

# Az informatikai biztonság alapjai

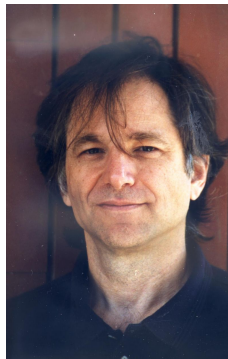
Pintér-Husztai Andrea

2023. október 1.

# Tartalom

- 1 RSA titkosítási séma
  - Matematikai alapok
  - RSA titkosítási séma
  - Biztonsági elemzés

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡



- 1977-ben jelent meg
- Tervezői: Ron Rivest, Adi Shamir, és Leonard Adleman
- Legtöbb Nyilvános Kulcs Infrastruktúra (PKI) termékben megtalálható, SSL/TLS tanúsítványok
- Biztonságos e-mail: PGP, Outlook

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ 🔍 ↺

# Kongruenciák

Két szám kongruens egy adott modulóra nézve, ha osztási maradékuk megegyezik.

## Definíció

Legyenek  $a$  és  $b$  egész számok és  $m$  pozitív egész. Azt mondjuk, hogy  $a$  **kongruens**  $b$ -vel modulo  $m$ , ha  $m \mid a - b$ .

Jelölés:  $a \equiv b \pmod{m}$

- Az  $m$  számot modulusnak nevezzük.
- Két szám pontosan akkor kongruens modulo  $m$ , ha  $m$ -mel osztva ugyanazt a maradékot adják
- Amennyiben  $a$  és  $b$  nem kongruensek modulo  $m$ , akkor  $a$  és  $b$  **inkongruensek** modulo  $m$ , jelölése:  $a \not\equiv b \pmod{m}$

Példák:  $13 \equiv 8 \pmod{5}$ ,  $25 \equiv -10 \pmod{7}$ ,  $25 \not\equiv 10 \pmod{7}$

# Euler-féle $\varphi$ függvény

## Definíció

Az  $a_1, a_2, \dots, a_n$  számok relatív prímek, ha nincs egységtől különböző közös osztójuk, azaz  $(a_1, a_2, \dots, a_n) = 1$ .

Példa: 5, -3, 15, -56 számok relatív prímek.

## Definíció

(Euler-féle  $\varphi$  függvény)

Tetszőleges  $n$  pozitív egész esetén  $\varphi(n)$  jelöli az  $1, 2, \dots, n$  számok közül az  $n$ -hez relatív prímek számát.

Példák:  $\varphi(10) = 4$ ,  $\varphi(7) = 6$

# Euler-féle $\varphi$ függvény

## Tétel

$$\varphi(n) = n \prod_{p|n} \frac{p-1}{p}, \text{ ahol } p \text{ prím.}$$

Példa:  $\varphi(100) = 100 \cdot \frac{1}{2} \cdot \frac{4}{5} = 40$

Vegyük észre:

$$\varphi(p) = p - 1,$$

ahol  $p$  prím, és

$$\varphi(p \cdot q) = (p - 1)(q - 1),$$

ahol  $p, q$  prímek

# Euler-Fermat tétel

## Tétel

(Euler-Fermat tétel, kétféle megfogalmazás)

Ha  $(a, m) = 1$ , akkor  $a^{\varphi(m)} \equiv 1 \pmod{m}$ .

## Tétel

(kis Fermat tétel)

- 1 Ha  $p$  prímszám,  $a \in \mathbb{Z}$  és  $p \nmid a$ , akkor  $a^{p-1} \equiv 1 \pmod{p}$ .
- 2 Ha  $p$  prímszám és  $a \in \mathbb{Z}$ , akkor  $a^p \equiv a \pmod{p}$ .

# Lineáris kongruenciák

Ezek olyan egyenletek, ahol az ismeretlen első fokon szerepel (lineáris), és modulóval dolgozunk.

## Definíció

*Ha  $a, b \in \mathbb{Z}$  és  $m$  pozitív egész, akkor az  $ax \equiv b \pmod{m}$  kongruenciát **lineáris kongruenciának** nevezzük.*

A kongruencia megoldásai olyan egész számok, melyeket  $x$  helyébe írva a kongruencia teljesül. Megoldások számán a különböző maradékosztályok számát értjük, melyekből vett egészek a kongruenciát kielégítik.

