Kriptográfia

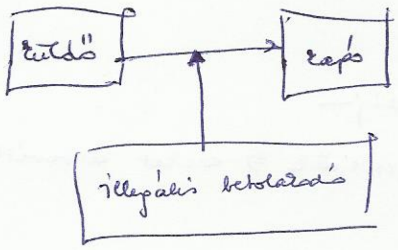
Kriptográfia = titkosítás, üzenetek rejtjelezésével (kódolás) foglalkozó tudomány.

- különben: matematikai eszközökkel biztosítani, hogy a fontos információk csak kijelölt körben legyenek elérhetők. (ne illetékteleneknek)

Régen

- ie. 1500. – ékírásos tábla (agyagmáz – készítéséről)
- ie 475 – görögök és spártaiak
- itáliai reneszánsz: virágkorát éli
- XIV Lajos korában: 587 véletlenszerűen kiválasztott elemből álló kódot használtak
- 1800-as években: E. A Poe novelláiban
- Morse távíró ⇒ Morse-kód
- I. vh: több nemzet készített mechanikus kódoló gépet
- Ezeket viszonylag rövid idő alatt lehetett dekódolni. ⇒ dekódolás nehezebbé válik a kódoló gépek idejében (II. vh: ellopták az ellenség kódoló gépét. :))

Üzenetküldéshez kapcsolódó szemlélet:



Kriptográfia részei

I. klasszikus kriptográfia

II. nyilvános kulcsú kr.

véges módszer ⇒ kulcs feltörni

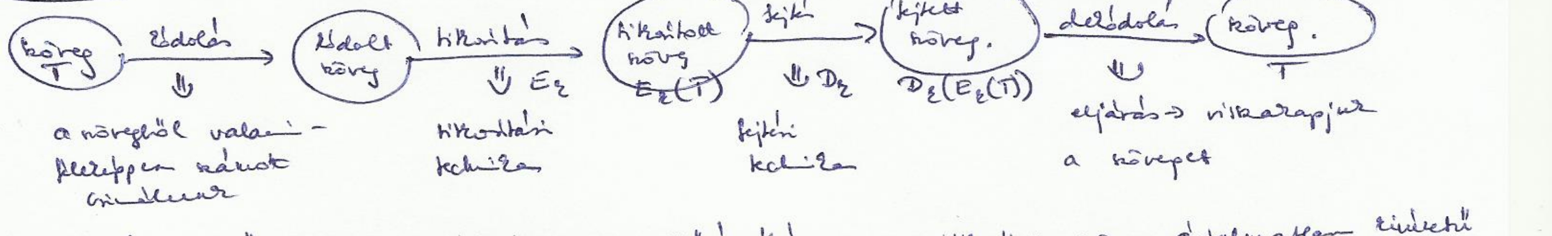
I. – régi korokban
- ha ismerjük a titkosítási módszert, visszafejthető (fejtési módszer: statisztikai módszer)
- a titkos szöveg ált-ban BETŰKBŐL ÁLL.

II. – modern titkosítási technika
- nehezen fejthető (még a módszer ismeretében is)
- Stat. módszerek nem működnek
- bitsorozatokból (esetleg számokból) áll a szöveg.

Nem mat-i módszerek:

- láthatatlan tinta, kopasz fej, code book.

Szöveg útja:



A jó titkosítás: – könnyű legyen a titkosított szöveg előállítása, – a titkosított szöveg értelmetlen legyen,
– kódoló kulcs ismeretében a fejtés könnyű legyen ⇒ fejtő kulcs hiányában a fejtés lehetetlen legyen

Fejtés

lehetséges, ha – elég nagy titkosított szöveg áll rendelkezésre
– $(T, E_k(T))$ pár ismeretében
– ha legális kapónak tüntetjük fel magunkat

Monoalfabetikus rendszerek:

1) Ceasar-féle titkosítás:

Lényege: A kulcsot az abc betűjének egy másik abc-vel való helyettesítésével (a nyílt ABC-hez viszonyítva 3 betűvel eltolta)

