Euler–Fermat-tétel és RSA

Számelmélet

Diszkrét modellek alkalmazásai

1. Redukált maradékrendszer

Korábban láttuk (lásd: Moduláris aritmetika), hogy $\mathbb{Z}_m$-ben pontosan azok az $\overline{a} \in \mathbb{Z}_m$ elemek invertálhatóak, amelyekre a és m relatív prímek, azaz $\text{lnko}(a, m) = 1$. Ezért érdemes ezeket az elemeket alaposabban megvizsgálni.

1.1. Definíció. Ha $m \in \mathbb{N}^+$, akkor $\mathbb{Z}_m^* := \{\overline{a} \in \mathbb{Z}_m \mid \text{lnko}(a, m) = 1\}$.

Azaz $\mathbb{Z}_m^*$ a $\mathbb{Z}_m$ azon elemei, amelyek relatív prímek m-hez. Például $\mathbb{Z}_{10}^* = \{\overline{1}, \overline{3}, \overline{7}, \overline{9}\}$, mert 1, 3, 7 és 9 relatív prímek 10-hez (míg a többiek, $0, 2, 4, 5, 6, 8$ oszthatók 2-vel vagy 5-tel).

Ha a modulus prímszám, akkor $\mathbb{Z}_m$ minden nemnulla eleme relatív prím m-hez, így ekkor $\mathbb{Z}_m^* = \{\overline{1}, \overline{2}, \ldots, \overline{m-1}\}$. Ez megfelel az előző jegyzetben (Diszkrét logaritmus) bevezetett $\mathbb{Z}_p^*$ jelölésnek – csak ott a "nemnulla" volt a definiáló tulajdonság, itt pedig az "invertálható" (ezek prímszámokra egybeesnek, de összetett számokra nem).

(Megjegyzés: ne feledjük, hogy a maradékosztályok halmazok, például $m = 10$ esetén $\overline{3} = \{10k + 3 \mid k \in \mathbb{Z}\} = \{\ldots, 3, 13, 23, \ldots\}$. De a fenti definíció nem függ attól, hogy a maradékosztályokat mely elemükkel reprezentáljuk, hiszen $\text{lnko}(a, m) = \text{lnko}(a + m, m)$, azaz ha a relatív prím m-hez, akkor $a + m$, $a + 2m$ stb. is az, vagyis az $\overline{a}$ maradékosztály összes eleme relatív prím m-hez. Ezért az egyszerűség kedvéért azt is mondjuk, hogy ekkor maga az $\overline{a}$ maradékosztály relatív prím m-hez.)

Az előző jegyzetben láttuk, hogy prímszám modulus esetén $\mathbb{Z}_p^*$-ban a szorzás és az osztás mindig elvégezhető, és nem vezet ki $\mathbb{Z}_p^*$-ból. Ugyanez igaz tetszőleges modulusra is:

1.2. Tétel. *Ha $m \in \mathbb{N}^+$, akkor $\forall \overline{a}, \overline{b} \in \mathbb{Z}_m^* : \overline{a} \cdot \overline{b} \in \mathbb{Z}_m^* \ \wedge \ \overline{a}/\overline{b} \in \mathbb{Z}_m^*$.*

Ez egy sokkal erősebb állítás, mint prímszám modulus esetén (ahol csak a $\overline{0}$-t zártuk ki). Például a $\mathbb{Z}_{10}^* = \{\overline{1}, \overline{3}, \overline{7}, \overline{9}\}$ elemeit bárhogy szorozzunk vagy osztjuk egymással, az eredmény mindig $\mathbb{Z}_{10}^*$-ban marad: $\overline{3} \cdot \overline{3} = \overline{9}$, $\overline{7} \cdot \overline{3} = \overline{1}$, $\overline{9}/\overline{7} = \overline{7}$ stb.

Korábban (Moduláris aritmetika) megismertük a *teljes maradékrendszer* fogalmát, amely minden maradékosztályból pontosan egy elemet tartalmaz, például $\{0, 1, 2, \ldots, m-1\}$. Most megismerjük a következőt:

1.3. Definíció. Egy halmazt *redukált maradékrendszer*nek nevezünk, ha minden $\overline{a} \in \mathbb{Z}_m^*$ maradékosztályból pontosan egy elemet tartalmaz.

Például $m = 10$-re redukált maradékrendszer az $\{1, 3, 7, 9\}$, a $\{11, 13, 17, 19\}$, az $\{1, 53, 77, -21\}$ stb.

2. Euler-féle φ-függvény

Hány eleme van $\mathbb{Z}_m^*$-nak? Ennek megválaszolásához vezessünk be egy függvényt:

2.1. Definíció. Egy $n \in \mathbb{N}^+$-ra legyen $\varphi(n)$ az n-hez relatív prímek száma 0 és $n-1$ között, azaz $\varphi(n) := |\{k \in \mathbb{Z} \mid 0 \leq k \leq n-1 \ \wedge \ \text{lnko}(n,k) = 1\}|$. Elnevezés: *Euler-féle φ-függvény* (vagy röviden: *Euler-függvény*). (φ kiejtése: "fí".)

Például $\varphi(10) = 4$, mert 0-tól 9-ig négy olyan szám van, amely relatív prím 10-hez: 1, 3, 7 és 9.

A definícióból következik, hogy $\mathbb{Z}_m^*$ elemszáma éppen $\varphi(m)$, például $|\mathbb{Z}_{10}^*| = |\{\overline{1}, \overline{3}, \overline{7}, \overline{9}\}| = 4 = \varphi(10)$.

