

# Kriptográfia és biztonság 1.

## Bevezető

## Követelmények

- **bejárni** (kivéve esti, msc, felmentés)
- **gyakorlat** előfeltétel az előadáshoz
- **DiMat II** ∨ **DiMOA** előfeltétel a gyakhoz
- jegyszerzés: **írásbeli vizsga** a teljes félév anyagából
- **elővizsga opció:**
  - 3 alkalom, az 5., 10. héten és a vizsgaidőszak 1. hetében
  - kumulatív számonkérés
  - mindegyiket legalább 2-esre

## Témakörök

- matematikai háttér (algebra, számelmélet, valszám)
- kriptográfiai protokollok (titkosítás, hash, aláírások)
- támadások (elméleti, fizikai, ...)
- alkalmazások (poszt/kvantum kripto, kriptopénzek, ...)
- ma: néhány példa + alkalmazás **informálisan!** (vizsgán ne így mondd el...)

## Irodalom

- Járai: Bevezetés a matematikába - Informatikai alkalmazásokkal
- Buttyán, Vajda: Kriptográfia és alkalmazásai
- Schneier: Applied cryptography
- Katz, Lindell: Introduction to modern cryptography

## Alapkérdések

- mi a kripto protokoll célja?
- kik a résztvevők? ki mit tud/küld? (kommunikációs modell)
- mi a támadás? támadó ismerete, erőforrása? (fenyegetés modell)
- mi a védelem? (criptográfia)

## „Definíció”

A *protokoll* kettő vagy több résztvevő közötti lépések sorozata, adott feladat végrehajtására kidolgozva.

## Tulajdonságok

- $\forall$  résztvevő ismeri a protokoll lépéseit
- $\forall$  résztvevő beleegyezik, hogy követi a lépéseket
- $\forall$  lépés jól definiált
- teljes leírás:  $\forall$  helyzetre  $\exists$  lépés
- *kripto protokoll* extra tulajdonság: nem lehet több infot szerezni, mint amit a prot. meghatároz

## Egyszerű történelmi példa

- cél: üzenetküldés két fél között
- résztvevők:
  - küldő (írni tud, elküldi az üzenetet)
  - fogadó (olvasni tud)
- támadás: az ellenség megszerzi az üzenetet
- védelem:
  - ne szerezhesse meg (nyílt csatorna miatt nem reális)
  - hiába szerzi meg, ne tudja elolvasni
  - fizikai megoldás: lelakatolt láda 1-1 kulccsal

## Informális kripto megoldás

- Üzenet:  $m$ , titkosított szöveg:  $c$
- titkosító algoritmus:  $Enc(\cdot)$ , visszafejtő algoritmus:  $Dec(\cdot)$
- titkosítás:  $Enc(m) = c$ , visszafejtés:  $Dec(c) = m$

## Informális kripto megoldás

- Üzenet:  $m$ , titkosított szöveg:  $c$
- titkosító algoritmus:  $Enc(\cdot)$ , visszafejtő algoritmus:  $Dec(\cdot)$
- titkosítás:  $Enc(m) = c$ , visszafejtés:  $Dec(c) = m$

## Támadás

- támadó megszerzi  $c$ -t. Mit tud meg ebből? Hogyan védjük?

# Játék példa

## Informális kripto megoldás

- Üzenet:  $m$ , titkosított szöveg:  $c$
- titkosító algoritmus:  $Enc(\cdot)$ , visszafejtő algoritmus:  $Dec(\cdot)$
- titkosítás:  $Enc(m) = c$ , visszafejtés:  $Dec(c) = m$

## Támadás

- támadó megszerzi  $c$ -t. Mit tud meg ebből? Hogyan védjük?

## Védelem

- **A terv:** algok működését eltitkoljuk ( $Enc, Dec$  nem ismert)
- *Security by obscurity*: nagyon rossz irány...

## Informális kripto megoldás

- Üzenet:  $m$ , titkosított szöveg:  $c$
- titkosító algoritmus:  $Enc(\cdot)$ , visszafejtő algoritmus:  $Dec(\cdot)$
- titkosítás:  $Enc(m) = c$ , visszafejtés:  $Dec(c) = m$

## Támadás

- támadó megszerzi  $c$ -t. Mit tud meg ebből? Hogyan védjük?

