Diszkrét matematika
11. előadás

MÁRTON Gyöngyvér
mgyongyi@ms.sapientia.ro

Sapientia Egyetem,
Matematika-Informatika Tanszék
Marosvásárhely, Románia

2022, őszi félév

Miről volt szó az elmúlt előadáson?

- hatványok, generátor elemek, biztonságos prímek,
- a diszkrét logaritmus (DL) probléma,
- a Diffie-Hellman kulcscsere,
- a kiterjesztett eukleidészi algoritmus,
- lineáris kongruenciák,
- moduláris inverz,
- az RSA rendszer, a *baby*-RSA rendszer,

Miről lesz szó?

- a kínai maradéktétel
- az RSA és a kínai maradéktétel
- másodfokú kongruenciák, kvadratikus maradékok
- a Legendre és Jacobi szimbólumok,
- a Rabin rendszer: titkosító, digitális aláírás rendszer

A kínai maradéktétel

- több kongruenciából álló egyismeretlenes szimultán kongruenciarendszer megoldását adja meg
- kínai matematikusok több mint 2000 éve ismerik a megoldást
- lehetővé teszi hogy a nagy számokkal szükséges számításokat kis számokkal végezhető műveletekre vezessünk vissza

Feladat: Ha egy tojásokkal teli kosárból kivesszük a tojásokat 2, 3, 4, 5, majd 6 -osával, akkor rendre 1, 2, 3, 4, 5 tojás marad mindig a kosárban. Ha 7-esével vesszük ki nem marad egy tojás sem. Hány tojás van a kosárban?

A feladat az alábbi kongruencia-rendszerrel modellezhető:

$$\begin{aligned} x &\equiv 1 \pmod 2 \\ x &\equiv 2 \pmod 3 \\ x &\equiv 3 \pmod 4 \\ x &\equiv 4 \pmod 5 \\ x &\equiv 5 \pmod 6 \\ x &\equiv 0 \pmod 7 \end{aligned}$$

A kongruencia-rendszer a kínai maradéktétellel oldható meg.

A kínai maradéktétel

1. tétel

Legyenek $m_1, m_2, ..., m_r$ pozitív, páronként relatív prímek. Ekkor az

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_r \pmod{m_r} \end{aligned}$$

kongruencia-rendszernek $M = m_1 \cdot m_2 \cdot ... \cdot m_r$ modulus szerint egy megoldása van.

A megoldás meghatározásának menete:

- meghatározzuk: $M_k = M/m_k = m_1 \cdot m_2 \cdot ... \cdot m_{k-1} \cdot m_{k+1} \cdot ... \cdot m_r$,
- meghatározzuk az M_k értékek inverzét $\pmod{m_k}$ szerint, jelöljük ezeket $\hat{M}_k$-val,
- $x = a_1 \cdot M_1 \cdot \hat{M}_1 + a_2 \cdot M_2 \cdot \hat{M}_2 + ... + a_r \cdot M_r \cdot \hat{M}_r$ lesz a rendszer megoldása.

A kínai maradéktétel

A tojásos feladat az alábbi kongruencia-rendszerre vezethető vissza:

$$\begin{aligned} x &\equiv 4 \pmod 5 \\ x &\equiv 11 \pmod{12} \\ x &\equiv 0 \pmod 7 \end{aligned}$$

mert

- az $x \equiv 3 \pmod 4$ megoldásai kielégítik az $x \equiv 1 \pmod 2$ megoldásait,
- az $x \equiv 5 \pmod 6$ megoldásai kielégítik az $x \equiv 2 \pmod 3$ megoldásait,
- az $x \equiv 11 \pmod{12}$ megoldásai kielégítik az $x \equiv 3 \pmod 4$ és $x \equiv 5 \pmod 6$ megoldásait.
- A megoldás menete:
  - $M = m_1 \cdot m_2 \cdot m_3 = 5 \cdot 12 \cdot 7 = 420$,
  - $M_1 = 84 \equiv 4 \pmod 5$ amelynek inverze 4, fennáll: $4 \cdot 4 = 1 \pmod 5$,
  - $M_2 = 35 \equiv 11 \pmod{12}$ amelynek inverze 11, fennáll: $11 \cdot 11 = 1 \pmod{12}$,
  - $M_3 = 60 \equiv 4 \pmod 7$ amelynek inverze 2, fennáll: $4 \cdot 2 = 1 \pmod 7$,
  - a rendszer megoldása: $x = 4 \cdot 84 \cdot 4 + 11 \cdot 35 \cdot 11 + 0 \cdot 60 \cdot 2 = 119 \pmod{420}$.