Pl.:

nyílt abc	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
rejtjel abc	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

BUDAPEST = EXGDSHVW

2) KULCSSZAVAS Ceasar titkosítás

kulcsszó: LIUS

nyílt abc	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
rejtjel abc	l	i	u	s	a	b	c	d	e	f	g	h	j	k	m	n	o	p	q	r	t	v	w	x	y	z

TOTHSEBI = RMRDQAIE

3) Polybios módszere:

↳ 3. pun háború nagy római hadvezérnek volt a tanácsadója

Lényege: az abc betűit magánhangzó párokkal helyettesítjük ⇒ ezeket észrevétlenül elrejtjük szövegben.

Pl.: Ica aludt, aki eladott egy uborkagyalut, itthon süvöltök. U

	A 1	E 2	I 3	O 4	U 5
1 A	A	B	C	D	E
2 E	F	G	H	I	J
3 I	K	L	M	N	O
4 O	P	Q	R	S	T
5 U	U	V	X	Y	Z

Páronként kigyűjtjük a szövegből a magánh-kat!
IA, UA, IE, AO, EU, OA, AU, IO, UU, OU
(sor, oszlop)

K U L D J   P E N Z T !

4) Hill - módszer:

Válasszunk egy tetszőleges mátrixot:

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 3 & 5 \\ 2 & 4 \end{pmatrix}$$

Titkosítandó szó: TEVE

$$T_1 = \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} T \\ E \end{pmatrix} = \begin{pmatrix} 20 \\ 5 \end{pmatrix} \qquad (T_2) = \begin{pmatrix} V \\ E \end{pmatrix} = \begin{pmatrix} 22 \\ 5 \end{pmatrix}$$

abc	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
	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	26

$$M \cdot T = \begin{pmatrix} g \\ h \end{pmatrix} = T'$$

$$T_1' = M \cdot T_1$$

$$\begin{pmatrix} 3 & 5 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} 20 \\ 5 \end{pmatrix} = \begin{pmatrix} 85 \\ 60 \end{pmatrix} \bmod 26 = \begin{pmatrix} 7 \\ 8 \end{pmatrix} = \begin{pmatrix} G \\ H \end{pmatrix}$$

$$T_2' = M \cdot T_2$$

$$\begin{pmatrix} 3 & 5 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} 22 \\ 5 \end{pmatrix} = \begin{pmatrix} 91 \\ 64 \end{pmatrix} \bmod 26 = \begin{pmatrix} 13 \\ 12 \end{pmatrix} = \begin{pmatrix} M \\ L \end{pmatrix}$$

TEVE → GHML

Pl:

$$M = \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \qquad T_1 = \begin{pmatrix} A \\ N \end{pmatrix} = \begin{pmatrix} 1 \\ 14 \end{pmatrix}$$

Titk. szó: ANYA

$$T_2 = \begin{pmatrix} Y \\ A \end{pmatrix} = \begin{pmatrix} 25 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \begin{pmatrix} 1 \\ 14 \end{pmatrix}: \begin{matrix} 2+42 \\ 4+70 \end{matrix} = \begin{pmatrix} 44 \\ 74 \end{pmatrix} \bmod 26 = \begin{pmatrix} 18 \\ 22 \end{pmatrix} = \begin{pmatrix} R \\ V \end{pmatrix}$$

$$\begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix} \begin{pmatrix} 25 \\ 1 \end{pmatrix} = \begin{pmatrix} 53 \\ 105 \end{pmatrix} \bmod 26 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} A \\ A \end{pmatrix}$$

ANYA → RVAA

Legális fejtő:

ismeri: M^{-1}

$$M T_1 = T_1'$$

$$M^{-1} \cdot M \cdot T_1 = M^{-1} \cdot T_1' = T_1$$

Titkosítsuk a CICA szót, majd fejtsük vissza:

$$T_1 = \begin{pmatrix} C \\ I \end{pmatrix} = \begin{pmatrix} 3 \\ 9 \end{pmatrix} \rightarrow T_1' = \begin{pmatrix} I \\ 15 \end{pmatrix} = \begin{pmatrix} G \\ O \end{pmatrix}$$

$$T_2 = \begin{pmatrix} C \\ A \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \rightarrow T_2' = \begin{pmatrix} 9 \\ 3 \end{pmatrix} = \begin{pmatrix} I \\ C \end{pmatrix}$$