Sage-ben a $\varphi(n)$ függvény elnevezése `euler_phi(n)`. Állítsuk elő az első néhány értékét:

```
sage: matrix([[n, euler_phi(n)] for n in range(1, 21)]).transpose()
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20]
[ 1  1  2  2  4  2  6  4  6  4 10  4 12  6  8  8 16  6 18  8]
```

Vagy ábrázolhatjuk is:

```
sage: plot(euler_phi, 1, 100)
```

A $\varphi(n)$ függvény legfontosabb tulajdonságai:

1. $\varphi(1) = 1$ (hiszen $\text{lnko}(1,0) = 1$).
2. Ha $n \geq 2$, akkor $\varphi(n) \leq n-1$ (hiszen $\text{lnko}(n,0) = n \neq 1$).
3. Ha p prím, akkor (és csak akkor) $\varphi(p) = p-1$ (hiszen 1-től $p-1$-ig minden szám relatív prím p-hez). Összevetve az előző tulajdonsággal, ez azt jelenti, hogy $\varphi(n)$ prímszámokra veszi fel a legnagyobb értékeket.
4. Ha p prím, akkor $\varphi(p^k) = p^k - p^{k-1} = (p-1)p^{k-1}$. Például $\varphi(16) = \varphi(2^4) = 2^4 - 2^3 = 8$. (Bizonyítás: p^k-hoz pontosan a p többszörösei nem relatív prímek, tehát az összesen p^k számból $p^k/p = p^{k-1}$-t kell kihagyni.)
5. Ha a és b relatív prím, akkor $\varphi(ab) = \varphi(a) \cdot \varphi(b)$. Például $\varphi(10) = \varphi(2) \cdot \varphi(5) = 1 \cdot 4 = 4$. (A bizonyítás nehéz, ezért mellőzzük.) Fontos: ez a tulajdonság csak relatív prímekre igaz! Például $\varphi(2) \cdot \varphi(4) = 1 \cdot 2 = 2$, de $\varphi(2 \cdot 4) = \varphi(8) = 2^3 - 2^2 = 4 \neq 2$.

Láttunk néhány speciális esetet, de általában hogyan tudjuk kiszámítani $\varphi(n)$-et? Ha ismerjük n prímtényezős felbontását, akkor az utolsó két szabály ad egy egyszerű számítási módszert. Például ha $n = 24 = 2^3 \cdot 3$, akkor, mivel 2^3 és 3 relatív prím, ezért:

$$\varphi(24) = \varphi(2^3) \cdot \varphi(3) = (2^3 - 2^2) \cdot (3-1) = 4 \cdot 2 = 8.$$

Általában is igaz, hogy különböző prímszámok hatványai relatív prímek, ezért a kanonikus alakból kiindulva könnyen felbonthatjuk $\varphi(n)$-et prímhatványok $\varphi()$-jeinek szorzatára, és prímhatványra már van képletünk. Még egy példa:

$$\varphi(4200) = \varphi(2^3 \cdot 3 \cdot 5^2 \cdot 7) = \varphi(2^3) \cdot \varphi(3) \cdot \varphi(5^2) \cdot \varphi(7) = (2^3 - 2^2) \cdot 2 \cdot (5^2 - 5) \cdot 6 = 960.$$

Általában is felírhatjuk a képletet:

2.2. Tétel. *Ha n kanonikus alakja $n = p_1^{n_1} p_2^{n_2} \cdots p_k^{n_k}$, akkor:*

$$\varphi(n) = \left(p_1^{n_1} - p_1^{n_1-1}\right)\left(p_2^{n_2} - p_2^{n_2-1}\right) \cdots \left(p_k^{n_k} - p_k^{n_k-1}\right).$$

Emeljünk ki egy speciális esetet, amely később fontos lesz: ha n két különböző prímszám szorzata, $n = pq$, akkor $\varphi(n) = (p-1)(q-1)$. Például $\varphi(10) = \varphi(2 \cdot 5) = 1 \cdot 4 = 4$.

Megjegyzés: a fentiek alapján $\varphi(n)$-et könnyen ki tudjuk számítani, ha ismerjük n prímtényezős felbontását. Ha viszont nem ismerjük – és, mint látni fogjuk, nagy n-ekre a prímfaktorizáció nagyon nehéz feladat –, akkor $\varphi(n)$-et sem fogjuk tudni kiszámítani. Ez később még kulcsfontosságú lesz.

3. Moduláris hatványozás II.

Az előző jegyzetben (Diszkrét logaritmus) megismerkedtünk a moduláris hatványozással, de csak prímszám modulus esetén. Most általánosítjuk a kérdést tetszőleges modulusra.

Állítsuk elő $\mathbb{Z}_{10}$ minden elemének első néhány hatványát:

```
sage: matrix([[power_mod(a, k, 10) for k in range(20)] for a in range(10)])
[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
[1 2 4 8 6 2 4 8 6 2 4 8 6 2 4 8 6 2 4 8]
[1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7]
[1 4 6 4 6 4 6 4 6 4 6 4 6 4 6 4 6 4 6 4]
[1 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5]
[1 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6]
[1 7 9 3 1 7 9 3 1 7 9 3 1 7 9 3 1 7 9 3]
[1 8 4 2 6 8 4 2 6 8 4 2 6 8 4 2 6 8 4 2]
[1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9 1 9]
```

Hasonlítsuk ezt össze az előző jegyzetben szereplő megállapításokkal, ahol a modulus prímszám volt, és vegyük észre a következőket:

- Itt az $\overline{1}$ nem minden sorban ismétlődik. Bizonyos sorokban periodikusan visszatér, máshol viszont az $\overline{a}^0 = \overline{1}$ az egyetlen hatvány, ami $\overline{1}$.
- Ha nem is az $\overline{1}$-től, de előbb-utóbb minden sorban periodikusan ismétlődnek a hatványok.
- Melyek azok az elemek, amelyek hatványaiban az $\overline{1}$ ismétlődik? Ezek az $\overline{1}$, a $\overline{3}$, a $\overline{7}$ és a $\overline{9}$. Vegyük észre, hogy ezek éppen $\mathbb{Z}_{10}^*$ elemei.
- Ezen elemek *rendje* (azaz az első pozitív kitevő, amelyre a hatvány $\overline{1}$) a táblázatból leolvasható: 1, 2 és 4 (pl. $o(\overline{7}) = 4$). Vegyük észre, hogy ezek éppen a 4 pozitív osztói.
- Akárcsak prímszám modulusra, itt is igaz, hogy $o(\overline{1}) = 1$ és $o(\overline{m-1}) = 2$.