A kínai maradéktétel

1. feladat

Írjunk egy Python függvényt, amely meghatározza a következő r egyenletből álló kongruencia-rendszer megoldását, feltételezve hogy az $m_1, m_2, \ldots, m_r$ pozitív egész számok páronként NEM relatív prímek.

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_r \pmod{m_r} \end{aligned}$$

Az algoritmus lépései:

- meghatározzuk az első két egyenlet megoldását:
  - kiterjesztett Eukleidészi algoritmussal meghatározzuk, azokat a g, k_1, k_2 értékeket, amelyekre fennáll, hogy: $g = k_1 \cdot m_1 + k_2 \cdot m_2$,
  - ha g nem osztja $a_1 - a_2$-t, akkor a kongruencia rendszernek nincs megoldása,
  - lcm-el jelölve az m_1, m_2 legkisebb közös többszörösét, a megoldást pedig a-val, akkor: $a = ((a_1 \cdot m_2 \cdot k_2)//g + (a_2 \cdot m_1 \cdot k_1)//g) \pmod{lcm}$,
- vesszük a harmadik egyenletet és most meghatározzuk a következő két egyenlet megoldását:

$$\begin{aligned} x &\equiv a \pmod{lcm} \\ x &\equiv a_3 \pmod{m_3} \end{aligned}$$

- vesszük a negyedik egyenletet és a kapott megoldást, majd így tovább, amíg minden egyenletet fel nem dolgoztunk,
- a kongruencia rendszernek a megoldása, az utolsó a lcm érték lesz,
- ha valamelyik pár esetében nincs megoldás, akkor az egyenletrendszernek sincs megoldása.

A kínai maradéktétel

```
from math import gcd
from eload10 import extEuclid

def kinaiMarTetel(aL, mL):
    aL0 = aL[0]
    mL0 = mL[0]
    for i in range(1, len(aL)):
        g, k1, k2 = extEuclid(mL0, mL[i])
        if abs(aL0 - aL[i]) % g != 0:
            print("nincs megoldas")
            return -1
        else:
            lcmV = lcm(mL0, mL[i])
            aL0 = ((aL0 * mL[i] * k2)//g + (aL[i] * mL0 * k1//g)) % lcmV
            mL0 = lcmV
    return aL0

def lcm(a, b):
    return abs(a * b) // gcd(a, b)

>>> kinaiMarTetel([8, 18, 13, 10], [9, 35, 20, 17])
    5093
```

Az RSA, a kínai maradéktétel

- a kínai maradéktétel alkalmazható az RSA-nál, a visszafejtési folyamat gyorsítható vele, mert az n nagyságrendjével megegyező d hatványkitevő helyett két kisebb hatványkitevővel való hatványozást lehet végezni,
- mivel p, q prímszámok, fennáll a következő két összefüggés:

$$\begin{array}{lll} d \equiv dp \pmod{p-1} & \Leftrightarrow & c^d \equiv c^{dp} \pmod{p} \\ d \equiv dq \pmod{q-1} & \Leftrightarrow & c^d \equiv c^{dq} \pmod{q} \end{array}$$

- a következő egyenletrendszert megoldása, pedig a c^d értékét fogja adni, amelyet a kínai maradéktétellel oldhatunk meg:

$$\begin{array}{lll} x & \equiv & c^{dp} \pmod{p} \\ x & \equiv & c^{dq} \pmod{q} \end{array}$$

- meghatározzuk $dp, dq, \hat{M}q, \hat{M}p$ értékeket:

$$\begin{array}{ll} dp = d \pmod{p-1} & dq = d \pmod{q-1} \\ \hat{M}q = inverz(q, p) & \hat{M}p = inverz(p, q) \end{array}$$

- a $c^d \pmod{n}$ értéket megadja az x értéke, ahol

$$\begin{array}{l} x = (\hat{M}q \cdot q \cdot xp + \hat{M}p \cdot p \cdot xq) \pmod{n} \\ xp = c^{dp} \pmod{p} \\ xq = c^{dq} \pmod{q}. \end{array}$$

Az RSA, a kínai maradéktétel

2. feladat

Az alábbi Python függvény a korábban megadott RSA_fel függvényben, a visszafejtő RSA_decrypt függvény helyett a RSA_decryptCR függvényt hívja meg, paraméterként meg kell neki adni a p, q értékeket.

```
from eload10 import RSA_key_gen
def RSA_fel():
    k = int(input('bit meret: '))
    e, d, n, p, q = RSA_key_gen(k)
    print ('nyilvanos kulcs: ', e, n)
    print ('titkos kulcs: ', d, n)
    print ('titkos adatok: ', p, q)

    print('kerek egy szamot, kisebb legyen, mint ', n)
    K = int(input())
    cK = pow(K, e, n)
    print ('titkositott ertek: ', cK)
    K = RSA_decryptCR(cK, d, n, p, q)
    print ('visszafejtett ertek: ', K)
```

Az RSA, a kínai maradéktétel

3. feladat

A visszafejtő RSA_decryptCR függvény

```
def RSA_decryptCR(cK, d, n, p, q):
    dp = d % (p-1)
    dq = d % (q-1)
    Mq = inverz(q, p)
    Mp = inverz(p, q)
    cKp = pow(cK, dp, p)
    cKq = pow(cK, dq, q)
    K = (Mq * q * cKp + Mp * p * cKq) % n
    return K
```

A `RSA_key_gen` függvényt, a 10. előadásban írtuk meg.

