

# Kriptográfia és Információbiztonság

## 1. előadás

MÁRTON Gyöngyvér

Sapientia Egyetem, Matematika-Informatika Tanszék  
Marosvásárhely, Románia  
`mggyongyi@ms.sapientia.ro`

2022

# Követelmények, osztályozás

- **Vizsga:** dolgozat és plusz pontok
- **Dolgozat:**
  - legalább 4, de nem több mint 6 oldalas kell legyen (A4 lapméret, 11-es font, 1.5 sorköz),
  - tartalma tudományos jellegű kell legyen és kapcsolódnia kell a 21. századi adatbiztonsághoz,
  - határidőre kell beküldeni, és 10-15 percben be kell mutatni.
  - beküldési határidő: április 24 éjféltől
  - bemutatási időpontok, megegyezés szerint az utolsó oktatási héten, illetve a vizsgaidőszakban
- **Plusz pontok:** kijelölt programozási feladatok megoldása és bemutatása laborórákon, a félév során (kivéve az utolsó oktatási hetet). Maximum 3 pluszpontot lehet szerezni, ami a dolgozatjegyet emeli.
- **Pótvizsgák:** szóbeli, ahol a diák kap egy tételsort, amelyre szóban kell válaszoljon. Egy tétel három, a törzsanyagból vett kérdést fog tartalmazni.

# Kriptográfia és információbiztonság

- Kriptográfia (cryptography)
  - az információ elrejtése, rejtjelezése
  - bizalmas információ-csere
  - az adatok sértetlenségének a biztosítása
  - kommunikáló felek azonosítása
  - stb.
- Hozzáférési jogosultságok (access control)
  - hitelesítés (authentication): adott eszköz, felhasználó beléphet-e egy rendszerbe, hozzáférhet-e egy rendszer adataihoz
  - engedélyezés (authorization): milyen jogosultságokkal rendelkezik egy adott eszköz, felhasználó egy adott rendszeren belül
- Biztonsági protokollok (protocols): szabályok összessége, amelyeket adott helyzetben be kell tartani, ahhoz hogy a rendszer biztonsága ne sérüljön
- Szoftverek (software) az információ tárolására, továbbítására, feldolgozására, stb. szoftvereket használunk; pl egy vírus olyan szoftver, ami kárt tehet egy adott informatikai rendszerben

# Alkalmazási terület

- A kommunikáció biztonsága:
  - a webforgalom biztonságát kriptográfiai eljárások biztosítják, a protokoll neve: HTTPS (RSA, Diffie-Hellman, AES, 3DES, RC4)
  - a wifi eszközök biztonságos adatmegosztását a WPA2 protokoll biztosítja
  - a mobil telefonok, a GSM alapú telefonok biztonságát folyamatirkosítási eljárás biztosítja: A5 titkosító család (linear feedback shift register)
  - bluetooth protokoll biztonságát folyamatirkosítási eljárás biztosítja: EO
- A merevlemez biztonsága: a Windows EFS (Encrypting File System) titkosítása, régebbi verziókban DESX, az újabbakban AES, SHA, ECC
- A Blu-ray, DVD, CD tartalmak biztonsága: AACS (Advanced Access Control System) másolásvédelmi technológia, a lemezen és a készüléken titkosító kulcsok vannak beállítva, amelyeket a lejáratí idő után meg kell újítani
- ...









# Kriptográfia, törzsanyag

- Klasszikus kriptográfiai rendszerek (Caesar és változatai, Affin, Hill)
- Titkos-kulcsú titkosító rendszerek (secret-key encryption systems, symmetric encryption systems)
  - matematikai modell, biztonság, tervezés
  - folyam-titkosító rendszerek (stream ciphers): OTP, álvéletlen-szám generátorok, RC4, LFSR, A5/1, Salsa20,
  - blokk-titkosító módok (block cipher mode)(ECB, CBC, CFB, OFB, CTR)
  - blokk-titkosító rendszerek (block ciphers): DES és változatai, TEA, AES.
- Üzenet-hitelesítő kódok (message authentication codes): HMAC
- Hash-függvények (hash functions): követelmények, biztonság, tervezés, az SHA-1 hashfüggvény

# Kriptográfia, törzsanyag

- Nyilvános-kulcsú kriptorendszerek (public-key cryptography, asymmetric cryptography)
  - Titkosító rendszerek
    - matematikai modell, biztonság
    - a faktorizációs problémán alapuló titkosító rendszerek: RSA, RSA-OAEP,
    - a kvadratikus maradék problémán alapuló titkosító rendszerek: Rabin, SAEP,
  - Álvéletlen számgenerátorokon alapuló rendszerek: Blum-Goldwasser, Goldreich-Levin
  - A diszkrét logaritmus problémán alapuló rendszerek: Diffie-Hellman kulcscsere, elliptikus görbéken alapuló kriptográfia
  - Digitális aláírások (digital signatures): matematikai modell, biztonság, az RSA, a DSA digitális aláírás