## Védelem

- **A terv:** algok működését eltitkoljuk ( $Enc, Dec$  nem ismert)
- **Security by obscurity:** nagyon rossz irány...
- **B terv:** kulcsok használata ( $Enc_{k_e}(m) = c, Dec_{k_d}(c) = m$ )
- $k_e$  és  $k_d$  lehet ugyanaz vagy különböző is
- **Modern kriptográfia:** ez kell nekünk.

## Biztonság

- biztonság (confidentiality)
- azonosítás (authentication)
- integritás (integrity)
- letagadhatatlanság (non-repudiation)
- ...

## Feladat

Biztonságos üzenetküldés két résztvevő között

Ötlet: közös *szimmetrikus* titkos kulcs használata

- közös titkos kulcs légyártása + megosztása
- titkosítás + visszafejtés a közös kulccsal
- résztvevők szerepe is szimmetrikus

Szimmetrikus kulcsú titkosítás (informálisan)

- $m$  : üzenet,  $c$  : titkos szöveg,  $k$  : közös kulcs
- három algoritmusból áll:
  - ➊ Gen kulcsgenerálás:  $k$  kulcs kiválasztása
  - ➋ Enc titkosítás: adott kulcsból és üzenetből titkos szöveg:  
 $c = Enc_k(m)$
  - ➌ Dec visszafejtés: adott kulcsból és titkos szövegből üzenet:  
 $m = Deck_k(c)$

## Szimmetrikus kulcsú titkosítás (informálisan)

- ❶ *Gen* kulcsgenerálás:  $k$  kulcs kiválasztása
- ❷ *Enc* titkosítás:  $c = Enc_k(m)$
- ❸ *Dec* visszafejtés:  $m = Dec_k(c)$

## Problémák

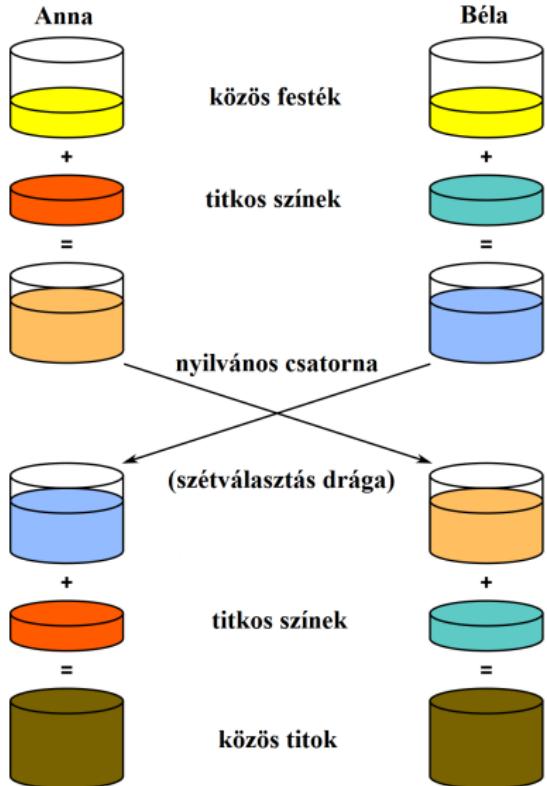
- kommunikáció védelme:
  - közös kulcs eljuttatása **védett** csatornán ( jó, de hogyan?)
  - a titkos szöveg átküldése **nyílt** csatornán
  - $c$  „elrejtse” az  $m$ -et
- tárolt adatok védelme:
  - kulcsok kezelése
- skálázhatóság:
  - minden küldő-fogadó párhoz másik kulcs

## Feladat

Közös titkos kulcs kiszámítása  
két résztvevő között **nyilvános  
kommunikációval**

## Ötlet:

Közös info + saját titkos info  
(kulcsok) kombinálása



## Feladat

Biztonságos üzenetküldés két résztvevő között

Ötlet: külön kulcsok a titkosításhoz és a visszafejtéshez

- minden résztvevőnek *kulcspárok*: nyilvános + titkos kulcs
- nyilvános kulcsot publikáljuk, ezzel titkosítunk  $\Rightarrow$  bárki tud
- visszafejteni csak a titkos kulccsal

Nyilvános kulcsú titkosítás (informálisan)