Másodfokú kongruenciák, kvadratikus maradékok

Határozzuk meg $\pmod{11}$ szerint a számok négyzetét:

$$\begin{array}{ll} 1^2 = 1 & 6^2 = 3 \\ 2^2 = 4 & 7^2 = 5 \\ 3^2 = 9 & 8^2 = 9 \\ 4^2 = 5 & 9^2 = 4 \\ 5^2 = 3 & 10^2 = 1 \end{array}$$

Vegyük észre, hogy az alábbi kongruenciák megoldhatóak és minden esetben két megoldás van:

$$\begin{array}{lll} x^2 \equiv 1 \pmod{11}, & \text{megoldások: } 1, 10 \\ x^2 \equiv 4 \pmod{11}, & \text{megoldások: } 2, 9 \\ x^2 \equiv 3 \pmod{11}, & \text{megoldások: } 5, 6 \\ x^2 \equiv 5 \pmod{11}, & \text{megoldások: } 4, 7 \\ x^2 \equiv 9 \pmod{11}, & \text{megoldások: } 3, 8 \end{array}$$

Az alábbi kongruenciák NEM oldhatóak meg:

$$\begin{array}{l} x^2 \equiv 2 \pmod{11} \\ x^2 \equiv 6 \pmod{11} \\ x^2 \equiv 7 \pmod{11} \\ x^2 \equiv 8 \pmod{11} \\ x^2 \equiv 10 \pmod{11} \end{array}$$

Másodfokú kongruenciák, kvadratikus maradékok

1. értelmezés

Az a számot kvadratikus maradéknak (négyzetes maradék) nevezzük, ha létezik olyan x amelyre az $x^2 \equiv a \pmod{n}$ kongruencia megoldható, ahol $lnko(a, n) = 1$. Ebben az esetben x-et az a négyzetgyökének hívjuk. Az a számot, ha nem kvadratikus maradék, akkor kvadratikus nemmaradéknak hívjuk.

- az $1, 4, 3, 5, 9$ számok kvadratikus maradékok $\pmod{11}$ szerint,
- 1 négyzetgyöke: $1, 10$,
- 4 négyzetgyöke: $2, 9$,
- 3 négyzetgyöke: $5, 6$,
- 5 négyzetgyöke: $4, 7$,
- 9 négyzetgyöke: $3, 8$,
- a $2, 6, 7, 8, 10$, számok, pedig kvadratikus nemmaradékok,
- $\pmod{11}$ szerint 5 szám kvadratikus maradék van, és szintén 5 szám kvadratikus nemmaradék.

Másodfokú kongruenciák, kvadratikus maradékok

A $\pmod P$ szerinti kongruenciákra kijelenthető, ahol P prímszám:

- ha egy a szám kvadratikus maradék, akkor az $x^2 \equiv a \pmod P$ kongruenciának két inkongruens megoldása van
- $\pmod P$ szerint ugyanannyi szám kvadratikus maradék, mint amennyi kvadratikus nemmaradék, számuk: $\frac{P-1}{2}$
- egy a szám akkor és csakis akkor kvadratikus maradék $\pmod P$ szerint ha: $a^{(P-1)/2} \equiv 1 \pmod P$
- egy a szám akkor és csakis akkor kvadratikus nemmaradék $\pmod P$ szerint ha: $a^{(P-1)/2} \equiv -1 \pmod P$

$$
\begin{array}{ll}
1^5 = 1 & 2^5 = 10 = (-1) \pmod{11} \\
4^5 = 1 & 6^5 = 10 = (-1) \pmod{11} \\
3^5 = 1 & 7^5 = 10 = (-1) \pmod{11} \\
5^5 = 1 & 8^5 = 10 = (-1) \pmod{11} \\
9^5 = 1 & 10^5 = 10 = (-1) \pmod{11}
\end{array}
$$