Ahhoz, hogy a fenti megállapítások többségét általánosíthassuk, szükségünk van egy általános tételre a moduláris hatványozásról; olyanra, mint a kis Fermat-tétel, amely prímszám modulusra azt mondta, hogy $a^{p-1} \equiv 1 \pmod{p}$, ha $p \nmid a$. Ezt általánosítjuk tetszőleges modulusra:

3.1. Tétel (Euler–Fermat-tétel)**.** *Ha $m \in \mathbb{N}^+$, $a \in \mathbb{Z}$ és $\operatorname{lnko}(a, m) = 1$, akkor $a^{\varphi(m)} \equiv 1 \pmod{m}$.*

Átfogalmazva maradékosztályokra: $\forall \overline{a} \in \mathbb{Z}_m^* : \overline{a}^{\varphi(m)} = \overline{1}$.

Általában tehát nem a p–1-edik hatvány lesz 1, hanem a $\varphi(m)$-edik (prímszámokra a kettő egybeesik: $\varphi(p) = p-1$). Például ha $m = 10$, akkor $\varphi(10) = 4$, ezért $3^4 \equiv 1 \pmod{10}$, $7^4 \equiv 1 \pmod{10}$ stb.

Nagyon fontos, hogy ez csak akkor igaz, ha a és m relatív prímek. Ha nem azok, akkor nemcsak a $\varphi(m)$-edik hatvány nem lesz 1, de semelyik pozitív kitevőjű sem. Miért nem? Nézzük meg a fenti táblázatot: a páros számok minden hatványa páros, és az 5 minden hatványa 5 (-tel osztható). Ez általában is igaz: könnyen belátható, hogy $\overline{a} \in \mathbb{Z}_m$ minden pozitív hatványa osztható $\operatorname{lnko}(a, m)$-mel. Ha tehát $\operatorname{lnko}(a, m) \neq 1$, azaz nem relatív prímek, akkor az $\overline{1}$ biztosan nem szerepelhet a hatványok között. Ha viszont relatív prímek, akkor az Euler–Fermat-tétel garantálja, hogy szerepel az $\overline{1}$.

A tételből az is következik, hogy az invertálható elemek rendje mindig osztója $\varphi(m)$-nek. Például $m = 10$ esetén a rend 1, 2 vagy 4 lehet (lásd fent), és valóban, ezek mind osztói $\varphi(10) = 4$-nek. (Ezzel általánosítottuk az előző jegyzetben prímszám modulusra kimondott Lagrange-tételt.)

Az Euler–Fermat-tétellel – bizonyos feltételek esetén – gyorsíthatjuk a moduláris hatványozást. Például számítsuk ki 137^{75} mod 10-et. Az előző jegyzetben megismertük a gyorshatványozást, amely itt is működik, de mielőtt ráeresztenénk a problémára, előbb próbáljuk meg egyszerűsíteni a számokat. Először is, természetesen az alapnak vehetjük a maradékát 10 szerint: $137 \equiv 7 \pmod{10}$, ezért $137^{75} \equiv 7^{75}$ (10). Így máris nem 137-tel kell szorozgatni, csak 7-tel. Most egyszerűsítsük a kitevőt. De vigyázat, ugyanez nem működik a kitevővel: $7^{75} \not\equiv 7^5$ (10)! Ehelyett viszont használhatjuk az Euler–Fermat-tételt, amely azt mondja, hogy $7^{\varphi(10)} \equiv 1$ (10). Ezért osszuk el maradékosan 75-öt, de ne 10-zel, hanem $\varphi(10) = 4$-gyel: $75 = 18 \cdot 4 + 3$, ezért $7^{75} \equiv 7^{18 \cdot 4 + 3} \equiv \left(7^4\right)^{18} 7^3 \equiv 1^{18} \cdot 7^3 \equiv 7^3$ (10). Vagyis 137^{75} helyett csak 7^3-t kell kiszámítani, amely sokkal egyszerűbb (gyorshatványozással is).
Ennek az egyszerűsítésnek azonban két feltétele van: egyrészt az Euler–Fermat-tételhez az kell, hogy az alap relatív prím legyen a modulushoz (amely itt teljesült: $\text{lnko}(7, 10) = 1$), másrészt pedig hogy ki tudjuk számítani $\varphi(m)$-et az m modulusra. Ez utóbbihoz szükség van az m prímtényezős felbontására, amelyet csak akkor tudunk kiszámítani, ha m nem túl nagy, vagy speciális alakú.
Összegezve tehát: ha az m modulusra ismerjük $\varphi(m)$-et, és $\text{lnko}(a, m) = 1$, akkor az $a^n \bmod m$ hatványt úgy számíthatjuk ki, hogy az alapot vesszük modulo m, a kitevőt pedig modulo $\varphi(m)$: $a^n \equiv (a \bmod m)^{n \bmod \varphi(m)} \pmod{m}$, és ebből az alakból hajtjuk végre a gyorshatványozást.
És mi a helyzet, ha az alap nem relatív prím a modulushoz? Például nézzük 138^{75} mod 10-et. Az első lépés továbbra is működik, azaz $138^{75} \equiv 8^{75}$ (10). Az Euler–Fermat-tételt viszont közvetlenül nem használhatjuk, mert $\text{lnko}(8, 10) \neq 1$. Az ötlet az, hogy bontsuk fel a modulust két részre: $10 = 5 \cdot 2$, és ennek megfelelően a feladatot is: 8^{75} mod 10 helyett számítsuk ki 8^{75} mod 5-öt és 8^{75} mod 2-t. Ez azért jó, mert az első felére már teljesül, hogy $\text{lnko}(8, 5) = 1$, ezért azt meg tudjuk oldani az Euler–Fermat-tétellel: $8^{75} \equiv 3^{75} \equiv 3^{75 \bmod 4} \equiv 3^3 \equiv 2$ (5), a második fele pedig triviális, hiszen $8^{75} \equiv 0^{75} \equiv 0$ (2). A két részeredményből ($138^{75} \equiv 2$ (5) és $138^{75} \equiv 0$ (2)) a kínai maradéktétellel tudjuk kiszámítani a végeredményt (lásd a Moduláris aritmetika jegyzetet).