## Tétel

*Ha  $a, b \in \mathbb{Z}$  és  $m$  pozitív egész, az  $ax \equiv b \pmod{m}$  lineáris kongruencia akkor és csak akkor oldható meg, ha  $(a, m) \mid b$ .*

Ha az  $ax \equiv b \pmod{m}$  lineáris kongruencia megoldható, akkor a megoldások száma  $(a, m)$ .

## Tétel

*Ha  $(a, m) = 1$ , akkor az  $ax \equiv b \pmod{m}$  lineáris kongruencia egyetlen megoldása  $x \equiv a^{\varphi(m)-1} \cdot b \pmod{m}$ .*

### Bizonyítás

Ha  $x \equiv a^{\varphi(m)-1} \cdot b \pmod{m}$ , akkor  $x$  valóban megoldás, hiszen az Euler-Fermat tétel szerint  $aa^{\varphi(m)-1} \cdot b \equiv b \pmod{m}$ .  $\square$

# Szimultán kongruenciarendszer

A szimultán kongruenciarendszer (vagy kongruenciarendszer egyidejű egyenletekkel) azt jelenti, hogy több kongruencia-egyenletet egyszerre kell megoldani ugyanarra az ismeretlentre.

## Definíció

*Ha ugyanazon ismeretlentre több különböző modulusú kongruenciafeltételt adunk, akkor **szimultán kongruenciarendszert** kapunk.*

## Tétel

Az

$$x \equiv y_1 \pmod{m}$$

$$x \equiv y_2 \pmod{n}$$

*szimultán kongruenciarendszer megoldhatóságának szükséges és elégséges feltétele, hogy  $(m, n) \mid y_1 - y_2$ . Az összes megoldás egy maradékosztályt alkot modulo  $[m, n]$ .*

Az  $[m, n]$  az  $m$  és  $n$  modulusok legkisebb közös többszörösét jelöli.

# Kínai maradéktétel

A kínai maradéktétel a szimultán kongruenciarendszerek megoldására ad általános és hatékony módszert.

## Tétel

*Legyenek  $m_1, m_2, \dots, m_k$  páronként relatív prímek. Ekkor*

$$x \equiv y_1 \pmod{m_1}$$

$$x \equiv y_2 \pmod{m_2}$$

$$\vdots$$

$$x \equiv y_k \pmod{m_k}$$

*szimultán kongruenciarendszer bármilyen  $y_1, y_2, \dots, y_k$  egészek esetén megoldható, és a megoldások egyetlen maradékosztályt alkotnak modulo  $m_1 m_2 \cdots m_k$ .*

## Kínai maradéktétel

*Bizonyítás*

Legyen  $M = m_1 m_2 \cdots m_k$  és  $M_i = \frac{M}{m_i}$ , ahol  $i = 1, 2, \dots, k$ .

Mivel az  $m_1, \dots, m_k$  modulusok páronként relatív prímek, ezért  $(M_i, m_i) = 1$ ,  $i = 1, 2, \dots, k$ . Tekintsük az  $M_i \cdot x \equiv 1 \pmod{m_i}$ , ahol  $i = 1, 2, \dots, k$  kongruenciákat. Bármely  $i$  esetén a kongruencia megoldható, hiszen  $(M_i, m_i) = 1$ , legyen  $x \equiv c_i \pmod{m_i}$  a megoldás. Ekkor  $M_i \cdot c_i \equiv 1 \pmod{m_i}$  és  $M_i \cdot c_i \equiv 0 \pmod{m_j}$ , ahol  $i \neq j$ . Legyen

$$x_0 \equiv \sum_{i=1}^k M_i c_i y_i \pmod{M}.$$

Az előbbiek alapján  $x_0 \equiv y_i \pmod{m_i}$ ,  $i = 1, \dots, k$ , tehát  $x_0$  megoldása a szimultán kongruenciarendszernek. Ezzel egy konstruktív bizonyítást adtunk a megoldás létezésére.