A Legendre és Jacobi szimbólum

- ha a modulus egy P prímszám, akkor a Legendre szimbólum, $\mathbb{L}_P(a)$ értéke jelzi, hogy az a szám kvadratikus maradék vagy sem $\pmod P$ szerint. A Legendre szimbólumot a következőképpen is jelöljük $\left(\frac{a}{P}\right)$, ahol

$$\mathbb{L}_P(a) = a^{(P-1)/2} \pmod P,$$

$$\mathbb{L}_P(a) = \left(\frac{a}{P}\right) = \begin{cases} 0, & \text{ha } a \equiv 0 \pmod P \\ 1, & \text{ha } a \not\equiv 0 \pmod P \text{ és } \exists\, x: \ a \equiv x^2 \pmod P \\ -1, & \text{ha } a \not\equiv 0 \pmod P \text{ és } \nexists \text{ ilyen } x \end{cases}$$

- ha a modulus egy N összetett szám, ahol $N = P_1^{\alpha_1} \cdot P_2^{\alpha_2} \dots P_n^{\alpha_n}$, akkor a Jacobi szimbólumot $\mathbb{J}_N(a)$-t definiáljuk, amely azonban nem jelzi egyértelműen, hogy a kvadratikus maradék vagy sem $\pmod N$ szerint:

$$\mathbb{J}_N(a) = \left(\frac{a}{N}\right) = \left(\frac{a}{P_1}\right)^{\alpha_1} \cdot \left(\frac{a}{P_2}\right)^{\alpha_2} \dots \left(\frac{a}{P_n}\right)^{\alpha_n}.$$

Egy szám kvadratikus maradék vagy sem?

A $\pmod N$ szerinti kongruenciákra, ahol $N = P \cdot Q$ és P, Q különböző prímszámok kijelenthető:

- egy a szám akkor és csakis akkor kvadratikus maradék $\pmod N$ szerint ha a kvadratikus maradék $\pmod P$ szerint és kvadratikus maradék $\pmod Q$ szerint is,
- ha egy a szám kvadratikus maradék, akkor az $x^2 \equiv \pmod N$ kongruenciának négy inkongruens megoldása van,
- $\pmod N$ szerint a kvadratikus maradékok száma: $\frac{(P-1)\cdot(Q-1)}{4}$,
- $\mathbb{J}_N(a) = \left(\frac{a}{N}\right) = \left(\frac{a}{P}\right) \cdot \left(\frac{a}{Q}\right)$.

Egy szám kvadratikus maradék vagy sem?

Megállapítható, hogy:

- ha $\mathbb{J}_P(a) = 1$ és $\mathbb{J}_Q(a) = 1$, akkor $\mathbb{J}_N(a) = 1$ és a kvadratikus maradék (mod N) szerint
- ha $\mathbb{J}_P(a) = -1$ és $\mathbb{J}_Q(a) = -1$, akkor $\mathbb{J}_N(a) = 1$ és a kvadratikus nemmaradék (mod N) szerint
- ha $\mathbb{J}_P(a) = 1$ és $\mathbb{J}_Q(a) = -1$, akkor $\mathbb{J}_N(a) = -1$ és a kvadratikus nemmaradék (mod N) szerint
- ha $\mathbb{J}_P(a) = -1$ és $\mathbb{J}_Q(a) = 1$, akkor $\mathbb{J}_N(a) = -1$ és a kvadratikus nemmaradék (mod N) szerint

Tehát csak abban az esetben állapítható meg egy a számról, hogy kvadratikus maradék (mod N) szerint vagy sem, ha az N prímtényezős felbontása alapján meghatározzuk a prímtényezők szerinti a Legendre szimbólumok értékét! Ha a Legendre szimbólumok mindegyike 1 lesz, akkor az a szám kvadratikus maradék lesz (mod N) szerint.

Egy szám kvadratikus maradék vagy sem?

2. értelmezés

A kvadratikus maradék probléma azt jelenti, hogy egy N összetett szám esetében állpítsuk meg egy a egész számról hogy az kvadratikus maradék vagy sem, azzal a feltételezéssel élve, hogy nem ismertek az N prímosztói, illetve fennáll, hogy $\mathbb{J}_N(a) = 1$.

- Nagy számok esetében a kvadratikus maradék problémára nem ismert hatékony algoritmus, jelenleg minimum 1024 bites összetett számok esetében jelenthető ez ki biztonsággal.
- A Jacobi szimbólum meghatározására, azonban létezik hatékony algoritmus nagy számok esetében is. Meghatározása, a kvadratikus reciprocitás tétel alapján lehetséges. A tételnek több formája is ismert.
- A kvadratikus reciprocitás tételt 1744-ben Euler fogalmazta meg, de a bizonyítás Legnedre-től származik 1785-ből. Gauss is megfogalmazta a tételt 1795-ben, és élete nagy felfedezésének tartotta.