- $m$  : üzenet,  $c$  : titkos szöveg,  $pk$  nyilvános,  $sk$  titkos kulcs
- három algoritmusból áll:
  - 1 Gen kulcsgenerálás:  $(sk, pk)$  kulcspár kiválasztása
  - 2 Enc titkosítás:  $c = Enc_{pk}(m)$
  - 3 Dec visszafejtés:  $m = Dec_{sk}(c)$

## Nyilvános kulcsú titkosítás (informálisan)

- ➊ *Gen* kulcsgenerálás:  $(sk, pk)$  kulcspár kiválasztása
- ➋ *Enc* titkosítás:  $c = Enc_{pk}(m)$
- ➌ *Dec* visszafejtés:  $m = Dec_{sk}(c)$

## Problémák

- kommunikáció védelme:
  - a titkos szöveg átküldése **nyílt** csatornán
  - $c$  „elrejtse” az  $m$ -et
- tárolt adatok védelme:
  - egyszerűbb kulcs-kezelés (csak  $sk$  érzékeny)
- skálázhatóság:
  - many-to-one kommunikáció egy kulcspárral!

## Feladat

Autentikált üzenet küldése + integritásvédelem

Ötlet: kézzel írt aláírás digitális megfelelője

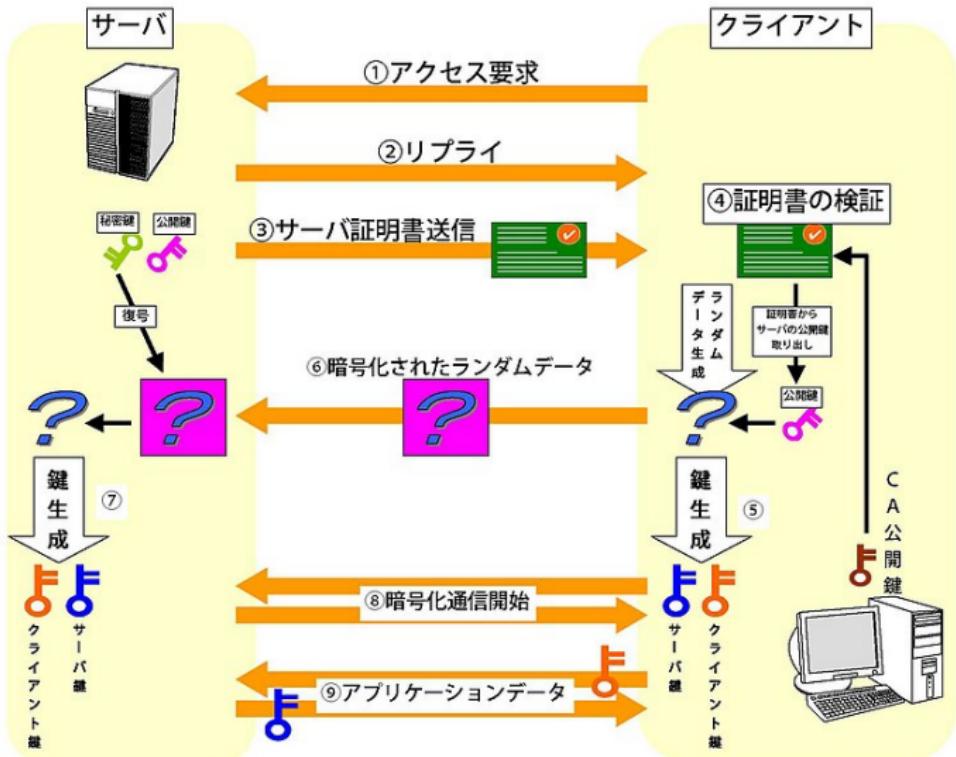
- tetszőleges  $m$  üzenethez  $s$  aláírás készítése
- $s$ -et ne(hezen) lehessen hamisítani
- bárki le tudja ellenőrizni, hogy  $s$  az  $m$  aláírása
- nyilvános-titkos kulcspárok, de fordított szerepben

Digitális aláírás (informálisan)