# Könyvészet I

-  Freud R., Gyarmati E., Számelmélet, Nemzeti Tankönyvkiadó, Budapest, 2000.
-  Cormen T.H., Leiserson C.E., Rivest R.L., Algoritmusok, Műszaki Könyvkiadó, Budapest, 2001.
-  Rónyai L. Ivanyos G., Szabó R., Algoritmusok, Typotex, Budapest, 2004
-  Buttyán L., Vajda I.: Kriptográfia és alkalmazásai, Typotex, Budapest, 2004.
-  Márton Gy, Kriptográfiai alapismeretek, Scientia Kiadó, Kolozsvár, 2008.
-  Buchmann J., Introduction to cryptography, Springer, 2002.
-  Boneh D.: Introduction to Cryptography. Online Cryptography.  
<https://www.coursera.org/>
-  Hoffstein J. , Pipher J., Silverman J.H., An Introduction to Mathematical Cryptography, Springer, 2008.

# Könyvészet II



Menezes J., van Oorschot P.C., Vanstone S.A., Handbook of Applied Cryptography, CRC Press, Boca Raton, Florida, 1997.



Stallings W., Cryptography and network security. Principles and practice, Pearson, 2011.



Stamp M., Information security. Principles and practice, John Wiley&Sons, 2006.



Stinson D.R., Cryptography theory and practice, Chapman&HallCRC, 2006.



# Történelmi háttér

- i.e. IV század: spártaiak titkosítása, szkütalé használata (transzpozíció/felcserélés)
- i.e. I. század: Caesar-titkosító (szubsztitúció/helyettesítés)
- Megjegyzés: a mai titkosító rendszerek alapl műveletei: transzpozíció és szubsztitúció
- 1926, Vernam titkosító
- 1949, Shannon: tökéletes biztonság (perfect secrecy)
- 1970, DES (Data Encryption Standard): Horst Feistel
- 1976, Diffie-Hellman: publikus-kulcsú kriptográfia alapjai
- 1977, RSA kriptorendszer: Rivest, Shamir, Adleman
- 1985, ElGamal kriptorendszer: Taher ElGamal
- 1994, RSA-OAEP titkosító rendszer, új biztonságértelmezések: Bellare, Rogaway
- 1998, Cramer-Shoup titkosító rendszer (az ElGamal egy kiterjesztett változata)
- 2001, AES (Advanced Encryption Standard): Daemen, Rijmen
- 2004, az ECC széleskörben való elterjedése, az első ajánlás még 1985-ben történt

# Bevezető

- Klasszikus rendszerek
  - alkalmazásuk: diplomáciai-katonai életben, csak titkosítást végeztek,
  - könnyen feltörhetőek statisztikai számítások segítségével.
- Kibővült feladatkör: kulcs-csere, hitelesítés, titok-megosztás, elektronikus szavazás, stb.
- Alapfogalmak: bemeneti ábécé (input alphabet), nyílt-szöveg (plaintext), titkosított-szöveg (ciphertext), kulcs (key), titkosítás/rejtjelezés (encryption), visszafejtés (decryption),
- Kerchoff elv: a titkosító, a hitelesítő algoritmus nyilvános, csak a rendszerben alkalmazott kulcsok titkosak.

- Támadási célok:
  - meghatározni a kulcsot,
  - megfejteni egy bizonyos titkosított-szöveget,
  - megfejteni egy részét egy bizonyos titkosított szövegnek,
  - különbséget tenni, két, különböző nyilvános-szöveg titkosított értékei között.
- Alapelv: standardizált rendszerek alkalmazása

# Caesar titkosító

- $M = C = \{0, 1, \dots, 25\}^*$ , az angol ábécé 26 betűjének megfelelő számkód:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

- $K = \{0, 1, \dots, 25\}$ ,
- *Gen*: kulcsgenerálás, kiválaszt egy *key* kulcsot,
- *Enc<sub>key</sub>*: titkosítás,  $c = (m + \text{key}) \pmod{26}$ ,
- *Dec<sub>key</sub>*: visszafejtés  $m = (c + 26 - \text{key}) \pmod{26}$ .
- Megjegyzés: *key* = 0 nincs titkosítás, *key* = 3, az eredeti Caesar titkosító
- Feltörési módszer: az összes lehetséges kulcs kipróbálása (exhaustive key search): 26 lehetséges kulcs van.