2. tétel (Kvadratikus reciprocitás tétel)

Ha P, Q páratlan prímszámok, akkor fennáll: $\left(\frac{P}{Q}\right) \cdot \left(\frac{Q}{P}\right) = (-1)^{\frac{(P-1)\cdot(Q-1)}{4}}$

A Jacobi szimbólum, tulajdonságok

Feltételezzük, hogy $n \geq 3$ páratlan egész szám:

$$\left(\frac{0}{n}\right) = \begin{cases} 1, & \text{ha } n = 1 \\ 0, & \text{ha } n \neq 1 \end{cases} \qquad \left(\frac{2}{n}\right) = \begin{cases} 1, & \text{ha } n \equiv 1 \pmod 8 \\ 1, & \text{ha } n \equiv 7 \pmod 8 \\ -1, & \text{ha } n \equiv 3 \pmod 8 \\ -1, & \text{ha } n \equiv 5 \pmod 8 \end{cases}$$

redukció:
$$\left(\frac{a}{n}\right) = \left(\frac{a \bmod n}{n}\right), \quad \text{ha } a \geq n$$

multiplikativitás:
$$\left(\frac{a \cdot b}{n}\right) = \left(\frac{a}{n}\right) \cdot \left(\frac{b}{n}\right),$$

a 2-es tényező leválasztása:
$$\left(\frac{a}{n}\right) = \left(\frac{2}{n}\right) \cdot \left(\frac{a \,//\, 2}{n}\right), \quad \text{ha } a \equiv 0 \pmod 2$$

reciprocitás:
$$\left(\frac{a}{n}\right) = \begin{cases} -1 \cdot \left(\frac{n}{a}\right), & \text{ha } a \equiv 3 \pmod 4 \text{ és } n \equiv 3 \pmod 4 \\ \left(\frac{n}{a}\right), & \text{másképp} \end{cases}$$

A Jacobi szimbólum, példa

Határozzuk meg $a = 23, n = 77$ esetében a Jacobi szimbólumot.

$$\text{reciprocitás:} \quad 77 \equiv 1 \pmod 4, 23 \equiv 3 \pmod 4 \Rightarrow \left(\frac{23}{77}\right) = \left(\frac{77}{23}\right),$$

$$\text{redukció:} \quad \left(\frac{77}{23}\right) = \left(\frac{77 \pmod{23}}{23}\right) = \left(\frac{8}{23}\right),$$

$$\text{multiplikativitás:} \quad \left(\frac{8}{23}\right) = \left(\frac{2}{23}\right)^3,$$

$$\text{a 2-es tényező leválasztása:} \quad 23 \pmod 8 = 7 \Rightarrow \left(\frac{2}{23}\right) = 1,$$

$$\text{tehát:} \quad \left(\frac{23}{77}\right) = 1.$$

Másfelől tudjuk 77 prímtényezős felbontását, ezért megállapítható az is, hogy 23 kvadratikus maradék $\pmod{77}$ szerint:

$$\left(\frac{23}{77}\right) = \left(\frac{23}{7}\right) \cdot \left(\frac{23}{11}\right),$$

$$\left(\frac{23}{7}\right) = 23^{(7-1)/2} \pmod 7 = 1, \quad \left(\frac{23}{11}\right) = 23^{(11-1)/2} \pmod 7 = 1.$$

A Jacobi szimbólum, példa

Határozzuk meg $a = 41$, $n = 77$ esetében a Jacobi szimbólumot.

reciprocitás: $77 \equiv 1 \pmod 4, 41 \equiv 1 \pmod 4 \Rightarrow \left(\frac{41}{77}\right) = \left(\frac{77}{41}\right)$,

redukció: $\left(\frac{77}{41}\right) = \left(\frac{77 \pmod{41}}{41}\right) = \left(\frac{36}{41}\right)$

multiplikativitás: $\left(\frac{36}{41}\right) = \left(\frac{4 \cdot 9}{41}\right)$,

a 2-es tényező leválasztása: $41 \pmod 8 = 1 \Rightarrow \left(\frac{2}{41}\right)^2 = 1 \cdot 1 = 1$,

reciprocitás: $\left(\frac{9}{41}\right) = \left(\frac{41}{9}\right)$,

redukció: $\left(\frac{41}{9}\right) = \left(\frac{41 \pmod 9}{9}\right) = \left(\frac{5}{9}\right)$,

reciprocitás: $\left(\frac{5}{9}\right) = \left(\frac{9}{5}\right)$,

redukció: $\left(\frac{9}{5}\right) = \left(\frac{9 \pmod 5}{5}\right) = \left(\frac{4}{5}\right)$,

a 2-es tényező leválasztása: $5 \pmod 8 = 5 \Rightarrow \left(\frac{2}{5}\right)^2 = (-1) \cdot (-1) = 1$,

tehát: $\left(\frac{41}{77}\right) = 1$.