Most lássuk be, hogy csak egy megoldása van. Tegyük fel, hogy  $x'_0$  szintén megoldása a szimultán kongruenciarendszernek. Így  $x'_0 \equiv y_i \pmod{m_i}$ ,  $i = 1, 2, \dots, k$ , azaz  $m_i \mid x'_0 - y_i$ , bármely  $i$ -re.

Ugyanakkor  $x_0$  szintén megoldás, tehát  $m_i \mid x_0 - y_i$ , ahonnan  $m_i \mid x'_0 - x_0$  bármely  $i$ -re. Ha az előbbi oszthatóság tetszőleges  $i \in \{1, 2, \dots, k\}$ -re teljesül, akkor  $m_1 m_2 \cdots m_k \mid x'_0 - x_0$  is áll, azaz  $M \mid x'_0 - x_0$ , így  $x'_0 \equiv x_0 \pmod{M}$ . Tehát  $x'_0$  és  $x_0$  megoldások egy maradékosztályba esnek modulo  $M$ .  $\square$

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡

# RSA titkosítási séma

Aszimmetrikus titkosítási séma:  $AE = (Key, Enc, Dec)$

- *Key*:

- 1 Véletlenül választunk két nagy prímet:  $p, q$ .
- 2 Kiszámítjuk az RSA modulust:  $n = p \cdot q$ .
- 3 Kiszámítjuk az Euler-féle  $\phi$  függvényt:  $\phi(n) = (p - 1)(q - 1)$ .
- 4 Választunk egy *véletlen*  $e$  egészt:  $1 < e < \phi(n)$  és  $(e, \phi(n)) = 1$ . ( $e$  titkosító kitevő)
- 5 Kiszámítjuk  $d$ -t:  $1 < d < \phi(n)$  és  $ed \equiv 1 \pmod{\phi(n)}$ . ( $d$  visszafejtő kitevő)

$PK = (n, e)$ ,  $SK = d$  and  $\phi(n), p, q$  titkos paraméterek

$\mathcal{P} = \mathcal{C} = \mathbb{Z}_n$

- $Enc_{PK}(m) = m^e \pmod{n} \forall m \in \mathcal{P}$  és  $PK = (n, e)$  mellett.
- $Dec_{SK}(c) = c^d \pmod{n} \forall c \in \mathcal{C}$  és  $SK = d$  mellett.

## Példa

Aszimmetrikus titkosítási séma:  $AE = (Key, Enc, Dec)$

- *Key*:

- 1 Véletlenül választunk két nagy prímet:  $p = 5, q = 11$ .
- 2 Kiszámítjuk az RSA modulust:  $n = p \cdot q = 55$ .
- 3 Kiszámítjuk az Euler-féle  $\phi$  függvényt:  
 $\phi(n) = (p - 1)(q - 1) = 40$ .
- 4 Választunk egy *véletlen*  $e$  egészt:  $1 < e < \phi(n)$  és  
 $(e, \phi(n)) = 1$ ,  $e = 3$ .
- 5 Kiszámítjuk  $d$ :  $1 < d < \phi(n)$  és  $ed \equiv 1 \pmod{\phi(n)}$ ,  $d = 27$ .

$PK = (n = 55, e = 3)$ ,  $SK = d = 27$  és

$\phi(n) = 40, p = 5, q = 11$  titkos paraméterek

$\mathcal{P} = \mathcal{C} = \mathbb{Z}_{55}$

- $m = 8$  és  $PK = (55, 3)$ :  $Enc_{PK}(m) = 8^3 \pmod{55}$   
 $8^3 \equiv 17 \pmod{55}$
- $c = 17$  és  $SK = 27$ :  $Dec_{SK}(c) = 17^{27} \pmod{55}$   
 $17^{27} \equiv 8 \pmod{55}$

# Kapcsolódó algoritmusok

- Key:
  - ① Véletlenül választunk két nagy prímet:  $p, q$ . -> **Prímtesztek**  
(pl. Miller-Rabin prímteszt)
  - ② Választunk egy véletlen  $e$  egészt:  $1 < e < \phi(n)$  és  
 $(e, \phi(n)) = 1$ . -> **Euklideszi algoritmus**
  - ③ Kiszámítjuk  $d$ -t:  $1 < d < \phi(n)$  és  $ed \equiv 1 \pmod{\phi(n)}$ . ->  
multiplikatív inverz számítása: **Kibővített Euklideszi Algoritmus**
- $Enc_{PK}(m) = m^e \pmod{n} \quad \forall m \in \mathcal{P}$  és  $PK = (n, e)$  mellett. -> **Gyors hatványozás**
- $Dec_{SK}(c) = c^d \pmod{n} \quad \forall c \in \mathcal{C}$  és  $SK = d$  mellett. -> **Kínai Maradéktétel alkalmazása**

