1. Bevezetés

Az egyéni kutatómunka 1 tantárgy keretein belül végzett munkám támája a diszkrét logaritmus probléma és a Diffie-Hellman kulcscsere. A félév során megismerkedtem a téma alapvető problémáival, valamint több megoldási algoritmussal is. A diszkrét logaritmus probléma nehézsége alapvető szerepet játszik számos kriptográfiai protokoll biztonságában, egy ilyen például a Diffie-Hellman kulcscsere. A probláma megoldására alkalmas algoritmusok közül a bébi-lépés-óriás-lépés algoritmust, az index kalkulus módszert valamint a Pollard-féle ρ-módszert fogom ismertetni, melyek közük kettőt le is programoztam a SageMath matematikai programban, ez is féléves munkám részét képezve.

2. A diszkrét logaritmus probléma

Általánosan a diszkrét logaritmus probláma az alábbi módon fogalmazható meg: legyen adott egy G véges csoport és legyen $g \in G$ egy generátorelem ebben a csoportban, valamint $c \in G$ egy tetszőleges másik elem. Keressük azt az x egész számot, melyre teljesül az alábbi:

$$c \equiv g^x.$$

Ezt a problámát röviden DLP-nek szokás hívni. A továbbiakban a DLP-t $G = \mathbb{Z}_p$ és $c \in \mathbb{Z}_p^*$ esetben vizsgáljuk. A DLP megoldása általános esetben nehéz, azonban számos algoritmus ismert a megoldásra. Ennek ellenére a DLP nem oldható meg gyorsan.

2.1. Megoldási módszerek

2.1.1. Bébi-lépés-Óriás-lépés

A Bébi-lépés-Óriás-lépés algoritmust Daniel Shanks ismertette. Az algoritmus előnye, hogy nemcsak $\mathbb{Z}_p$-ben, hanem tetszőleges ciklikus csoportban is alkalmazható.

Legyen G véges multiplikatív, ciklikus csoport, melynek rendje n és $g \in G$ egy rögzített generátoreleme. Az $a = g^x$ egyenlet megoldását keressük.

Az algoritmus lépései röviden: Legyen $m = \lceil \sqrt{n} \rceil$, és számoljuk ki minden $i \in \{1, \ldots, m\}$ esetén g^i értékét, és az (i, g^i) párokat rögzítsük egy táblázatba, majd a táblázat elemeit rendezzük növekvő sorrendbe g^i értékei szerint. Ezután számítsuk ki g^{-m} értékét. Legyen $j \in \{1, \ldots, m-1\}$ és számítsuk ki az ag^{-jm} értékeket, majd ellenőrizzük, hogy ezek közül van-e olyan, amely szerepel a táblázatunkban. Ha találunk ilyen indexet, azaz $ag^{-jm} = g^i$, akkor ebből kapjuk, hogy $a = g^{jm+i}$, mely a diszkét logaritmus probléma keresett megoldása.

Ezt az algoritmust SageMath kód segítségével is ismertetem, $G = \mathbb{Z}_p$ esetén:

```
m=ceil(sqrt(n))
L=[(g^k)%n for k in [0..m-1]]
j=0
while not (a*g^(-m*j))%n in L:
    j=j+1
i=L.index((a*g^(-m*j))%n)
sol=(i+j*Nm)%(m-1)
print(sol)
```

2.1.2. Index-kalkulus módszer

Az index kalkulus módszer a DLP megoldásához bemenetnek használ egy úgynevezett faktorbázist. Ez a faktorbázis t darab kicsi prímet tartalmaz, $B = \{p_1, \ldots, p_t\}$. Az algoritmus három lépésből áll. Az első lépés abból áll, hogy keresünk $s \gg t$ darab lineárisan független összefüggést a faktorbázis és a g generátor hatványai között. Azaz olyan u-kat keresünk, melyekre fenáll, hogy $g^u \equiv p_1^{\gamma_1} p_2^{\gamma_2} \cdots p_t^{\gamma_t} \mod p$.

A második lépésben felírunk ezekre az indexekre vonatkozóan egy egyenletrendszert:

$$\begin{aligned} \gamma_{11}\mathrm{ind}_g(p_1) + \ldots + \gamma_{1t}\mathrm{ind}_g(p_t) &\equiv u_1 \mod p-1 \\ &\vdots \\ \gamma_{s1}\mathrm{ind}_g(p_1) + \ldots + \gamma_{st}\mathrm{ind}_g(p_t) &\equiv u_s \mod p-1 \end{aligned}$$

Majd ezt megoldva megkapjuk $\mathrm{ind}_g(p_i)$ értékeit $(i = 1, \ldots, t)$.

A harmadik lépésben olyan α értéket keresünk, amelyre ag^α felírható a

faktorbázis elemeinek segítségével:

$$ag^{\alpha} \equiv p_1^{\gamma_1} p_2^{\gamma_2} \cdots p_t^{\gamma_t} \mod p$$

Ebből ind$a$ számolható, és így megoldottuk a DLP-t:

$$\text{ind}a \equiv -\alpha + \gamma_1 \,\text{ind}p_1 + \ldots + \gamma_t \,\text{ind}p_t \mod p-1.$$

2.1.3. Pollard-féle ρ-módszer

Legyen G egy ciklikus csoport, melynek rendje n, és g egy generátoreleme. Az $a = g^x$ DLP-t szeretnénk megoldani. Ehhez konstruálunk egy véletlen sétát G-ben, amelyet úgy alakítunk ki, hogy várhatóan egy azonos elem (ütközés) keletkezik, és ez majd lehetővé teszi a DLP kiszámítását. Legyen ez a véletlen séta: $x_0, x_1, \ldots$, ahol $x_i = a^{u_i} g^{v_i}$. Ebben a konstruálása miatt lesz egybeeső elem, azaz létezik olyan i és j, hogy

$$a^{u_i} g^{v_i} = a^{u_j} g^{v_j}.$$

Ezt rendezve, majd kihasználva, hogy g egy generátorelem, kapjuk, hogy:

$$a^{u_i - u_j} = g^{v_j - u_j}$$
$$(u_i - u_j)\,\text{ind}_g a \equiv v_j - u_j \mod n$$