Másfelől tudjuk 77 prímtényezős felbontását, ezért megállapítható az is, hogy 41 NEM kvadratikus maradék $\pmod{77}$ szerint:

$$\left(\frac{41}{77}\right) = \left(\frac{41}{7}\right) \cdot \left(\frac{41}{11}\right) = 1, \quad \left(\frac{41}{7}\right) = 41^{(7-1)/2} \pmod 7 = -1, \quad \left(\frac{41}{11}\right) = 41^{(11-1)/2} \pmod 7 = -1$$

A Jacobi szimbólum

```
def Jacobi(a, n):
    while True:
        if a == 0: return 0
        if a == 1: return 1
        e, r = 0, a
        while r & 1 == 0:
            e, r = e + 1, r >> 1
        #print('a, n, e, r:', a, n, e, r)
        if e & 1 == 0: s = 1
        else:
            temp = n % 8
            if temp == 1 or temp == 7: s = 1
            elif temp == 3 or temp == 5: s = -1
        if n % 4 == 3 and r % 4 == 3: s = -s
        n = n % r
        if r == 1: return s
        else: return s * Jacobi(n, r)

>>> Jacobi(41, 77)
    1
```

A négyzetgyök meghatározása

- ha a modulus egy P prímszám, és $P \equiv 3 \pmod 4$, akkor az $x^2 \equiv a \pmod P$ kongruencia megoldása (2 megoldás lesz), azaz a négyzetgyökök meghatározása a következőképpen történik:

$$x \equiv \pm a^{(P+1)/4} \pmod P$$

- Példa:
  - oldjuk meg az $x^2 = 5 \pmod{11}$ kongruenciát.
  - meghatározzuk $5^{(11+1)/4} \equiv 5^3 \equiv 4 \pmod{11}$
  - a kongruencia megoldása: $\pm 4 \pmod{11}$, azaz $4, 7$
  - ellenőrzés: $4^2 \equiv 5 \pmod{11}$ és $7^2 \equiv 5 \pmod{11}$.

A négyzetgyök meghatározása

Ha a modulus egy összetett N szám és tudjuk, hogy $N = P \cdot Q$ és $P \equiv Q \equiv 3 \pmod 4$, akkor az $x^2 \equiv a \pmod N$, kongruencia megoldása (4 megoldás) a következőképpen történik:

- először meghatározzuk a következő értékeket:

$$\begin{aligned} x_P &= a^{(P+1)/4} \pmod P, \\ x_Q &= a^{(Q+1)/4} \pmod Q, \end{aligned}$$

- majd alkalmazzuk a Kínai maradéktételt, ahol $x_1, -x_1, x_2, -x_2$ lesz a 4 lehetséges megoldás, ahol:

$$\begin{aligned} x_1 &= (P^{-1} \cdot P \cdot x_Q + Q^{-1} \cdot Q \cdot x_P) \pmod N, \\ x_2 &= (P^{-1} \cdot P \cdot x_Q - Q^{-1} \cdot Q \cdot x_P) \pmod N \end{aligned}$$

- P^{-1} érték P inverze $\pmod Q$ szerint
- Q^{-1} érték Q inverze $\pmod P$ szerint

A négyzetgyök meghatározása

Példa:

- legyen $N = P \cdot Q = 2773$, ahol $P = 47$ és $Q = 59$
- oldjuk meg a $x^2 = 17 \pmod{2773}$ kongruenciát
- meghatározzuk P^{-1}, Q^{-1} értékeket:

$$\begin{aligned} P^{-1} &\equiv 54 \pmod{59}, && \text{mert fennáll:} && 47 \cdot 54 \equiv 1 \pmod{59} \\ Q^{-1} &\equiv 4 \pmod{47}, && \text{mert fennáll:} && 59 \cdot 4 \equiv 1 \pmod{47} \end{aligned}$$

- meghatározzuk:

$$\begin{aligned} x_P &= 17^{(P+1)/4} = 17^{12} \equiv 8 \pmod{47} \\ x_Q &= 17^{(Q+1)/4} = 17^{15} \equiv 28 \pmod{59} \\ x_1 &= 54 \cdot 47 \cdot 28 + 4 \cdot 59 \cdot 8 \pmod{2773} = 854 \\ x_2 &= 54 \cdot 47 \cdot 28 - 4 \cdot 59 \cdot 8 \pmod{2773} = 2624 \end{aligned}$$

