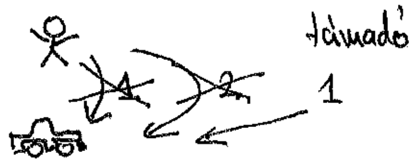
Kódoláselmélet 2020 szeptember 9.

Kódolás:

üzenet → $\boxed{\text{kódoló}}$ → $\boxed{\text{csatorna}}$ → $\boxed{\text{dekódoló}}$ → üzenet

1. titkosítás / hitelesítés (RSA, DH, MD5, SHA)
2. tömörítés (Huffman, ZIP, JPG, MP3)
3. hiba javítás / észlelés (CRC, BCH, turbo, LDPC) QR kód

A csatorna más az egyes esetekben



1. valaki hozzáfér az adatokhoz (lehallgat, megismétel, blokkol, beír)
2. drága az adatátvitel (idő, tár)
3. hiba keletkezik valamilyen ~~adathibákkal~~ eloszlással (módosít, beír, töröl, stb)

Általában több kódoló / dekódoló van egymás után fűzve: tömörítés + titkosítás + hibajavítás

alkalmazás: mobil telefon

Titkosítás: algoritmus + kulcs (paraméterek)

Kerckhoff-elv (tapasztalati észrevétel)

A titkosítás biztonsága nem függhet az algoritmus titkosságától (érdemes azt publikussá tenni) hanem csak a kulcs titkosságán alapulhat.

ROT 13: 26 latin betű, mindegyiket 13 hellyel arrébb toljuk, szóköz megmarad

0	1	2			12
A	B	C	…	L	M
↕	↕	↕	…	↕	↕
N	O	P		Y	Z
13					

Önmagának inverze, spoilereket kódolják vele online, nagyon gyönge "titkosítás"

Caesar: algoritmus $\varphi_c : \mathbb{Z}_{26} \to \mathbb{Z}_{26}$

$$a \mapsto a + c$$

betűnként végezzük el

$c \in \mathbb{Z}_{26}$ kulcs

$\boxed{\begin{array}{l} \pi \in S_{26} \text{ kulcs} \\ a \mapsto a\pi \end{array}}$

Hogyan lehet feltörni?

Vigenère 1553-ban találták fel, 1863-ban kriptoanalízis (módszer a feltörésre)

```
  ATTAC|KATDA|WN
+ LEMON|LEMON|LE
-----------------
  LXFOP|VEFRN|HR
```

i j

␣AZ␣   ␣AZ␣

????   ????

....

$(i \equiv j \mod k)$

Hogyan törjük fel?

kulcs: $c = (c_0, c_1, \dots, c_{k-1}) \in \mathbb{Z}_{26}^{k}$

$\varphi : \mathbb{Z}_{26}^{n} \to \mathbb{Z}_{26}^{n}$ ~~nával~~ üzeneten

$$a \mapsto b$$

$$b_i = a_i + c_{i \bmod k} \pmod{26}$$

Vernam kódolás

kulcs: $c \in \mathbb{Z}_{26}^n$

algoritmus $\varphi: \mathbb{Z}_{26}^n \to \mathbb{Z}_{26}^n$

$a \mapsto b$ $\qquad b = a + c$

egyszer használatos

$b_i = a_i + c_i$

Bizonyíthatóan nem törhető fel a kulcs ismerete híján: ugyanaz a kódolt üzenet tetszőleges üzenet kódolása lehet ha jól választjuk meg a kulcsot

$$c = b - a.$$

Ha többször használjuk a kulcsot, akkor gyakoriságanalízissel feltörhető!

Enigma:

3–4 tárcsa segítségével nagyon hosszú periodicitású permutáció létrehozása.

$b_i = a_i + c_i$ ahol c_i kvázi random

- gyengesége: $c_i \neq 0$
- a "nincs változás" (keine besonderen Ereignisse) üzenetet nagyon sokszor küldték el
- sikerült megszerezni az egyik kódolót

Kulcsegyeztetés

- nemilyesen, telefonon (másik megbízható csatornán)
- IoT eszközöknél nagy probléma

Szimmetrikus kulcsú titkosítás:

ugyan az a kulcs a kódolásnál és a dekódolásnál

Publikus (nyílt) kulcsú titkosítás:

két külön kulcs van a kódoláshoz és a dekódoláshoz, az egyik publikus a másik titkos.

encoder

Diffie, Hellman 1975-ben tett javaslata

decoder

- $E: X \to X$ kódoló, publikus (alg + kulcs)
- $D: X \to X$ dekódoló, titkos (titkos kulcs)
- E és D egymás inverzei $E \circ D = id \Downarrow D \circ E = id$
- E és D gyorsan számítható
- D-t nem lehet gyorsan meghatározni E ismeretében sem.

Első nyílt kulcsú titkosítás az RSA

RSA titkosítás (Rivest-Shamir-Adleman, 1977)

Alapötlet: találhatók olyan e, d, n egészek, hogy $(m^e)^d \equiv m \pmod n$ minden m-re.

$$E: \mathbb{Z}_n \to \mathbb{Z}_n, \quad m \mapsto m^e \qquad D: \mathbb{Z}_n \to \mathbb{Z}_n, \quad m \mapsto m^d$$

$\mathbb{Z}_6$-ban

$2^0 = 1$ $2^4 = 4$
$2^1 = 2$
$2^2 = 4$
$2^3 = 2$

publikus kulcs: e, n

titkos kulcs: d (és n)

$D = E^{-1}$

gyors hatványozás (nehéz logaritmust számolni)

Kulcs generálás — keygen

① válasszunk két különböző primet p, q (elég nagyokat, prímtesztelés gyors) $\pi(n) \sim \frac{n}{\ln(n)}$ ezek titkosak

② $n = pq$ publikus (prímfaktorizáció lassú, de lásd a kvantum számítógépet)

③ keressünk $1 < e < \varphi(n)$ egészet, hogy $\text{lnko}(e, \varphi(n)) = 1$ e lehet prím, pl. $2^{16}+1$

$$\varphi(n) = \varphi(p) \cdot \varphi(q) = (p-1)(q-1) \text{ — titkos!!}$$

④ oldjuk meg az $\underline{ed \equiv 1 \mod \varphi(n)} = (p-1)(q-1)$ diofantoszi egyenletet az euklideszi algoritmus segítségével, így kapjuk d-t

$$\underline{e}x + \underline{\varphi(n)}y = 1$$

Tétel: p, q különböző prímek, $n = pq$

$(p-1)(q-1) \mid ed - 1$

Ekkor $m^{ed} \equiv m \pmod{n}$ minden m-re, azaz az RSA titkosítás működik.

Biz: mivel $p \neq q$ prímek, ezért elég megmutatni, hogy $m^{ed} \equiv m \pmod{p}$

① ha $m \equiv 0 \pmod{p}$ akkor világos

② ha $m \not\equiv 0 \pmod{p}$ akkor a kis Fermat-tételből $m^{p-1} \equiv 1 \pmod{p}$.

De $ed \equiv 1 \mod p-1$, ezért

$$m^{ed} \equiv m^1 \pmod{p}$$

$$ed = k(p-1)+1 \qquad m^{ed} = (m^{p-1})^k \cdot m^1 \equiv m$$

Feladat: Generáljunk RSA kódot

Kódoljunk be egy üzenetet

Dekódoljuk a kódolt üzenetet

Felhasználás: Ha ismerjük a titkos kulcsot, akkor

① bárki küldhet nekünk titkos üzenetet

② küldhetünk olyan üzenetet amelyről bárki meggyőződhet hogy a titkos kulcs tulajdonosától jött