# Információs rendszerek biztonságtechnikája

Kriptográfia
Szimmetrikus kódú algoritmusok

*Vakulya Gergely*

# A kriprográfia négy célja

## Titkosság (secrecy, confidentality)

Annak biztosítása, hogy az üzenetet harmadik fél **ne tudja elolvasni**.

## Hitelesség (authentication)

Annak bizonyíthatósága, hogy az üzenet valóban a **feladótól származik**.

## Letagadhatatlanság (nonrepudiation)

Annak bizonyíthatósága, hogy az üzenetet a **feladó valóban elküldte**.

## Sértetlenség (integrity)

Annak biztosítása, hogy az üzenetet harmadik személy **ne tudja megváltoztatni**.

# A kriptográfia modellje

## Jelölések

- Nyílt szöveg (plaintext), $P$
- Kulcs (key), $K$
- Titkosított szöveg (cyphertext), $C$

$$C = E_K(P)$$

$$P = D_K(C)$$

$$D_K(E_K(P)) = P$$

## Kerchoff elve

**Minden algoritmusnak nyilvánosnak kell lennie; csak a kulcsok titkosak (1883)**

# Kriptoanalízis

## Csak titkosított szöveg alapú támadás

A kódtöréshez csak egy, vagy több titkosított szöveget ismerünk. A legnehezebb szituáció.

## Ismert nyílt szöveg alapú támadás

Nyílt szöveg - titkosított szöveg párokat ismerünk. Talán ez a leggyakoribb eset.

## Választott nyílt szöveg alapú támadás

Lehetőségünk van tetszőleges nyílt szöveget titkosítani (tehát a nyílt szövegekhez előállítani azok titkosított szöveg párját), de magát a kulcsot nem ismerjük. Ez az eset nyilvános kulcsú algoritmusok, illetve „blackbox” titkosítók esetében fordul elő.

# Az ismeretlenség biztonsága (security by obscurity)

A rendszer (nem feltétlenül titkosítási eljárás) működésével kapcsolatban bizonyos gyengeségeket szándékosan nem hoznak nyilvánosságra azt remélve, hogy azokat nem fedezik fel / használják ki. Napellenzőbe „rejtett” forgalmi engedély, lábtörlő alá „rejtett” kulcs esete.

- Zárt forráskódú firmware-ek.
- A fájlrendszer mélyére „jól elrejtett” fájlokban tárolt érzékeny adatok.
- Beépített kiskapuk (például pótjelszavak).

**Rossz gyakorlat. Kerülendő!**

# A feltörhetetlen kód (one-time pad)

## A kizáró vagy (XOR) művelet

- $K \oplus K = O$
- $C = P \oplus K$
- $P = C \oplus K = P \oplus K \oplus K$
- Egy kulcs csak egyszer használható fel!