4. Az RSA-eljárás

Az *RSA-eljárás* egy gyakran használt titkosítási (kriptográfiai) módszer. A betűk az alkotók neveinek kezdőbetűit jelentik (Rivest, Shamir és Adleman).

4.1. Aszimmetrikus titkosítás

Az RSA egy *aszimmetrikus titkosítás*, ami azt jelenti, hogy két különböző kulcsot használ: az egyiket titkosításra, a másikat pedig a titkosított üzenet megfejtésére. Ezzel szemben egy *szimmetrikus titkosítás* (ilyen például a gyakran használt AES) ugyanazt a kulcsot használja mind a titkosításhoz, mind a megfejtéshez. A szimmetrikus titkosítások előnye az egyszerűségük (általában sokkal hatékonyabbak, mint az aszimmetrikus eljárások), viszont hátrányuk, hogy mindkét félnek ismerni kell ugyanazt a (titkos) kulcsot. Ehhez vagy találkozniuk kell előre titokban, hogy megbeszéljék a közös kulcsot, vagy alkalmazniuk kell egy olyan eljárást, mint a Diffie–Hellman-kulcscsere (lásd az előző jegyzetet: Diszkrét logaritmus).
Az aszimmetrikus titkosításoknál viszont, mint amilyen az RSA, két kulcs van: egy *titkos kulcs* és egy *nyilvános kulcs*, amelyek közül elég az előbbit titokban tartani, az utóbbit nyugodtan nyilvánosságra

lehet hozni. Tehát a két félnek nem kell előre megbeszélniük egy közös kulcsot, hanem az egyik fél közzéteszi a nyilvános kulcsát, a másik fél pedig azt használva titkosítja az üzenetet, amit az első fél a titkos kulcsával meg tud fejteni. Vagyis aszimmetrikus titkosításnál bárki tud titkosított üzenetet küldeni egy címzettnek, de azt csak a címzett tudja megfejteni.
Ahhoz, hogy egy aszimmetrikus titkosítás működjön, egyrészt az kell, hogy a két kulcs függjön egymástól, hiszen amit az egyikkel titkosítunk, azt a másikkal meg kell tudnunk fejteni; másrészt viszont az is kell, hogy a nyilvános kulcsból ne lehessen kiszámítani a titkos kulcsot, hiszen akkor az egésznek nem lenne semmi értelme. Ez a két feltétel látszólag ellentmond egymásnak. Ráadásul ha nem lehet az egyikből kiszámítani a másikat, akkor hogyan tudjuk őket egyáltalán előállítani?

4.2. Az RSA működése

Az RSA-eljárás azon alapul, hogy a prímfaktorizáció, azaz a prímtényezős felbontás kiszámítása nagy számokra nagyon nehéz feladat. A tudomány mai állása szerint nincs olyan hatékony algoritmus, amellyel faktorizálni tudnánk egy kellően nagy számot. Azaz ha van két nagy prímszámunk, p és q, azokat bár könnyen össze tudjuk szorozni: $n := p \cdot q$, de az n-ből senki nem fogja tudni kiszámítani p-t és q-t. (Legalábbis ha azok elég nagyok, és bizonyos tulajdonságoknak megfelelnek, amelyeket itt nem részletezünk.) Megjegyzés: ez hasonlít az előző jegyzetben megismert diszkrét logaritmushoz, ahol szintén volt egy művelet, a moduláris hatványozás, amelyet könnyű elvégezni, viszont a fordítottját, a diszkrét logaritmust nehéz. Az ilyen műveletet *egyirányú függvény*nek nevezzük.
Az RSA alapvető művelete, akárcsak a Diffie–Hellman-kulcscserének, a moduláris hatványozás, de ezúttal nem prímszám a modulus, hanem két prímszám szorzata: $n = p \cdot q$. Mind a titkosítás, mind a megfejtés egy hatványozás az n modulus szerint, csak különböző kitevőkkel: ha a bemenő üzenet m, akkor a titkosítás során kiszámítjuk a $c \equiv m^e \pmod{n}$ értéket, ahol c a titkosított üzenet és e a *nyilvános kitevő*, a megfejtés során pedig m-et visszaállítjuk az $m \equiv c^d \pmod{n}$ hatványozással, ahol d a *titkos kitevő*. A kérdés persze az, hogy mi legyen e és d, hogy ez így működjön.
Ahhoz, hogy a megfejtéskor ugyanazt az m-et kapjuk vissza, mint amelyet titkosítottunk, az kell, hogy $m \equiv c^d \equiv (m^e)^d \equiv m^{ed} \pmod{n}$ legyen. Az Euler–Fermat-tételből tudjuk, hogy a kitevőnek vehetjük a maradékát modulo $\varphi(n)$, azaz $m^{ed} \equiv m^{ed \bmod \varphi(n)} \pmod{n}$. Tehát ha e és d olyan, hogy $ed \equiv 1 \pmod{\varphi(n)}$ – azaz e és d egymás inverze modulo $\varphi(n)$ –, akkor éppen $m^{ed} \equiv m \pmod{n}$ lesz, ahogy szeretnénk. (Megjegyzés: ha $\text{lnko}(m, n) \neq 1$, akkor az Euler–Fermat-tételt nem használhatjuk, de más módszerrel akkor is bebizonyítható, hogy $m^{ed} \equiv m \pmod{n}$.)
A nyilvános kitevőből, e-ből csak akkor tudjuk kiszámítani a titkos kitevőt, d-t, ha ismerjük $\varphi(n)$-et. De ahhoz, hogy kiszámítsuk $\varphi(n)$-et, szükségünk van az n prímtényezős felbontására, azaz p-re és q-ra. Tehát valóban, az egész RSA-eljárás biztonságát az adja, hogy a prímfaktorizáció nehéz.
Az RSA első lépése a *kulcsgenerálás*:

1. Generáljunk két véletlen, titkos prímszámot: p-t és q-t.
2. Számítsuk ki a szorzatukat: $n := pq$.
3. Számítsuk ki $\varphi(n)$-et: $\varphi(n) = \varphi(pq) = \varphi(p)\varphi(q) = (p-1)(q-1)$.
4. Válasszunk egy nyilvános kitevőt, e-t úgy, hogy $2 < e < \varphi(n)$ és $\text{lnko}(e, \varphi(n)) = 1$.
5. Oldjuk meg az $ed \equiv 1 \pmod{\varphi(n)}$ kongruenciát d-re, azaz számítsuk ki e moduláris inverzét modulo $\varphi(n)$. (Használhatjuk a bővített euklideszi algoritmust.)

6. Nyilvános kulcs: az (n, e) pár. Ezt közzétehetjük.
7. Titkos kulcs: az (n, d) pár. Ezt tartsuk titokban.

(Megjegyzés: nemcsak d, hanem p, q és $\varphi(n)$ is titkos értékek, de azokra a továbbiakban nem lesz szükség, így törölhetjük azokat.)

Ezután az RSA-val így lehet titkosítani:

1. Írjuk fel a titkosítandó üzenetet egy m egész számként 0 és $n-1$ között. (Ha az üzenet ehhez túl hosszú, akkor bontsuk fel kisebb részekre, és külön-külön titkosítsuk azokat.)
2. Titkosítás: számítsuk ki a $c \equiv m^e \pmod{n}$ hatványt az (n, e) nyilvános kulcsból.
3. Megfejtés: számítsuk ki az $m \equiv c^d \pmod{n}$ hatványt az (n, d) titkos kulcsból.

Akárcsak a Diffie–Hellman-kulcscserénél, itt is nagyon fontos, hogy a két prímszámot, p-t és q-t biztonságos véletlenszám-generátorral állítsuk elő, hogy kívülről ne lehessen őket megjósolni. Szintén nagyon fontos, hogy olyan nagyok legyenek (és megfelelő tulajdonságúak), hogy n-et ne lehessen felbontani p-re és q-ra. Manapság jellemző a 2048-bites és a 4096-bites RSA, azaz n hossza kettes számrendszerben 2048, ill. 4096 bit (tehát p és q hossza egyenként kb. 1024, ill. 2048 bit).

És hogyan állítjuk elő e-t, a nyilvános kitevőt? Mivel nyilvános, ezért nem kell véletlennek lennie, viszont érdemes olyan értéket választani, amellyel hatékonyan lehet számolni. Gyakran használják az $e = 2^{16} + 1 = 65537$ értéket, amely elég egyszerű ahhoz, hogy könnyű legyen vele számolni (gyorshatványozással), de elég összetett ahhoz, hogy ne veszélyeztesse az RSA biztonságát.

4.3. Digitális aláírás

Mint láttuk, az RSA-val úgy lehet titkosítani, hogy az üzenetet a nyilvános kulccsal titkosítjuk, és a titkos kulccsal fejtjük meg. Ez azt jelenti, hogy bárki tud titkos üzenetet küldeni egy címzettnek, de csak a címzett, azaz a titkos kulcs birtokosa tudja megfejteni azt.

De mi történik, ha ezt megfordítjuk? Mi történik, ha a titkos kulccsal titkosítjuk, és a nyilvános kulccsal fejtjük meg az üzenetet? Ekkor csak a titkos kulcs birtokosa tud titkosítani, de bárki meg tudja fejteni az üzenetet. Miért jó ez?

Ezt úgy hívják, hogy *digitális aláírás* (vagy *hitelesítés*), amely a titkosításon kívül az RSA egy másik fontos alkalmazása. Ilyenkor az m üzenetet a titkos kulccsal "titkosítjuk" (kódoljuk), azaz előállítjuk a $c \equiv m^d \pmod{n}$ értéket, és ezt továbbítjuk m-mel együtt. Ezután az üzenet címzettje a nyilvános kulcs segítségével vissza tudja állítani c-ből m-et: $m \equiv c^e \pmod{n}$, és ha ez sikerült, akkor biztos lehet benne, hogy az üzenet valóban a feladótól származik, hiszen (jó esetben) csak ő rendelkezik a titkos kulccsal.

4.4. Gyorsítás kínai maradéktétellel

Mint láttuk, a nyilvános kitevőt, e-t megválaszthatjuk úgy, hogy könnyű legyen vele számolni. A titkos kitevőre, d-re viszont már nincs közvetlen befolyásunk, ezért az lehet nagyon nagy is – ami a biztonság szempontjából persze egyáltalán nem baj. Ez viszont azt jelenti, hogy d-ik hatványra emelni (azaz megfejteni az üzenetet) sokkal lassabb, mint e-ikre (azaz titkosítani).

