G$ elemet.
4. A kísérlet eredménye 1, ha $y = g^{x_1 \cdot x_2}$, 0 különben.

4.5.1. Definíció. *A CDH probléma nehéz $\mathcal{G}$-hez képest, ha minden probabilisztikus polinom idejű $\mathcal{A}$ algoritmushoz létezik egy $\boldsymbol{e}$ elhanyagolható függvény, hogy*

$$\mathrm{P}(\mathrm{CDH}_{\mathcal{A},\mathcal{G}}(n) = 1) \leq \boldsymbol{e}(n).$$

Ha a diszkrét logaritmus probléma könnyű egy $\mathcal{G}$-hez képest, akkor a CDH probléma is: adott h_1 és h_2, először kiszámoljuk $x_1 := \log_g h_1$-et, majd visszaadjuk $h_2^{x_1}$-et, ugyanakkor ez fordítva nem igaz. A DDH probléma ezzel szemben megkülönböztetni $\mathrm{DH_g}(h_1, h_2)$-t egy véletlen csoportelemtől.

4.5.2. Definíció. *Azt mondjuk, hogy a DDH probléma nehéz $\mathcal{G}$-hez képest, ha minden probabilisztikus polinom idejű $\mathcal{A}$ algoritmushoz létezik egy $\boldsymbol{e}$ elhanyagolható függvény, hogy*

$$|\mathrm{P}(\mathcal{A}(G,\, q,\, g,\, g^x,\, g^y,\, g^z) = 1) - \mathrm{P}(\mathcal{A}(G,\, q,\, g,\, g^x,\, g^y,\, g^{xy}) = 1)| \leq \boldsymbol{e}(n),$$

ahol a valószínűségek a megszokott kísérletből származnak. Ha a CDH probléma könnyű $\mathcal{G}$-hez képest, akkor a DDH probléma is, ez nyílvánvaló. Fordítva az állítás nem igaz, mert vannak olyan csoportok ahol a CDH probléma feltehetően nehéz, de a DDH nem, ilyen például $\mathbb{Z}_p^*$. Elliptikus görbék esetén, mvel a csoportműveletet $+$-szal jelöljük, ezért a számolási Diffie–Hellman probléma meghatározni adott xP esetén x-et, az eldöntési pedig megkülönböztetni $(xP,\, yP,\, cP)$-t $(xP,\, yP,\, xyP)$-től. Ezek a problémák feltehetőleg nehezek az elliptikus görbe csoportokban.

Ami a Diffie–Hellman feltevések alkalmazását illeti, a kriptográfiában alapvető fontosságú probléma, hogy két fél, Alíz és Bob hogyan tudnak egymással biztonságosan kommunikálni, még akkor is ha esetleg a kommunikációjukat lehallgatják. Ehhez szükség van egy közös kulcsra és erre ad egy megoldást a Diffie–Hellman kulcscsere.

A Diffie–Hellman kulcscsere lépései:

1. Alíz futtatja $\mathcal{G}$-t, hogy megkapja G, g, q-t, ahol G ciklikus, q rendű csoport ($\|q\| = n$) g generátorelemmel.
2. Alíz választ egy $x_1 \in_R G$-t, és kiszámolja $h_A := g^{x_1}$-et.
3. Alíz elküldi (G, q, g, h_A)-t Bob-nak.
4. Bob választ egy $x_2 \in_R$-t, és kiszámolja $h_B = g^{x_2}$.
5. Bob elküldi h_B-t Alíznak, valamint kiszámolja $h_A^{x_2}$-t, ezzel megkapja a közös kulcsot.
6. Alíz kiszámolja $h_B^{x_1}$-et, ezzel ő is megkapja a közös kulcsot.

Az Alíz és Bob által kapott közös kulcs valóban ugyanaz, mivel $h_A^{x_2} = g^{x_1 \cdot x_2} = h_B^{x_1}$. Kérdéses még, hogy mikor biztonságos a kulcscsere egy lehallgató ellen. Egy kulcscserét akkor tekintünk definíció szerint biztonságosnak, ha egy probabilisztikus polinomiális idejű algoritmus nem tudja elhanyagolhatónál nagyobb eséllyel megkülönböztetni a közös kulcsot egy véletlen G-beli elemtől.

Egy lehallgató hozzáférhet a g^{x_1}, g^{x_2}, g, q, G elemek bármelyikéhez, de x_1, x_2 és $g^{x_1 \cdot x_2}$-höz nem juthat hozzá lehallgatás útján, hiszen ezeket nem küldi el egymásnak Alíz és Bob. A kulcscsere biztonságához nélkülözhetetlen, hogy a diszkrét logaritmus probléma nehéz legyen $\mathcal{G}$-hez képest, hiszen ellenkező esetben a lehallgató g^{x_1} és g ismeretében ki tudná számolni x_1-et, majd $h_B^{x_1}$-t. Továbbá, mivel a biztonságos kulcscsere definíciója megkülönböztethetetlenségről szól, szükség van arra is, hogy a DDH probléma nehéz legyen G-ben, a CDH nem elég.

Irodalomjegyzék

[1] Jonathan Katz, Yehuda Lindell. *Introduction to modern cryptography, second edition.* Chapman & Hall/CRC Press, 2015. Felhasznált oldalak: 285-358.

[2] Kiss Emil. *Bevezetés az algebrába.* Typotex kiadó, 2007. Felhasznált oldalak: 115-168.

[3] Király Zoltán. *Algoritmuselmélet.* Typotex kiadó, 2014, 1.80 verzió. Felhasznált oldalak: 31-32.

[4] Eric Bach. *Discrete logarithms and factoring.* Computer Science Division, University of California Berkeley, 1984. Elérhető: https://www2.eecs.berkeley.edu/Pubs/TechRpts/1984/CSD-84-186.pdf.

[5] Gyarmati Katalin. *Elemi számelmélet.* Eötvös Loránd Tudományegyetem, Egyetemi Jegyzet, 2022. Felhasznált oldalak: 21-22. és 111-115.

[6] A 3.1-es ábra saját készítésű.