| $A$ | $B$ | $A \oplus B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

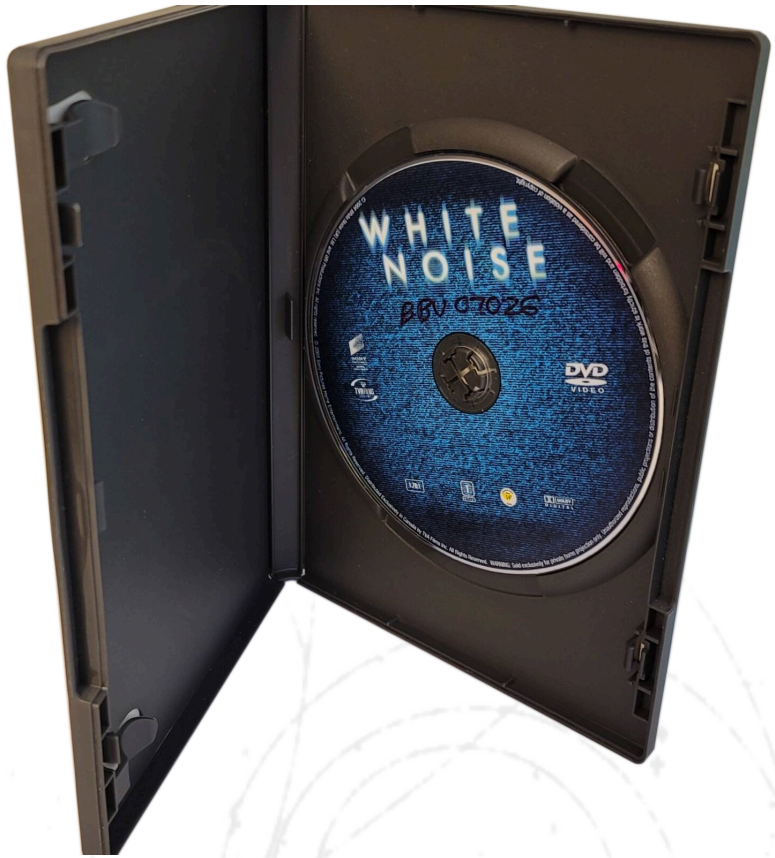
## A titkosítás menete

- Alice és Bob megállapodnak egy közös kulcsban.
- A kódolás akkor működőképes, ha a kulcs véletlenszerű bitekből áll és legalább akkora, mint a kódolandó adat.
- Például Alice generál egy DVD-nyi véletlenszerű bitet és a lemezt eljuttatja Bobnak.
- Ezt követően bármelyikük titkos kommunikációt tud folytatni a másikkal a kulcs (a DVD) méretéig bezárólag.

# A feltörhetetlen kód (one-time pad)

## Hátrányok

- A kulcs elkészítése problematikus.
  - A számítógép általában pszeudorandom szekvenciákat tud gyorsan generálni.
  - Ténylegesen véletlen számok előállítása nehéz / lassú.
- A kulcs megosztása, tárolása problematikus.
  - Elzárva kell tartani.
  - Könnyen lemásolhatják.

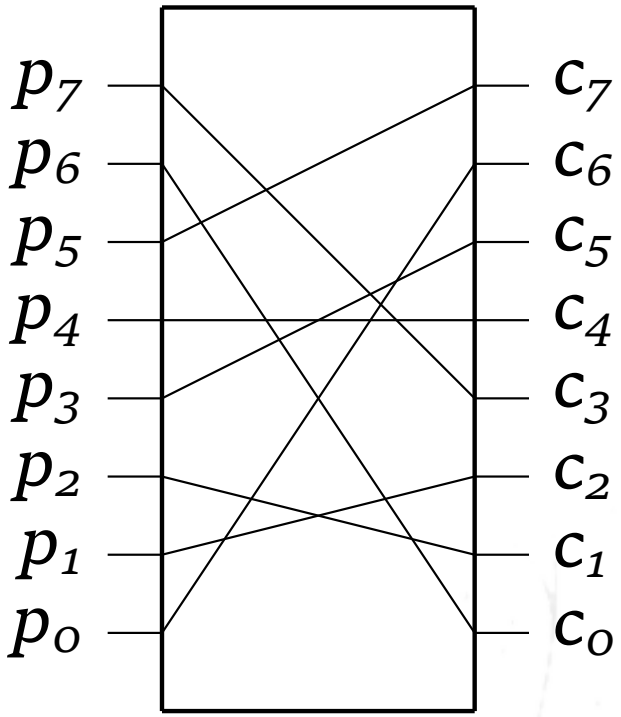


# Szimmetrikus kulcsú titkosítások

- A szimmetrikus kulcsú algoritmusok mind a kódoláshoz, mind a dekódoláshoz **ugyanazt a kulcsot** használják.
- A blokk kódolók $n$ bites blokkokra bontják a bemenetet és a kódolást / dekódolást ilyen egységenként végzik.
- Kívánatos, hogy egy $k$ bites nyílt szöveg egy szintén $k$ bites titkosított szöveget eredményezzen, viszont $k$ nem mindig egész számú többszöröse $n$-nek. Megoldás: Az utolsó blokk feltöltése, **padding**.
- A vevő oldalnak tudnia kell, hogy a használt kulcs helyes-e. Megoldás: **redundancia** alkalmazása (pl. ellenőrző összeg).

# P-doboz

- P: Permutate
- A bemenet bitjein keverést végez



# S-doboz

- S: Substitution
- Minden bitmintát egy másik bitmintára cserél

# Data Encryption Standard (DES)

# Tripe-DES (3DES)

# Az AES története

# Az AES főbb paraméterei

# Egyéb szimmetrikus kódú titkosító algoritmusok

# Elektronikus kódkönyv (ECB) mód

# TItkosított blokkok láncolása (CBC)

# Kimenet visszacsatolása (OFB)

# Számláló mód