- a négy megoldás:

$$\begin{aligned} 854, && \text{ellenőrzés:} && 854^2 && \equiv 17 && \pmod{2773} \\ -854 \equiv 1919 \pmod{2773}, && \text{ellenőrzés:} && 1919^2 && \equiv 17 && \pmod{2773} \\ 2624, && \text{ellenőrzés:} && 2624^2 && \equiv 17 && \pmod{2773} \\ -2624 \equiv 149 \pmod{2773} && \text{ellenőrzés:} && 149^2 && \equiv 17 && \pmod{2773} \end{aligned}$$

A Rabin rendszer

- a Rabin-rendszert 1979-ben publikálta Michael O. Rabin,
- biztonsága a faktorizációs és kvadratikus maradék problémán alapszik,
- alkalmas titkos információ megosztására, ekkor hatékonyabb, mint az RSA,
- gyakorlatban az SAEP (2001, Boneh) rendszert használják, magas biztonsággal rendelkezik,
- megmutatható, hogy a Rabin rendszer feltörése ugyanolyan nehézségű, mint a fakorizációs probléma, ez nem igaz a "textbook" RSA-ra
- alkalmas digitális aláírásra: egy titkos információ alapján az eredeti üzenetet az aláíró egység aláírja/hitlesíti, amelyet egy másik egység letud ellenőrizni,
- a digitális aláírást hitelesítésre alkalmazzák, például az RSA, Diffie-Hellman publikus kulcsainak a hitelesítésére,
- a digitális aláírást alkalmazzák még adatintegritásra, letagadhatatlanságra,
- a rendszerben három algoritmust kell megadni: kulcsgeneráló, aláírás előállító és aláírás ellenőrző algoritmusok,
- az eredeti Rabin digitális aláírás rendszert módosították,

A Rabin titkosító rendszer

- a kulcsgeneráló algoritmus:
  - bemenete egy k biztonsági paraméter, a generált kulcsméret,
  - véletlenszerűen generál két k bites prímszámot, legyenek ezek p és q, ahol $p \equiv q \equiv 3 \pmod 4$, meghatározza az $n = p \cdot q$-t,
  - a publikus kulcs: (n), a privát kulcs: p, q,
- a titkosító algoritmus: bemeneti paramétere a publikus kulcs (n) és a K szöveg, meghatározza a $cK = K^2 \pmod n$ értéket
- a visszafejtő algoritmus: bemeneti paramétere a privát kulcs (p, q) és a titkosított szöveg, meghatározza a négyzetgyököt, azaz a $\hat{K} = cK^{\frac{1}{2}} \pmod n$ értéket:
  - $k_P = cK^{(p+1)/4} \pmod p$, $k_q = cK^{(q+1)/4} \pmod q$,
  - a kiterjesztett eukleidészi algoritmussal meghatározza p^{-1}-t, azaz p inverzét $\pmod q$ szerint és q^{-1}-t, azaz q inverzét $\pmod p$ szerint,
  - $K_1 = (p^{-1} \cdot p \cdot k_q + q^{-1} \cdot p \cdot k_p) \pmod n$,
  - $K_2 = (p^{-1} \cdot p \cdot k_q - q^{-1} \cdot q \cdot k_p) \pmod n$
- az $\hat{K} = K$ értéke a következő 4 megoldás között lesz:
  $K_1, K_2, K_3 = n - K_1, K_4 = n - K_2$

A Rabin titkosító rendszer, példa

- legyen $p = 11$, és $q = 31 \Rightarrow n = 341$,
- legyen $K = 42$, a nyílt-szöveg,
- titkosítás: $cK = 42^2 = 59 \pmod{341}$,
- visszafejtés:
  - $59^{(11+1)/4} = 9 \pmod{11}$,
  - $59^{(31+1)/4} = 20 \pmod{31}$,
  - meghatározzuk 31 inverzét mod 11 szerint: $31^{-1} = 5$, azaz fennáll: $31 \cdot 5 = 1 \pmod{11}$.
  - meghatározzuk 11 inverzét 31 szerint: $11^{-1} = 17$, azaz fennáll $11 \cdot 17 = 1 \pmod{31}$
  - $K_1 = (31 \cdot 31^{-1} \cdot 9 + 11 \cdot 11^{-1} \cdot 20) = 20 \pmod{341}$,
  - $K_2 = (31 \cdot 31^{-1} \cdot 9 - 11 \cdot 11^{-1} \cdot 20) = 42 \pmod{341}$,
  - $K_3 = 341 - K_1 = 321$,
  - $K_4 = 341 - K_2 = 299$.