} CICA ⇒ GOIC

GOIC:

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 2 & 3 \\ 7 & 8 \end{pmatrix}$$

$D = 21 \pmod{26}$ $\quad D^{-1} = 5$

$$A^{-1} = \begin{pmatrix} D^{-1}d & -D^{-1}b \\ -D^{-1}c & D^{-1}a \end{pmatrix} = \begin{pmatrix} 5\cdot 8 & -5\cdot 3 \\ -5\cdot 7 & 5\cdot 2 \end{pmatrix} \pmod{26} = \begin{pmatrix} 14 & 11 \\ 17 & 10 \end{pmatrix}$$

$$\begin{pmatrix} 14 & 11 \\ 17 & 10 \end{pmatrix} \left| \begin{pmatrix} 98+165 & 126+33 \\ 119+150 & 153+30 \end{pmatrix} \right. \pmod{26} = \begin{pmatrix} 3 & 3 \\ 9 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} C \\ I \end{pmatrix} \begin{pmatrix} C \\ A \end{pmatrix}$$

$\begin{pmatrix} 7 & 9 \\ 15 & 3 \end{pmatrix} \rightarrow \begin{pmatrix} G \\ O \end{pmatrix}$ $\quad \begin{pmatrix} I \\ C \end{pmatrix}$

5) Affin titkosrendszer:

Változnak a, b-t. $\quad 0 \le a, b < 25$

$(a, 25) = 1$

$\alpha = E(\alpha) = a\alpha + b \pmod{25}$

$a\alpha + b \pmod{25}$

$a\alpha' + b \pmod{25}$

ez két lép akkor egyezik meg, ha $a(\alpha - \alpha') \equiv 0 \pmod{25}$

POLIALFABETIKUS MÓDSZEREK

1) Playfair módszer:

kódolandó szöveg:

HIDDEN|NAP HETFO|

(ékezet nélkül)

↓

PK|XO|TO|CO|AR|TN|EA

S	Y	D	W	Z
R	I	P	U	L
H	C	A	X	F
T	N	O	G	E
B	K	M	J	V

szabály: • ha a két betű 1 sorban van ⇒ jobbra csúszás

• ha a két betű 1 oszlopban van ⇒ lefelé csúszás

2) Kulcsszavas Playfair:

Követelmény: A kulcsszóban ne legyen betűismétlés és legyen elég hosszú.

K	U	N	H	A
R	C	O	S	B
D	E	F	G	I
J	L	M	P	T
V	W	X	Y	Z

MI | ND | EN | NA | P H | ET | FO.

FT | FK | UF | HK | YS | LI | MF.

3) Blaise de Vigenère

A	B	C	D	E	F	G	H	
B	C	D	E	F	G	H	I	
C	D	E	F	G	H	I	J	
D	E	F	G	H	I	J	K	

LENNI VAGY NEM LENNI

Kulcsszó: MARS

MARSM ARSM ARS MARSM → oszlop
LENNI VAGY NEM LENNI → sor

⇓

XEEFU URYK NVE XEEFU

⇓

L. sor M. eleme

4) Autoclave Cardano

Szöveg: Aki mer az nyer.
Kulcs: CERUZA AKIMER

} Vigenère táblázatból

MINDIG AZ EREDETI SZÖVEG A SOR!

CO2GDR AJ UKII ⇒ kódolt szöveg

NYILVÁNOS KULCSÚ RENDSZEREK

1) Knapsack (hátizsák) módszer: (Merkle - Hellmann -féle algoritmus)

Nyilvános kulcsú kriptorendszerben sok névvel ellátott, nyitott lakat van ⇒ kulcsa a tulajdonosánál. (Bezárás után csak ők tudják kinyitni)

Üzenet küldésekor a megfelelő névvel ellátott lakatot rátesik a ládára és elküldik. Kulcsot nem kell küldeni, a tulajdonosnál van.