Ekkor ha $(u_i - u_j, n) = 1$, akkor $\text{ind}_g a \equiv (v_j - u_j)(u_i - u_j)^{-1}$ és ezzel megoldottuk a DLP-t. Ahhoz, hogy működjön az algoritmus, egy megfelelő $f : G \to G$ függvényt kell találnunk, mely hozzárendeli x_i-hez x_{i+1}-et. Ezt egy SageMath kódon keresztül fogom ismertetni $G = \mathbb{Z}_p$ esetén, valamint az algoritmus általános leírása megtalálható az [1] cikkben. A fentiekben leírt x_i-ket, valamint az u_i és v_i értékeket a következő módon kapjuk: A sétánk kezdő értéke legyen $x_0 = 1$, és legyen $u_0 = 0$, $v_0 = 0$. Ezután legyen

$$x_{i+1} = \begin{cases} ax_i & \text{ha } 0 \le x_i < p/3 \\ x_i^2 & \text{ha } p/3 \le x_i < 2p/3 \\ bx_i & \text{ha } 2p/3 \le x_i < p \end{cases}$$

$$u_{i+1} = \begin{cases} u_i + 1 & \text{ha } 0 \le x_i < p/3 \\ 2u_i & \text{ha } p/3 \le x_i < 2p/3 \\ u_i & \text{ha } 2p/3 \le x_i < p \end{cases}$$

$$v_{i+1} = \begin{cases} v_i & \text{ha } 0 \leq x_i < p/3 \\ 2v_i & \text{ha } p/3 \leq x_i < 2p/3 \\ v_i + 1 & \text{ha } 2p/3 \leq x_i < p \end{cases}$$

```
def PollardRho(g,h,p):
def xn(A):
    if A[0]>=0 and A[0]<p/3:
        return[(g*A[0])%p,(A[1]+1)%(p-1),A[2]]
    if A[0]>=p/3 and A[0]<2*p/3:
        return [(A[0]^2)%p,(2*A[1])%(p-1),(2*A[2])%(p-1)]
    else:
        return[(h*A[0])%p,A[1],(A[2]+1)%(p-1)]
x0=[1,0,0]
y0=[1,0,0]
fut=True
while fut:
    x0=xn(x0)
    y0=xn(xn(y0))
    if x0[0]==y0[0]:
        fut=False
u0=(x0[1]-y0[1])%(p-1)
v0=(y0[2]-x0[2])%(p-1)
S=solve_mod(v0*x-u0,p-1)
return[k[0] for k in S if (g^k[0])%p==h][0]
```

3. A Diffie-Hellman kulcscsere

A Diffie Hellman kulcscserét a diszkrét logaritmus probléma kriptográfiai alkalmazásának is tekinthetjük. Ez egy olyan módszer, mely segítségével a kommunikáló felek meg tudnak állapodni egy közös titkos kulcsban, úgy, hogy nyilvános, nem titkos csatornán kommunikálnak. Ehhez a kommunikáló felek először megállapodnak egy G csoportban, és annak egy g generátorelemében (vagy egy nagy rendű elemben). Legyen most $G = \mathbb{Z}_p^*$, valamint legyen a két fél A és B. A közös kulcs konstruálása az alábbi módon zajlik:

- A választ egy titkos $1 \leq a \leq p-1$ égész számot, kiszámítja $g^a \mod p$ értékét, és elküldi B-nek
- B választ egy titkos $1 \leq b \leq p-1$ égész számot, kiszámítja $g^b \mod p$ értékét, és elküldi A-nak
- A kiszámítja $(g^b)^a \mod p$, B kiszámítja $(g^a)^b \mod p$ értékét, így mindketten megkapják a közös titkos kulcsot: $g^{ab} \mod p$

A módszer biztonságossága azon múlik, hogyha van egy egyén, aki lehallgatja A és B nyilvános kommunikációját, akkor ő g, $g^a \mod p$ és $g^b \mod p$ ismeretében nehezen tudja kiszámolni $g^{ab} \mod p$ értékét. Ezt szokás Diffie-Hellman problémának (DHP) nevezni. Látható, hogy ha valaki meg tudja oldani az általános DLP-t, akkor ki tudja számítani a és b értékét, és ezzel $g^{ab} \mod p$ értékét is, ezzel megoldva a DHP-t is. Az, hogy a megfodítás igaz-e, azaz hogyha ismerjük a DHP megoldását, abból következik, hogy meg tudjuk oldani a DLP-t is, egy nyitott kérdés.

Hivatkozások

[1] McCurley, K. S., *The Discrete Logarithm Problem, Cryptology and Computational Number Theory, volume 42, pages 49-74, American Mathematical Society, 1990.*

[2] K. Gyarmati, *Számítógépes Számelmélet, ELTE TTK Matematikai Intézet (2023)* `https://gyarmatikati.web.elte.hu/jegyzet/compszam2206.pdf`

[3] Wikipédia, *Discrete logarithm*, `https://en.wikipedia.org/wiki/Discrete_logarithm`

[4] Wikipédia, *Baby-step giant-step* `https://en.wikipedia.org/wiki/Baby-step_giant-step`

[5] Wikipédia, *Index calculus algorithm* `https://en.wikipedia.org/wiki/Index_calculus_algorithm`