Másodfokú kongruenciák, további tulajdonságok

Legyenek $p \equiv q \equiv 3 \pmod 4$ különböző prímek és legyen $n = p \cdot q$. Jelöljük $\mathbb{Q}_n$-nel a $\pmod n$ szerinti kvadratikus maradékok által alkotott halmazt, ekkor:

- ha $(a, n) = 1$, akkor $a^{(p-1)\cdot(q-1)/2} \equiv 1 \pmod n$,
- ha $a \in \mathbb{Q}_n$, akkor $a^{(n-p-q+5)//8} \pmod n$ az a egy négyzetgyöke $\pmod n$ szerint,
- ha $\left(\frac{a}{n}\right) = 1$, és $d = (n - p - q + 5)//8$, akkor

$$a^{2d} \pmod n = \begin{cases} a, & \text{ha } a \in \mathbb{Q}_n, \\ n - a, & \text{ha } a \notin \mathbb{Q}_n \end{cases}$$

- ha $p \not\equiv q \pmod 8$, akkor $\left(\frac{2}{n}\right) = -1$, ezért ha az a-t megszorozzuk 2-vel, vagy 2 inverzével $\pmod n$ szerint akkor a Jacobi szimbólum előjelet vált,
- a $p \equiv q \equiv 3 \pmod 4$ és $p \not\equiv q \pmod 8$ tulajdonsággal rendelkező prímszámokat Williams egészeknek mondják.

A Rabin digitális aláírás rendszer

- a kulcsgeneráló algoritmus:
  - bemenete egy k biztonsági paraméter, a generált kulcsméret,
  - véletlenszerűen generál két k bites prímszámot, legyenek ezek p és q, ahol $p \equiv 3 \pmod 8, q \equiv 7 \pmod 8$ meghatározza az $n = p \cdot q$-t,
  - a publikus kulcs: (n), a privát kulcs: $d = (n - p - q + 5)//8)$,
- az aláírás előállító algoritmus:
  - egy $K \in \{2, \ldots, (n-6)//16\}$ üzenetre meghatározza $\hat{K} = 16 \cdot K + 6$-t,
  - meghatározza a $\left(\frac{\hat{K}}{n}\right)$ Jacobi szimbólumot, és az s aláírást:

$$s = \begin{cases} \hat{K}^d \pmod n, & \text{ha } \left(\frac{\hat{K}}{n}\right) = 1 \\ (\hat{K}//2)^d \pmod n, & \text{ha } \left(\frac{\hat{K}}{n}\right) = -1 \end{cases}$$

- az aláírás ellenőrző algoritmus:
  - meghatározza a $\hat{K}_1 = s^2 \pmod n$,

$$\hat{K} = \begin{cases} \hat{K}_1, & \text{ha } \hat{K}_1 \equiv 6 \pmod 8 \\ 2 \cdot \hat{K}_1, & \text{ha } \hat{K}_1 \equiv 3 \pmod 8 \\ n - \hat{K}_1, & \text{ha } \hat{K}_1 \equiv 7 \pmod 8 \\ 2 \cdot (n - \hat{K}_1), & \text{ha } \hat{K}_1 \equiv 2 \pmod 8 \end{cases}$$

  - $K = (\hat{K} - 6)//16$

A Rabin digitális aláírás rendszer

```
def genPrimeRW(k):
    while True:
        p = getrandbits(k)
        if p % 8 == 3 and miller_rabinT(p): break
    while True:
        q = getrandbits(k)
        if q % 8 == 7 and miller_rabinT(q): break
    return p, q

def Rabin_key_gen(k):
    p, q = genPrimeRW(k)
    n = p * q
    d = (n - p - q + 5) // 8
    return n, d, p, q

>>> Rabin_key_gen(16)
    (2677933, 334144, 4139, 647)
>>> Rabin_key_gen(1024)
```

A Rabin digitális aláírás rendszer

```
def Rabin_sign(K, d, n):
    kK = 16 * K + 6
    j = Jacobi(kK, n)
    if j == 1: s = pow(kK, d, n)
    else: s = pow(kK // 2, d, n)
    return s

def mainRabin(k = 128):
    n, d, p, q = Rabin_key_gen(k)
    print('n: ', n)
    #K = int(input('K: '))
    while True:
        K = randint(2, (n-6) // 16)
        if K % 16 == 6: break
    print('K: ', K)
    s = Rabin_sign(K, d, n)
    print('s: ', s)
    K1 = Rabin_verify(s, n)
    print('K: ', K1)
    if K1 == K: return True
    return False
```

```
def Rabin_verify(s, n):
    kK1 = (s*s) % n
    temp = kK1 % 8
    if temp == 6: kK = kK1
    elif temp == 3: kK = 2 * kK1
    elif temp == 7: kK = n - kK1
    elif temp == 2: kK = 2 * (n - kK1)
    return (kK - 6) // 16

>>> mainRabin(16)
    n:  212768197
    K:  7557878
    s:  179085455
    K:  7557878
    True
```