Szükséges: nyitott lakatok és kulcsaik kiosztása.

Nyilvános kulcsú rendszerek:
- órarend
- utazóügynök problémája
- hátizsák probléma
- parkolási probléma

Pakolási probléma:

Adottak: • különböző nagyságú hátizsákok } ⇒ úgy pakoljuk be, hogy a
• apróságok legnehezebb hátizsák teljen!

$A = \{a_1; a_2; a_3 \dots a_n\}$ ⇒ súlyok, apróságok

a_i-t kifejezni kell, hogy $\sum_{i=1}^{n} a_i < a_j \quad (j = 2 \dots n)$

A hátizsákba α súlyú felmérés fér:

$(t, m) = 1 \qquad t < m \qquad m) \sum_{i=1}^{n} a_i$

$b_i = (t \cdot a_i \bmod (m)) \qquad i = 1, 2 \dots n$

$B = (b_1, b_2 \dots b_n)$

(A, t, m, B) nyilvános kulcsú hátizsákrendszer

(t, m) : titkos kulcs, B : nyilvános kulcs.

Visszafejtés: $\alpha = (t^{-1} \cdot \beta \bmod (m)) \wedge a_i\ (t^{-1} \cdot b_i \bmod (m))$

Az értékek meghatározása után az (A, α) hátizsákfeladat megoldásával.

(A, α) hátizsákfeladat:

Adott: $A = (a_1, a_2 \dots a_n) \quad n \geq 3$

α

Keressük: $X = (x_1, x_2 \dots x_n)$, amelyre igaz $x_1 \cdot a_1 + x_2 a_2 + \dots x_n a_n = \alpha$

Pl.: $A\ (2, 6, 9, 18, 36)$

$\alpha = 56$

Relációk	X
$56 > 36$	1
$20 > 18$	1
$2 < 9$	0
$2 < 6$	0
$2 \geq 2$	1

2) RSA algoritmus:

1978. R. Rivest, A. Shamir ∧ L. Adleman

- egy nyílt és egy titkos kulcs tartozik
- a nyílt kulcs segítségével kódolhatjuk mások a nekünk szánt üzeneteket
- csak a titkos kulccsal tudjuk megfejteni az üzeneket.

Kulcsgenerálás:

① véletlenszerűen választunk két nagy prímet: p-t és q-t. (min 100decimális jegy!)

② $N = pq \Rightarrow$ ez lesz a modulusa a két kulcsnak.

③ kiszámítjuk az Euler-fgv értékét N-re: $\varphi(N) = (p-1)(q-1)$

④ választunk $e \in \mathbb{Z}$-t, amelyre: $\underbrace{(e, p-1) = (e, q-1) = 1}_{\text{relatív prímek}} \wedge (e, \varphi(N)) = 1$

⑤ Keressünk olyan d számot, amelyre: $d \cdot E = 1 \pmod{\varphi(N)}$ $\quad 1 < d < \varphi(N)$,
azaz: $d \cdot E = 1 + t \cdot \varphi(N)$

⑥ d-t titokban tartjuk, mint a Bob titkos kitevője.

⑦ A nyilvános kulcs N modulusból és E nyilvános kitevőből áll. ($p, q, \varphi(N)$ már nem kell.) megőrizzük, mert a gyors kiszámolását tudjuk lehetővé E által.

- Az N szám bináris alakban írt bitjeinek a száma adja a rejtjelzőkód komplexitását, ami a gyakorlatban általában: $n = 512;\ 1024;\ 2048$ között lenni.
pl. népszerű választás: $E = 2^{16} + 1 = 65537$.

Kódolás:

1) Alice (A) továbbítja a nyilvános kulcsát (N, E) Bob (B) felé, titkos kulcsát őrzi!
B ezután kinek küldeni üzenetét (M) A-nak.