# Biztonsági elemzés

# $SK$ kiszámítása $PK$ ismeretében nehéz $\longrightarrow$ prímfaktorizáció

A támadó célja a titkos kulcs megszerzése:

Tétel: A  $d$  exponens kiszámítása az  $(n, e)$  paraméterek ismeretében **ugyanolyan nehéz**, mint az  $n$  modulus  $p$  és  $q$  prímfaktorainak meghatározása, ahol  $|p| = |q|$ .

Megjegyzés:

- Ha meg tudjuk határozni  $n$  faktorait, akkor  $d$  kiszámítható a  $ed \equiv 1 \pmod{\phi(n)}$  kongruenciából.  $\phi(n)$  kiszámításához  $p$  és  $q$  ismerete szükséges.
- Ha van egy hatékony algoritmus  $d$  kiszámítására  $(n, e)$  ismeretében, akkor ez az algoritmus alkalmas  $n$  faktorizálására.

# Prímfaktorizáció

- Ha az  $n$  modulus  $p$  és  $q$  faktora elég nagyok, akkor nem ismerünk hatékony (polinomiális idejű) algoritmust  $n$  faktorizálására.
- Paraméterek méretei:  $|n| = 768$ , ahol  $|p| = |q| = 384$  modulust faktorizálták 2009-ben Számtest Szita algoritmussal.  
 $|n| = 1024$ , ahol  $|p| = |q| = 512$  biztonságos.  
<https://unideb.hu>  $\rightarrow |n| = 4096$  (hosszú távra archiválás)

\_\_\_\_\_



# A nyílt üzenet $m$ kiszámítása a $c$ ismeretében nehéz——>

## RSA probléma

A támadó célja a nyílt üzenet meghatározása:

- **Az RSA Probléma:** Adott  $(n, e)$  RSA nyilvános kulcs és  $c \equiv m^e \pmod{n}$  titkosított üzenet mellett  $m$  nyílt üzenet kiszámítása.
- A támadó nem feltétlenül ismeri a titkos kulcsot. Feladat:  $c^{\frac{1}{e}} \pmod{n}$  kiszámítása.
- Az RSA Probléma nehéz, ha az  $n$  modulus elég nagy és a prímek véletlenül generáltak, valamint az  $m$  nyílt üzenet (emiat a  $c$  titkosított üzenet is) egy a 0 és  $n - 1$  közé eső véletlen egész.
- Az  $m$  nyílt üzenet véletlensége a  $[0, n - 1]$  intervallum felett fontos feltétel. Ha  $m$  egy kis halmazból vett, akkor a támadó könnyen meghatározhatja  $m$ -et úgy, hogy egyenként próbálgatja az összes lehetséges  $m$ -et (brute force).

# Prímfaktorizáció vs. RSA probléma

Adott egy nagyon nagy összetett szám ( $n$ ) és egy másik, hozzá kapcsolódó szám ( $e$ ), valamint egy titkosított üzenet ( $c$ ), mennyire nehéz kitalálni az eredeti üzenetet ( $m$ ) anélkül, hogy ismernénk a titkos kulcsot ( $d$ ) vagy az  $n$  szám prímtényezőit ( $p$  és  $q$ )?

- Prímfaktorizáció  $\rightarrow$  RSA probléma

Az RSA Probléma nem nehezebb, mint a prímfaktorizáció, hiszen ha a támadó képes az  $n$  modulus faktorizálására, akkor ki tudja számolni a  $d$  titkos kulcsot az  $(n, e)$  nyilvános kulcsból.

- RSA probléma  $\rightarrow$  Prímfaktorizáció

Nem tudjuk, hogy ha az RSA probléma megoldható, akkor tudunk-e hatékony algoritmust adni a prímfaktorizációra.