Lehet azonban ezt gyorsítani, ha nem közvetlenül számítjuk ki az $m \equiv c^d \pmod{n}$ hatványt, hanem két részletben, azaz az $n = pq$ modulus helyett külön-külön p-re és q-ra: $m_p \equiv c^d \pmod{p}$ és $m_q \equiv c^d \pmod{q}$. A két eredményt a kínai maradéktétellel tudjuk egyesíteni (lásd a Moduláris

aritmetika jegyzetet). Ez a felbontás azért jó, mert így nemcsak kisebb számokkal kell számolni (a kisebb modulusok miatt), de a kitevő hosszát is csökkenthetjük: a kis Fermat-tétel miatt elég a $d_p := d \bmod (p-1)$, ill. a $d_q := d \bmod (q-1)$ kitevőket használni. Így bár egy helyett két moduláris hatványozást végzünk, de azok egyenként több mint négyszer olyan gyorsak: a kitevők is fele olyan hosszúak, tehát fele annyi szorzás kell, és a szorzandó számok is fele akkorák.
A gyorsított $m \equiv c^d \pmod{n}$ hatványozás tehát így történik:

1. Számítsuk ki előre a következő értékeket:
   - $d_p := d \bmod (p-1)$,
   - $d_q := d \bmod (q-1)$,
   - q inverzét (q^{-1}) modulo p.
2. $m_p := c^{d_p} \bmod p$.
3. $m_q := c^{d_q} \bmod q$.
4. A kínai maradéktétellel számítsuk ki m_p-ből és m_q-ból m-et:
   - $m_d := q^{-1}(m_p - m_q) \bmod p$.
   - $m := m_q + m_d q$.

A fenti gyorsítás miatt gyakran nem is az (n, d) párt szokták titkos kulcsként tárolni, hanem a (p, q, d_p, d_q, q^{-1}) ötöst.

5. Prímtesztek

Az RSA-val kapcsolatban egy fontos részletről még nem volt szó: hogyan generáljuk a két véletlen prímszámot, p-t és q-t? Biztonságos véletlenszám-generátorokról már volt szó, de honnan tudjuk, hogy az eredmény prímszám-e? Hiszen az RSA éppen azon alapul, hogy egy nagy számot nem tudunk faktorizálni – de ha nem tudjuk az osztóit megtalálni, akkor hogyan ellenőrizzük, hogy prímszám-e? Szerencsére eldönteni egy számról, hogy prímszám-e, sokkal könnyebb feladat, mint prímtényezőkre bontani. Az ilyen módszereket *prímteszt*eknek nevezzük. Kétféle prímteszttel foglalkozunk:

1. A *determinisztikus prímteszt*ek biztosan meg tudják mondani, hogy egy szám prímszám-e.
2. A *valószínűségi prímteszt*ek nem 100%-ban adnak helyes választ, néha hibáznak. A tipikus tesztek azonban csak az egyik irányban tévednek: prímszámokra biztosan helyen választ adnak, de összetett számokat néha tévesen prímszámnak jeleznek.

De miért is használnánk olyan teszteket, amelyek nem mindig mondanak igazat? Azért, mert a valószínűségi prímtesztek általában sokkal hatékonyabbak, mint a determinisztikusak. Például a prímszámokról szóló jegyzetben (Oszthatóság, prímszámok) leírt próbaosztásos módszer (azaz hogy 2-től $\sqrt{n}$-ig megnézzük az összes számot, hogy osztja-e) egy determinisztikus prímteszt, de nagy számokra nagyon lassú (hiszen lényegében olyan, mint a prímfaktorizáció). Az alábbiakban leírt módszerek viszont hatékonyak, és elég ritkán tévednek ahhoz, hogy ez elfogadható legyen.

5.1. Fermat-prímteszt

A kis Fermat-tétel csak prímszámokról szól: ha p prímszám, és $p \nmid a$, akkor $a^{p-1} \equiv 1 \pmod{p}$. De vajon igaz-e ez az állítás, ha p nem prímszám?
Írassuk ki az $a^{n-1} \bmod n$ értékeket különböző n-ekre és különböző a-kra (ahol $1 \leq a \leq n-1$):

```
sage: for n in range(9, 16):
....:     print(n, [power_mod(a, n-1, n) for a in range(1, n)])
9 [1, 4, 0, 7, 7, 0, 4, 1]
10 [1, 2, 3, 4, 5, 6, 7, 8, 9]
11 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
12 [1, 8, 3, 4, 5, 0, 7, 8, 9, 4, 11]
13 [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
14 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
15 [1, 4, 9, 1, 10, 6, 4, 4, 6, 10, 1, 9, 4, 1]
```

Vegyük észre a következőket:

1. Ha n prímszám, akkor minden érték 1, ahogy azt a kis Fermat-tétel garantálja.
2. Ha $a = 1$ (első érték), akkor annak minden hatványa 1.
3. Ha $a = n-1$ (utolsó érték), akkor is gyakran kapunk 1-et, hiszen $n-1 \equiv -1 \pmod{n}$, és -1 minden második hatványa 1.
4. Az összes többi esetben azonban, azaz ha n összetett szám és $2 \leq a \leq n-2$, szinte soha nem kapunk 1-et.
5. De néha igen: például $n = 15$-re és $a = 4$-re $4^{14} \equiv 1 \pmod{15}$.

Adja magát az ötlet, hogy ha egy n-ről szeretnénk eldönteni, hogy prímszám-e, akkor valamilyen a-ra számítsuk ki a^{n-1} mod n-et, és ha 1-től különböző az eredmény, akkor n biztosan nem prímszám, ha pedig 1, akkor *valószínűleg* prímszám.