2) M-et számokra alakítja (pl: ASCII kódból), és úgy daraboljа, hogy $m < N$.
Ezután kiszámolja c kódszöveget: $c = m^E \pmod N$
Ez gyorsan végrehajtható az ismételt négyzetre emeléses hatványozással. ⇒ B továbbítja üzenetét A-nak.

Dekódolás:

A ezután saját titkos kulcsát, d-t használva tudja visszafejteni m-et c-ből:

$$m = c^d \pmod N$$

Pl:

	p	q	N	$\varphi(n)$	E	d
A	11	23	253	220	17	13
B	13	19	247	216	65	113

$\varphi(N) = (11^{10} - 1)(23^{22} - 1)$
$\varphi(n) = (13^{12} - 1)(19^{18} - 1)$

Titkosítsuk a TITOK szót. (ASCII táblázat)
Minden blokk 1 betű és a kód nem haladhatja meg a modulus értékét.

A titkosít B-nek

M_i	ASCII kód	C_i
T	84	145
I	73	47
T	84	145
O	79	105
K	75	56

$C_i = (M_i^E, \text{mod}(n))$
$C_i = (M_i^{65}, \text{mod}(247))$
$84^{65} \bmod (247) = 145$

B visszafejt

C_i	ASCII kód	M_i
145	84	T
47	73	I
145	84	T
105	79	O
56	75	K

$(C_i^d, \text{mod}(n)) = M_i$
$M_i = (C_i^{113}, \text{mod } 247)$
$145^{113} \equiv 84 \pmod{247}$

3) Digitális aláírás:

A aláírja a TITOK követ. ⇒ ehhez göngyölített XOR művelettel kiszámítja az MD kivonatot.

XOR	0	1
0	0	1
1	1	0

M_i	kód	
T	84	1010100
I	73	1001001
T	84	1010100
O	79	1001111
K	75	1001011
MD	77	1001101

(0 XOR 1) = 1
1 XOR 0 = 1
1 XOR 1 = 0
0 XOR 1 = 1

MD = 77.

Az aláírást: $DS = (MD^d \bmod(n))$

$DS = (77^{13} \bmod(253)) = 110$ értéket adja

Ellenőrzés: $KP = (E, n)$

$KP = (17, 253)$ nyilvános kulcs ismeretében $(DS, \bmod(n)) = MD$

$MD = (110^{17}, \bmod(253)) = 77.$ számítás adja

Az aláírás és a titkosítás egymástól független ⇒ lehet küldeni a titkosított és aláírt üzenetet is.

4) Prímtesztek:

a) álprímek tesztelése:

$Z_n^+ = \{1, 2 \dots n-1\}$ Ha n prím ⇒ $Z_n^+ = Z_n^*$

Azt mondjuk, hogy az n szám „a" alapú álprím, ha n összetett és

$$a^{n-1} \equiv 1 \ (\bmod(n))$$

```
álprím (n)
if moduláris hatványozás (2, n-1, n) ≢ 1 (mod(n))
   then return összetett   ⇒ BIZTOS!
   else return prím        ⇒ REMÉLJÜK!
```

Mindig helyes, ha n összetett. Csak akkor hibázik az algoritmus, ha n szám 2 alapú álprímszám.

b) Miller-Rabin valószínűségi prímteszt:

Legyen $n > 2$ és n páratlan, $n-1 = 2^s \cdot r$, ahol r páratlan

ha $(a, n) = 1 \wedge 1 < a < n \Rightarrow a^r \equiv 1 \ (\bmod(n)) \vee a^{2^{s'} \cdot r} \equiv -1 \ (\bmod(n))$ $\forall\, 0 < s' < s$ esetén

Teszt:

$(a, n) = 1 \Rightarrow a^{p-1} \equiv 1 \ (\bmod(p))$ → ha hamis ⇒ összetett

→ ha igaz ⇒ sorozatos gyökvonást végzünk.

c) Tanik algoritmus

d) faktorizálás (Fermat-féle faktorizálás)

e) Polland-féle heurisztikus módszer (maradéktételt használ)