# Támadások a tankönyvi RSA-val szemben

- Speciális a nyílt üzenetek halmaza

*Támadás:* Csak a titkosított üzenet ismert (COA)

Input:  $c, (n, e)$  Output:  $m$ , ahol  $m^e \equiv c \pmod{n}$

Algoritmus: Minden lehetséges nyílt üzenetet kipróbálunk.  
(Brute Force)

- Titkosított üzenetek közötti kapcsolat

*Támadás:* Választott üzenet alapú támadás (CCA)

Input:  $c, (n, e)$  és  $m'$  egy választott  $c'$ -re

Output:  $m$

Algoritmus:

- 1 Választunk egy véletlen  $r \in \mathcal{P}$
- 2 Kiszámítjuk  $r' \equiv r^e \pmod{n}$ , és kérjük  $c' \equiv r' \cdot c \pmod{n}$  visszafejtését
- 3 Megkapjuk  $m'$ -t, ahol  $m' \equiv (c')^d \equiv (r' \cdot c)^d \equiv r \cdot m \pmod{n}$
- 4  $r$  ismeretében  $m$  kiszámítható  $m'$ -ből

# RSA-OAEP

- A tankönyvi RSA egy nem biztonságos titkosítási séma.
- Gyakorlatban: RSA-OAEP (Optimal Asymmetric Encryption Padding):

Adott:  $G : \{0, 1\}^k \rightarrow \{0, 1\}^l$ ,  $H : \{0, 1\}^l \rightarrow \{0, 1\}^k$ ,

$\mathcal{P} = \{0, 1\}^l$

Titkosítás:

- Input:  $m \in \{0, 1\}^l$  és  $r \in \{0, 1\}^k$
- Output:  $c = ((m \oplus G(r)) || (r \oplus H(m \oplus G(r))))^e \pmod{n}$

Visszafejtés:

- Input:  $c \in \{0, 1\}^{l+k}$
- Kiszámítjuk:
  - $c^d = ((m \oplus G(r)) || (r \oplus H(m \oplus G(r)))) \pmod{n}$
  - Meghatározzuk  $r$ :  $r = (r \oplus H(m \oplus G(r))) \oplus H(m \oplus G(r))$
  - $m = (m \oplus G(r)) \oplus G(r)$
- Output:  $m$

# Egyirányú függvény

- Egyirányú függvény:
  - **Kiszámítani könnyű:** Adott  $x$ , és könnyű kiszámítani  $f(x)$ -et.
  - **Nehéz invertálni:** Adott  $f(x)$ , nehéz kiszámítani  $x$ -et.
- Nem tudjuk, hogy létezik -e egyirányú függvény.
- Egyirányúnak bizonyul:  **$p, q$  prímek szorzata**, ahol  $|p| = |q|$ 
  - **Könnyű kiszámítani:** Adott  $p, q$ , könnyű  $f(p, q) = p \cdot q$  kiszámítása.
  - **Nehéz invertálni:** Adott  $f(p, q) = p \cdot q$ , nehéz kiszámítani  $p$  vagy/és  $q$ -t. (prímfaktorizáció)

# Egyirányú csapóajtó függvény

- Egyirányú csapóajtó függvény:
  - Egyirányú függvény
    - **Kiszámítani könnyű:** Adott  $x$ , és könnyű kiszámítani  $f(x)$ -et.
    - **Nehéz invertálni:** Adott  $f(x)$ , nehéz kiszámítani  $x$ -et.
  - Csapóajtó információ: Bizonyos plusz információval viszont könnyű invertálni:  $x$  kiszámítása  $f(x)$ -ből.
- Nem tudjuk, hogy létezik -e egyirányú csapóajtó függvény.
- Egyirányú csapóajtó függvénynek bizonyul: **moduláris hatványozás RSA modulussal**
  - Egyirányú függvény:
    - **Kiszámítani könnyű:** Adott  $m$ , könnyű kiszámítani  $f(m) = m^e \pmod{n}$  (gyors hatványozás).
    - **Nehéz invertálni:** Adott  $f(m) = m^e \pmod{n}$  és  $(n, e)$ , nehéz kiszámítani  $m$ -et. (RSA probléma)
  - Csapóajtó információ:  $d, p, q, \phi(n)$ .