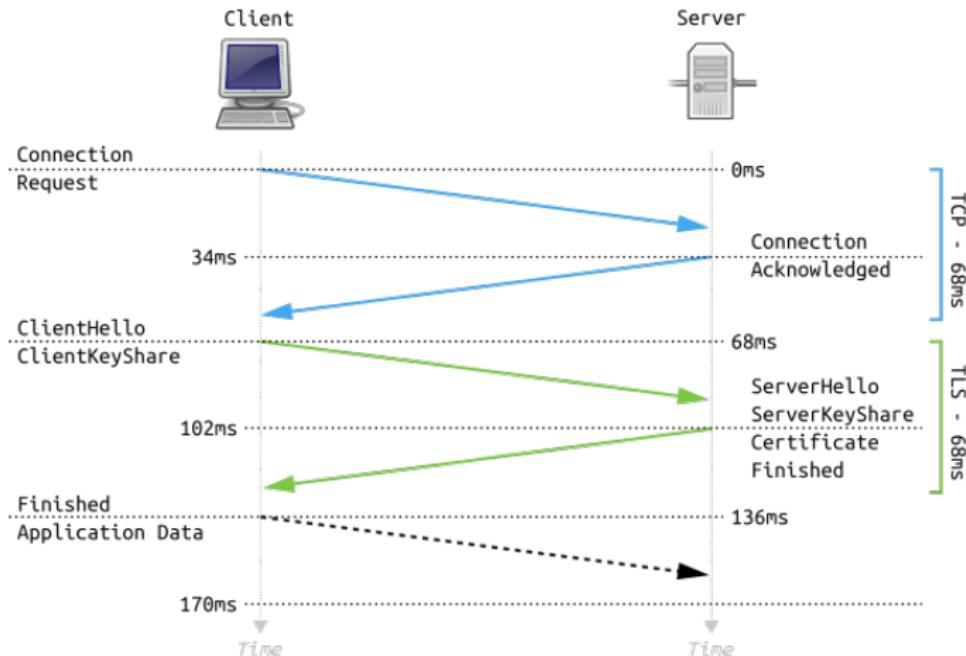
- $m$  : üzenet,  $s$  : aláírás,  $pk$  nyilvános,  $sk$  titkos kulcs
- három algoritmusból áll:
  - ➊ Gen kulcsgenerálás:  $(sk, pk)$  kulcspár kiválasztása
  - ➋  $Sign$  aláírás:  $s = Sign_{sk}(m)$
  - ➌  $Vrfy$  ellenőrzés:  $Vrfy_{pk}(m, s) = 1$ , ha  $s$  jó aláírás  $m$ -hez



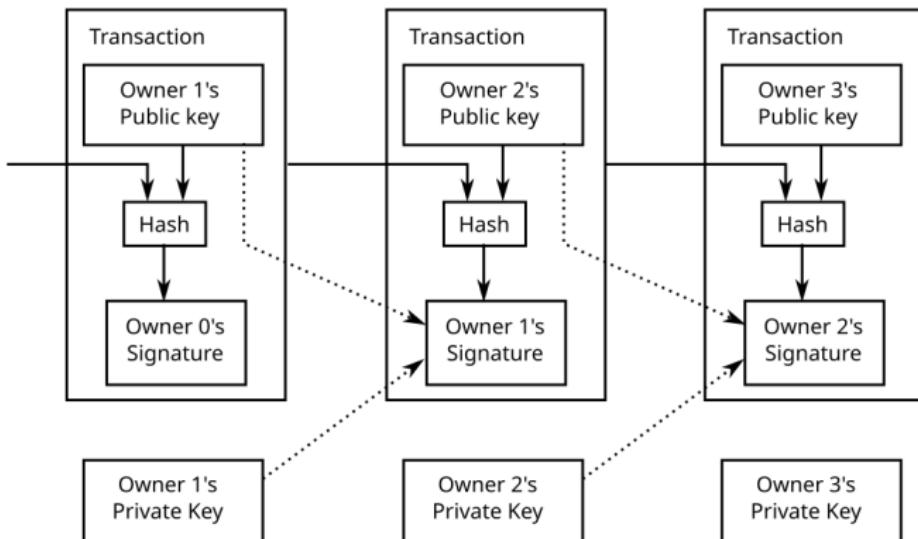
# Alkalmazás: TLS ( minden egyben)



# Alkalmazás: TLS ( minden egyben)



# Alkalmazás: Bitcoin



## Összegzés

- feladat definiálása, kommunikációs + fenyegetés modell
- kriptografiával: minden, mindenhol