DES titkosítás:

(Data Encryption Standard)

- Carl Meyer és Walter Tuchman fejlesztette a Luciferből (IBM) 1970-ben
- 1977-ben vált szabványá
- szintén Feistel titkosításnak is hívni.
- 64 bites blokkos algoritmus: a nyílt szöveg egy 64 bites blokkjához egy ugyanilyen blokkot rendel hozzá.
- Minden lépésben az előző lépést használja kulcstól függően (egy ilyen lépést körnek nevezünk)
- a körök száma a használatos algoritmus jellemzője.
- A DES kulcsmérete 64 bit, de minden 8.-at elhagyjuk a felhasználásból. (ellenőrzési célja van)

kulcsméret 56 bit.

(pl: 13 34 57 79 9B BC DF F1 ⇒ kulcs)

1. lépés: a bemenet bitjeit jól összekeverjük, és utolsó lépésben ennek az inverzét alkalmazzuk. (DES titkosításban: inicializációs permutáció (IP)) és egy bitpermutáció a 64 bites vektorokon ⇒ független a kulcstól.

2. lépés: Ezután 16 körös Feistel módszert alkalmazunk

3. lépés: Végül a titkosított szöveget az IP inverzével kapjuk.

$$C = IP^{-1}(R_{16}, L_{16})$$

Használunk benne S_i függvényeket = S-dobozokat

helyettesítéseket végez

- nem lineáris
- minden s-doboz reprezentálható egy táblázattal (4 sor, 16 oszlop)
- ha $B_i = b_1b_2b_3b_4b_5b_6$ string esetén $S_i(B_i)$ értéke egy számolható ki, hogy b_1b_6 lesz a sorindex $b_2b_3b_4b_5$ oszlopindex (ha nincs elég, használunk további 0-kat, hogy a hossza 4 legyen.)

P-doboz: felcserélést végez

Kulcsok a DES-ben:

- Egy Des kulcs: $\in \{0,1\}^{64}$

AES Rijndael szimmetrikus kulcsú algoritmus

- 2000-ben publikálták
- Advanced Encryption Standard
- Elvárások:
  - szimmetrikus kulcsú, blokkos algoritmust kell megvalósítani
  - 128-as blokkokat kell használnia
  - 128-192-256 bites kulcsmérettel kell dolgoznia
  - gyorsabb legyen, mint 3DES és nyújtson jobb védelmet
  - a kg erőforrásait hatékonyan használja
  - legyen elég flexibilis, alkalmazkodjon jól a különböző platformok lehetőségeihez.

- Rijndael algoritmus:
  - nagyon stabil
  - ellenáll a mai támadási módszereknek
  - az egyetlen lehetséges módszer az összes lehetőség végigpróbálása (BRUTE-FORCE támadás)

- Alapok:
  - helyettesítő és lineáris transzformációkat ötvöző módszer
  - ismétlődő körfüggvények 4 egymástól független transzformációból állnak (ezek lesznek a rétegek)
  - a lineáris keverőréteg feladata a dobozok nagyfokú keverésének megvalósítása: a MixColumns nevű réteg
  - A nem lineáris réteg egyetlen S-dobozt használ (SubBytes réteg)
  - A kulcsfüggő réteg egyszerű XOR műveletet használ és minden körben más-más körkulcsot (RoundKey)

1. Round (State, RoundKey)
   a) SubBytes (State)
   b) ShiftRows (State)
   c) MixColumns (State)
   d) AddRoundKey (State, RoundKey)

2. FinalRound (State, RoundKey)
   a) SubBytes (State)
   b) ShiftRows (State)
   c) AddRoundKey (State, RoundKey)

A körfüggvény rétegei:

- State struktúra: ebben tárolják a bemenő és kimenő adatokat
- minden négyzet 1-1 byte-t jelent
- state-struktúra oszlopvektorait szintként kezelhetjük, szavaknak

s_{00}	s_{01}	s_{02}	s_{03}
s_{10}	s_{11}	s_{12}	s_{13}
s_{20}	s_{21}	s_{22}	s_{23}
s_{30}	s_{31}	s_{32}	s_{33}