Ez tehát egy valószínűségi prímteszt. Úgy hívják, hogy *Fermat-prímteszt*, és a következőképpen működik:

1. Bemenet: $n \in \mathbb{N}, n \geq 4$, a tesztelendő szám.
2. Válasszunk egy véletlen $a \in \{2, 3, \ldots, n-2\}$ alapot.
3. Számítsuk ki a^{n-1} mod n-et.
4. Ha ez nem 1, akkor az eredmény "hamis" (azaz n nem prímszám).
5. Ha ez 1, akkor az eredmény "igaz" (azaz n valószínűleg prímszám).

Ez az algoritmus nagyon hatékony, hiszen csak egy moduláris hatványozást kell elvégezni, amely gyorshatványozással megoldható. (Összehasonlítva a próbaosztásos módszerrel, ahhoz kb. $\sqrt{n}$ osztás kell, míg a Fermat-prímteszthez csak kb. $2\log_2 n$ szorzás (lásd a gyorshatványozást). Például ha $n \approx 1000000$, akkor az előbbi kb. 1000, míg az utóbbi csak kb. 40.)

A tévedés valószínűségét csökkenthetjük, ha többször futtatjuk le a tesztet, azaz több véletlen a-ra is kiszámítjuk a hatványt, és csak akkor fogadjuk el a számot prímszámnak, ha minden teszt igazat ad. Például ha $n = 15$-öt csak $a = 4$-gyel tesztelnénk, akkor n átmenne a teszten, hogy "prímszám". De ha megismételjük a tesztet pl. $a = 2$-vel, akkor n már "lebukik", hogy összetett szám.

5.1. Definíció. Egy adott $a \in \mathbb{N}^+$ számra az n összetett számot *Fermat-álprím*nek nevezzük, ha $a^{n-1} \equiv 1 \pmod{n}$. (Néha egyszerűen csak *álprím*nek hívják, jelző nélkül.)

Vagyis a Fermat-álprímek azok, amelyek "becsapják" a Fermat-prímtesztet. Például $n = 15$ egy Fermat-álprím $a = 4$-re. (Megjegyzés: ez a fogalom függ az alaptól, a-tól.)

Szerencsére azonban a Fermat-álprímek ritkák. Például $a = 2$-re 1-től 10000-ig csak 22 Fermat-álprím van (a legkisebb a 341) – miközben ugyanezen intervallumban 1229 prímszám van.

Vannak viszont olyan álprímek, amelyek nagyon makacsul ellenállnak a Fermat-prímtesztnek:

5.2. Definíció. Egy n összetett szám *Carmichael-szám*, ha minden $a \in \mathbb{N}$ alapra, amely relatív prím n-hez, $a^{n-1} \equiv 1 \pmod{n}$.

Vagyis a Carmichael-számok azok, amelyek a lehető legtöbb alapra Fermat-álprímek. (Ennél több alapra nem is lehetnének, hiszen ha a és n nem relatív prím, akkor a hatvány biztosan nem lehet 1.) A legkisebb Carmichael-szám az 561. Ezek szerencsére még ritkábbak, mint a közönséges Fermat-álprímek (1-től 10000-ig csak 7 Carmichael-szám van).

A jó hír az, hogy ha egy Fermat-álprím nem Carmichael-szám, akkor viszonylag könnyű "lebuktatni":

5.3. Tétel. *Ha n összetett szám, de nem Carmichael-szám, akkor az 1 és $n-1$ közötti a-k több mint felére $a^{n-1} \not\equiv 1 \pmod{n}$.*

Ezek szerint tehát a Fermat-prímteszttel kapcsolatban három eset lehetséges:

1. Ha n prímszám, akkor a teszt biztosan helyes eredményt ad.
2. Ha n Carmichael-szám (ami ritka), akkor a teszt nagyon megbízhatatlan.
3. Egyébként viszont a teszt több mint 50% valószínűséggel helyes eredményt ad. Ez így első ránézésre kevésnek tűnik, de ez csak egy pesszimista alsó korlát, és ha ismételjük a tesztet különböző a-kra, akkor ez a korlát is tovább csökkenthető: k futtatás után a hiba valószínűsége $< 1/2^k$, pl. $k = 10$ esetén kevesebb, mint 0,001.

5.2. Miller–Rabin-prímteszt

Valós számokra tudjuk, hogy az $x^2 = 1$ egyenletnek csak két megoldása van: $x = 1$ és $x = -1$. A következő tétel azt mondja, hogy ez bizonyos esetekben moduláris aritmetikára is igaz:

5.4. Tétel. *Ha p prímszám, és $x^2 \equiv 1 \pmod{p}$, akkor $x \equiv 1 \pmod{p}$ vagy $x \equiv -1 \pmod{p}$.*

(Bizonyítás: a feltétel szerint $p \mid x^2 - 1 = (x-1)(x+1)$, de prímszám egy szorzatot csak úgy oszthat, ha valamelyik tényezőjét is osztja, azaz $p \mid (x-1)$ vagy $p \mid (x+1)$.)

Például ha a modulus $p = 5$, akkor a $0, 1, 2, 3, 4$ elemek moduláris négyzete rendre $0, 1, 4, 4, 1$, azaz az $x^2 \equiv 1\ (5)$ kongruenciának valóban csak az $x \equiv 1\ (5)$ és az $x \equiv 4 \equiv -1\ (5)$ megoldásai vannak. Összetett számokra viszont nem feltétlenül igaz a tétel, például $3^2 \equiv 1\ (8)$.