Példa:

- Ha a nyílt szöveg a következő: *THISISAPLAINTEXT*
- Ha a kulcs **5**, akkor a titkosított szöveg: *YMNXNXFUQFNSYJCY*
- A megfelelő titkosító tábla:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E

# Helyettesítéses rejtjelezés

- $M = C = \{0, 1, \dots, 25\}$
- $K$ , a 26 szimbólum összes lehetséges permutációja,
- $Enc(m) = Perm(m) \pmod{26}$ ,
- $Dec(c) = Perm^{-1}(c) \pmod{26}$ .
- Feltörési módszerek:
  - az összes lehetséges kulcs kipróbálása nem működik, mert a lehetséges kulcsok száma:  $26! = 403291461126605635584000000$
  - működik: betű, betű-pár, betű-hármas, szavak gyakoriság vizsgálat.

Példa, Keyword Caesar titkosító, ahol a kulcs két értékből tevődik össze:

- Ha a nyílt szöveg a következő: *THISISAPLAINTTEXT*
- Ha a kulcs **7, PASWD**, akkor a titkosított szöveg: *PLMZMZEVQEMTPIDP*
- A megfelelő titkosító tábla, ahol vegyük észre, hogy  $26 - 19 = 7$ :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
E	F	G	H	I	J	K	L	M	N	O	Q	R	T	U	V	X	Y	Z	P	A	S	W	D	B	C

# Az Affin rejtjelezés

- helyettesítéses, monoalfabetikus rejtjelezés,
- $M = C = \{0, 1, \dots, 25\}^* = \mathbb{Z}_{26}$ ,  $K = \{(a, b) \in \mathbb{Z}_{26}, \gcd(a, 26) = 1\}$ ,
- $key = (a, b)$
- $Enc(m) = (a \cdot m + b) \pmod{26} \Rightarrow c$ ,
- $Dec(c) = a^{-1} \cdot (c + 26 - b) \pmod{26}$ .
- $a^{-1}$ , az  $a$  multiplikatív inverze  $\pmod{26}$  szerint, ami azt jelenti, hogy meg kell határozni azt az  $a^{-1}$  értéket, amelyre fennáll:

$$a \cdot a^{-1} = 1 \pmod{26}.$$

- a multiplikatív inverz akkor létezik, ha  $\gcd(a, 26) = 1$ , ahol  $\gcd$  - greatest common division/legnagyobb közös osztó
- Ha létezik multiplikatív inverz, akkor az meghatározható :
  - az összes érték kipróbálásával: 12 szorzás szükséges,
  - a kiterjesztett Eukleidészi algoritmussal,
  - az Euler tételt alkalmazva:  $a^{-1} = a^{\phi(26)-1} = a^{11} \pmod{26}$ , mert  $\phi(26) = \phi(2) \cdot \phi(13) = 12$ .

# Az Affin rejtjelezés - példa

- Ha a kulcs  $(5, 2)$  és a nyílt-szöveg a következő:

*AMATHEMATICIAN,*

- akkor a titkosított-szöveg:

*CKCTLWKCTQMCP,*

- ahol a titkosított-szöveg első 6 karakterét a következőképpen határoztuk meg:

$$\begin{array}{llll} A & \rightarrow & C : & (5 \cdot 0 + 2) = 2 \pmod{26} \\ M & \rightarrow & K : & (5 \cdot 12 + 2) = 10 \pmod{26} \\ A & \rightarrow & C : & (5 \cdot 0 + 2) = 2 \pmod{26} \\ T & \rightarrow & T : & (5 \cdot 19 + 2) = 19 \pmod{26} \\ H & \rightarrow & L : & (5 \cdot 7 + 2) = 11 \pmod{26} \\ E & \rightarrow & W : & (5 \cdot 4 + 2) = 22 \pmod{26} \end{array}$$

- A visszafejtéshez szükséges kulcs:  $(21, 2)$ , mert  $5 \cdot 21 = 105 = 1 \pmod{26}$ , azaz, 5 multiplikatív inverze  $\pmod{26}$  szerint 21.

# A bináris legnagyobb közös osztó algoritmus

- meghatározza az  $a$  és  $b$  egész számok legnagyobb közös osztóját, a szorzás és osztás műveletek helyett kivonást, inkrementációt és 2 hatványaival való osztást és szorzást végez, ahol az utóbbi két művelet hatékonyan implementálható
- az algoritmus futási ideje:  $O(l^2)$ , where  $l = \max(\text{len}(a), \text{len}(b))$

```
typedef long long uInteger;
#define swap(x,y) { x = x ^ y; y = x ^ y; x = x ^ y; }

uInteger binGCD(uInteger a, uInteger b) {
    uInteger k = 0;
    while (!(a & 1) && !(b & 1)) {
        a = a >> 1;
        b = b >> 1;
        k++;
    }
    while (1) {
        while (!(a & 1)) a = a >> 1;
        while (!(b & 1)) b = b >> 1;
        if (b < a) swap(a, b);
        b = b - a;
        if (b == 0) return a << k;
    }
}
```



# Az Eukleidészi algoritmus és változatai

A kiterjesztett Eukleidészi algoritmus:

- meghatározza azokat a  $d, x_0, y_0$  egész számokat, amelyek esetében fennáll, hogy  $d$  az  $a$  és  $b$  legnagyobb közös osztója és  $a \cdot x_0 + b \cdot y_0 = d$ ,
- az algoritmus egy  $x_0, y_0$  párt/megoldást határoz meg, de ezek száma végtelen
- ha  $x_0, y_0$  egy megoldás, akkor bármilyen más  $x, y$  megoldás a következőképpen határozható meg, ahol  $k$  tetszőleges egész szám:

$$x = x_0 - k \cdot \frac{b}{d}, y = y_0 + k \cdot \frac{a}{d}.$$

- az algoritmus futási ideje:  $O(1^2)$ , where  $1 = \max(\text{len}(a), \text{len}(b))$

Multiplikatív inverz meghatározása:

- abban az esetben ha  $d = 1$  és  $d$  az  $a$  és  $b$  legnagyobb közös osztója meghatározza az  $x_0$  egész számot, úgy hogy  $a \cdot x_0 \equiv 1 \pmod{b}$ ,
- az algoritmus futási ideje:  $O(1^2)$ , where  $1 = \text{len}(b)$

# A kiterjesztett Eukleidészi algoritmus

```
uInteger* binExtEuclid(uInteger a, uInteger b) {
    uInteger k = 0;
    while (!(a & 1) && !(b & 1))
        a = a >> 1, b = b >> 1, k++;
    uInteger aT = a, bT = b, x0 = 1, x1 = 0, y0 = 0, y1 = 1;
    while (1) {
        while (!(a & 1)) {
            a = a >> 1;
            if (!(x0 & 1) && !(y0 & 1))
                x0 = x0 >> 1, y0 = y0 >> 1;
            else
                x0 = (x0 + bT) >> 1, y0 = (y0 - aT) >> 1;
        }
        while (!(b & 1)) {
            b = b >> 1;
            if (!(x1 & 1) && !(y1 & 1))
                x1 = x1 >> 1, y1 = y1 >> 1;
            else
                x1 = (x1 + bT) >> 1, y1 = (y1 - aT) >> 1;
        }
        if (b < a){
            swap(a, b); swap(x0, x1); swap(y0, y1);
        }
        b = b - a;
        x1 = x1 - x0, y1 = y1 - y0;
        if (b == 0) {
            uInteger* res = new uInteger[3];
            res[0] = a << k, res[1] = x0, res[2] = y0;
            return res;
        }
    }
}
```

# Multiplikatív inverz meghatározása

```
uInteger binInverse(uInteger a, uInteger b) {
    uInteger k = 0, aF = a, bF = b;
    while (!(a & 1) && !(b & 1))
        a = a >> 1, b = b >> 1, k++;
    uInteger aT = a, bT = b;
    uInteger g, x0 = 1, x1 = 0;
    while (1) {
        while (!(a & 1)) {
            a = a >> 1;
            if (!(x0 & 1)) x0 = x0 >> 1;
            else x0 = (x0 + bT) >> 1;
        }
        while (!(b & 1)) {
            b = b >> 1;
            if (!(x1 & 1)) x1 = x1 >> 1;
            else x1 = (x1 + bT) >> 1;
        }
        if (b < a) {
            swap(a, b); swap(x0, x1);
        }
        b = b - a;
        x1 = x1 - x0;
        if (b == 0) {
            g = a << k;
            if (g != 1) return -1;
            else
                if (x0 < 0) x0 += aF;
                return x0;
        }
    }
}
```

# Az Affin rejtjelezés - ismert nyílt-szöveg támadás

Feltörési módszerek:

- gyakoriság vizsgálat
- az összes lehetséges kulcs kipróbálása, kulcsok száma: 312
- ismert nyílt-szöveg támadás: rendelkezünk két betű rejtjelezett értékével,
  - tudva hogy az  $m_1$  rejtjele  $c_1$ , és az  $m_2$  rejtjele  $c_2$ , akkor a következő kongruencia-rendszer megoldásával, megállapítható a titkosításhoz használt  $(a, b)$  kulcs, ahol  $(m_1 - m_2)^{-1}$  az  $m_1 - m_2$  multiplikatív inverze:

$$\begin{aligned}m_1 \cdot a + b &= c_1 \pmod{26} \\m_2 \cdot a + b &= c_2 \pmod{26}.\end{aligned}$$

$$(m_1 - m_2) \cdot a = (c_1 - c_2) \pmod{26}$$

$$a = (m_1 - m_2)^{-1} \cdot (c_1 - c_2) \pmod{26}$$

$$b = (c_1 - m_1 \cdot a) \pmod{26}$$

vagy

$$b = (c_2 - m_2 \cdot a) \pmod{26}$$