SubBytes transzformáció:

- nemlineáris, invertálható S-doboz
- minden bájt helyettesítése nyparassal az S-dobozzal történik
- b_i: az adott bájt i-dik bitje, c_i az i-dik bitje $C = 01100011_{(2)}$ számnak

MixColumns transzformáció:

- Polinomok maradékos osztását használja
- state-struktúra bájtjait alakítja át, a bájtokat egy előre meghatározott polinommal szorozza meg.
- minden új bájt függ az eredeti bájt oszlopban lévő összes bájttól.
- kis változás 1 bájtban a kép teljes megváltozását vonja maga után.

AddRoundKey transzformáció:

- ez a réteg teszi kulcsfüggővé a titkosítási módszert
- a művelet egy egyszerű XOR az eddig kialakult struktúra és a körkulcs bájtjai között
- az általunk megadott titkos kulcsból csak kezdő, ún. kiterjesztett kulcsot készít az algoritmus.

- a kiterjesztett kulcsot a state-mátrixával egyező méretű darabokra kell vágni
- az első körkulcs, azaz az első N_B darab jó a nulladik körhöz
- a körkulcsokban a szavak rendre az első, 2., 3., ill 4. oszlophoz tartoznak, azaz ilyen sorrendben kell elvégezni a XOR műveletet.

Az AES jól működik különböző platformokon és az általa nyújtott titkosság is eléri a kívánt szinvonalat. :)

Hash titkosítás

Def: • A hash algoritmusok egyirányú kódolási rutinok, amelyek a bemeneti adatból a köv. feltételek teljesülése mellett képeznek kimeneti adatot.

- adott bemeneti adatból mindig ugyanazt a kimenetet adja.
- a kimeneti adat egyértelműen utal a bemeneti adatra, de a kimeneti adatból nem állítható elő a bemeneti adat.
- a bemeneti adat legkisebb változása más teljesen más kimenetet eredményez.

Felhasználás - azonosítási rendszerek:

- A hash algo azon tulajdonsága, hogy a kimeneti adatból nem állítható elő a bemeneti adat, alkalmassá teszi biztonságos azonosítási rendszerek fejlesztésére.

Betörésészlelés:

- Mivel a hash algo-ra általában jellemző, hogy a bemeneti adat méretéhez képest a kimeneti adat elhanyagolhatóan rövid ⇒ fájlrendszerben történő változások detektálására is használható.
- A fájlrendszer összes fájljának hash kimenetét tároljuk, majd újbóli ellenőrzés esetén a fájlra újra lefuttatjuk a hash algoritmust. Amennyiben a két kimenet egyezik, a fájl nem változott az utolsó ellenőrzés óta.

Digitális aláírás:

Az interneten böngészve gyakran találhatunk olyan oldalakat, ahol a letölthető fájl mellett megadják annak valamilyen hash algoritmus által generált kimenetét. Ha a letöltött fájlra lefuttatjuk az algoritmust, ellenőrizhetjük, hogy valóban a szerző által „aláírt" fájlt kaptuk-e meg.

Feltörésének lehetőségei:

- Brute force
- Dictionary attack (a lehetséges jelszavak közül csak azokat teszteli, amelyek értelmes szavakból állnak. Ez ún. szótárfájl segítségével oldható meg. E módszer ellen úgy védekezhetünk, ha csak bizonyos karaktereket tartalmazó karakterláncokat fogadunk el jelszóként.)

MD4: Message - Digest Algorithm
Ronald Rivest (1990)
Az MD5 és SHA1 algoritmusokat befolyásolta

MD5:

- 128 bit = 16 bájt, 32 hexadigit
- adatintegritás ellenőrzésére előszeretettel használják
- Ronald Rivest (1991) készítette az MD4 kiváltására
- 1996-ban hibát talált az algoritmusban, amíg nem javították ki, addig egyéb (SH1) algoritmust használtak.
- 2004: MD5 rendben van
  Ezután már használták az SSL ill. digitális aláírás algoritmusokban is.