Induljunk ki ismét a kis Fermat-tételből, azaz hogy egy p prímszámra $a^{p-1} \equiv 1\ (p)$ (ha $p \nmid a$). Például ha $p = 29$, akkor $a^{28} \equiv 1\ (29)$. Ha alkalmazzuk erre a fenti tételt $x = a^{14}$-re (ekkor $x^2 = a^{28}$), azaz felezzük a kitevőt, akkor azt kapjuk, hogy $a^{14} \equiv 1\ (29)$ vagy $a^{14} \equiv -1\ (29)$. Ha $a^{14} \equiv 1\ (29)$, akkor még egyszer felezhetjük a kitevőt, és a tételből azt kapjuk, hogy $a^7 \equiv 1\ (29)$ vagy $a^7 \equiv -1\ (29)$. Itt viszont a kitevő páratlan, ezért nem tudjuk ezt tovább folytatni.

A fenti műveletsor minden lépéséhez felhasználtuk, hogy p prímszám. Ha nem az, akkor könnyen lehet (de nem biztos), hogy valamelyik lépés nem fog működni. Ezzel tehát kaptunk egy újabb valószínűségi prímtesztet. Például ha $n = 15$ és $a = 4$, akkor $a^{n-1} \equiv 1\ (n)$, azaz $4^{14} \equiv 1\ (15)$ még teljesül (ahogy azt a Fermat-prímtesztnél láttuk), viszont ha felezzük a kitevőt, akkor $4^7 \equiv 4\ (15)$, de $4 \neq 1$ és $4 \neq -1$; ebből tehát következik, hogy a 15 nem lehet prímszám.

A *Miller–Rabin-prímteszt* ezen az elven működik, csak a műveleteket fordított sorrendben hajtja végre: nem az a^{n-1}-t számítja ki először, és felezgeti a kitevőt, hanem először kiszámítja az utolsó, már nem felezhető kitevőre a hatványt (pl. a fenti $p = 29$-re a^7-t), és onnan duplázgatja a kitevőt $n-1$-ig. Ennek az az értelme, hogy a kisebb hatványt könnyebb kiszámítani, és nem mindig kell elmenni $n-1$-ig, ha hamarabb is eldőlt az eredmény (pl. "lebukott" az összetett szám).

A Miller–Rabin-prímteszt a következőképpen működik:

1. Bemenet: $n \in \mathbb{N}, n \geq 4$, a tesztelendő szám.
2. Ha n páros, akkor az eredmény "hamis" (azaz n nem prímszám).
3. Válasszunk egy véletlen $a \in \{2, 3, \ldots, n-2\}$ számot.
4. Írjuk fel $n-1$-et $n-1 = 2^k d$ alakban, ahol d páratlan (azaz felezgessük $d := n-1$-et addig, amíg páratlan nem lesz az eredmény, és közben számoljuk a felezéseket k-val).
5. $x := a^d \bmod n$.
6. Ha $x = 1$, akkor az eredmény "igaz" (azaz n valószínűleg prímszám).
7. Ismételjük k-szor:
   - Ha $x = n-1$ (azaz $x \equiv -1\ (n)$), akkor az eredmény "igaz" (azaz n valószínűleg prímszám).
   - $x := x^2 \bmod n$.
   - Ha $x = 1$, akkor az eredmény "hamis" (azaz n nem prímszám).
8. Az eredmény "hamis" (azaz n nem prímszám).

Az algoritmus addig duplázza a kitevőt, amíg 1 vagy -1 nem lesz a hatvány, vagy el nem érjük az $n-1$-es kitevőt. Ha -1-et kapunk, akkor nem kell folytatni, mert annak a négyzete 1, és onnantól kezdve mindig 1-et kapunk, vagyis sikerült a teszt. Ha pedig 1-et kapunk, és nem a legelején, az csak úgy lehet, ha az előző érték nem 1 és nem -1 volt (hiszen akkor már megálltunk volna), vagyis a teszt "lebuktatta" az összetett számot. Ha pedig elértük az $n-1$-es kitevőt (azaz k-szor dupláztunk, ahol $n-1 = 2^k d$), és még mindig nem kaptunk 1-et, akkor ezzel megsértettük a kis Fermat-tételt, vagyis n biztosan nem lehet prímszám.

5.5. Definíció. Egy adott $a \in \mathbb{N}^+$ számra az n összetett számot *erős álprím*nek nevezzük, ha a Miller–Rabin-prímteszt elfogadja n-et prímszámnak az adott a-ra.

Mivel a Miller–Rabin-prímteszt a Fermat-prímteszt kiterjesztése, ezért az erős álprímek egyúttal Fermat-álprímek is, de ez fordítva nem igaz (például $n = 15$ Fermat-álprím $a = 4$-re, de nem erős álprím, ahogy azt a fenti példából láthattuk).

Hogy mennyivel jobb a Miller–Rabin-prímteszt, azt a következő tétel mutatja:

5.6. Tétel. *Egy n összetett szám az 1 és $n-1$ közötti a-k legfeljebb negyedére lehet erős álprím.*

Ez egyrészt azt jelenti, hogy itt nincsenek olyan számok, mint a Fermat-prímtesztnél a Carmichael-számok, azaz amelyekre a Miller–Rabin-prímteszt majdnem mindig rossz eredményt adna. Másrészt pedig ez a teszt nem 50%-ban, hanem legalább 75%-ban ad helyes választ összetett számokra – és persze ez is tovább javítható, ha ismételjük a tesztet k-szor, mert akkor a hiba valószínűsége $< 1/4^k$.

A Miller–Rabin-prímteszt további előnye, hogy hatékonyabb, mint a Fermat-prímteszt, mert nem mindig számítja ki a teljes a^{n-1} hatványt, hanem gyakran hamarabb is megállhat.

6. Szerző

Ezt a jegyzetet Uray M. János írta, részben felhasználva Nagy Ádám jegyzetét:
https://compalg.elte.gitlab-pages.hu/dimoa-web/tematika/dimoa
Hibákat, észrevételeket itt lehet jelezni: uray.janos@inf.elte.hu.