SHA1:

- NSA készítette az USA-ban
- Standardként publikálták
- 160 bit, 20 bájt, 40 hexadigit
- 4 titkosító algoritmus (SHA-0, 1, 2, 3)
- SHA1 nagyon hasonlít az SHA-0-ra, de kijavították a hibáit.
- széles körben elterjedt, használják
- 2005: kriptoanalízisek szerint az SHA1 nem elég biztonságos a folyamatos használatra a lehetséges támadások miatt.

- ütközéses támadás (collision attack), amikor ismer 2 olyan nyílt szöveget, melyekre: MD5(k1) = MD5(k2)
- csak a számítási tejesítményen múlik

Algoritmus:

- 128 bites kimenet
- a bemeneti fájl 512 bites blokkokra tördelése
- szükség esetén az utolsó blokkot kiegészíteni 512 bitre
- 1 db nem lineáris F fgv van benne.
- Továbbfejlesztés: MD6 (ami lehet 256 bites)

- 2010: SHA2
- 2012: SHA3 fejlesztések és a kriptográfiai algoritmusok erősítése

Digitális aláírás:

- a nyílt kulcsú titkosítórendszerek egy lehetősége, amellyel hagyományos aláírást helyettesíthetünk az informatika világában.
- Igazolni tudjuk az aláíró személyét és azt, h. a dokumentum az aláírás óta nem változott.
- A digitális aláírás a dokumentum egy jellemző kivonatát, ellenőrzőösszegét tartalmazza, így biztosítja azt, hogy az aláírás ne legyen átvihető más dokumentumra.
- A digitális aláírás, közvetlenül logikailag kapcsolódik a dokumentumhoz, és az ellenőrzőösszeg miatt a módosítás könnyen felfedezhető.
- A digitális aláírás csak akkor hamisítható, ha a titkos kulcsot megszerzi valaki. ⇒ mivel nem hamisítható, letagadhatatlan.
- Az aláírás tartalmaz egy ellenőrzőösszeget, amihez szükség van egy MD algoritmusra (hashfgv.) és általában: MD5 v. SHA1.
- Ehhez hozzáfűzzük az aláíró nevét v. azonosítóját, az aláírás idejét, az MD algoritmus nevét.
- Ezután az aláíró kódolja a titkos kulcsával. (csak az ő nyilvános kulcsával olvasható el)
- Ezt a kódot csatolnunk kell az üzenethez, így kell elküldeni
- Ha a címzett meg akar győződni az üzenet hitelességéről, akkor kell készíteni egy ellenőrzőösszeget ugyanazzal az MD algoritmussal, amit az aláíráskor is használtak, és dekódolnia kell az aláírást a küldő nyilvános kulcsával. Az abban található kódot össze kell hasonlítani az újjal.
  Ha megegyeznek ⇒ a küldő írta alá az üzenetet és az nem változott meg az aláírás óta (feltételezzük, h. nem került illetéktelenek kezébe a kulcs)
- Törvény: az elektronikus aláírást szabályozó törvény 2001. szeptemberében lépett hatályba. (teljesen egyenrangúvá teszi a hagyományos aláírással)

Jellemzők:
- technológia független
- könnyen létrehozható és ellenőrizhető
- nem hamisítható és letagadhatatlan
- hitelesíti a dokumentum tartalmát és az aláíró személyét
- nyilvános kulcs személyhez kötődésének valódisága biztosított
- elektronikus kézbesítés biztosított.

PKI: Public Key Infrastructure

- a digitális aláírás a kétkulcsos titkosítás alkalmazására épül.
- lehetővé teszi a törvények által elfogadott kétkulcsú harmadik személyes hitelesítési és adatbiztosítási eljárások használatát.
- a saját és nyilvános kulcsok hitelesek, annak valódiságát és az adott félhez való tartozását hiteles 3. fél garantálja!