Számítógépes Hálózatok
2008

12. Szállítói réteg – TCP, hatékonyság, fairness; Biztonság

Torlódás elkerülési elv: AIMD

- A TCP a „fast recovery” mechanizmussal lényegében a következőképp viselkedik:

  x: csomagok száma per RTT

- Kapcsolatfelépítés:

  $x \leftarrow 1$

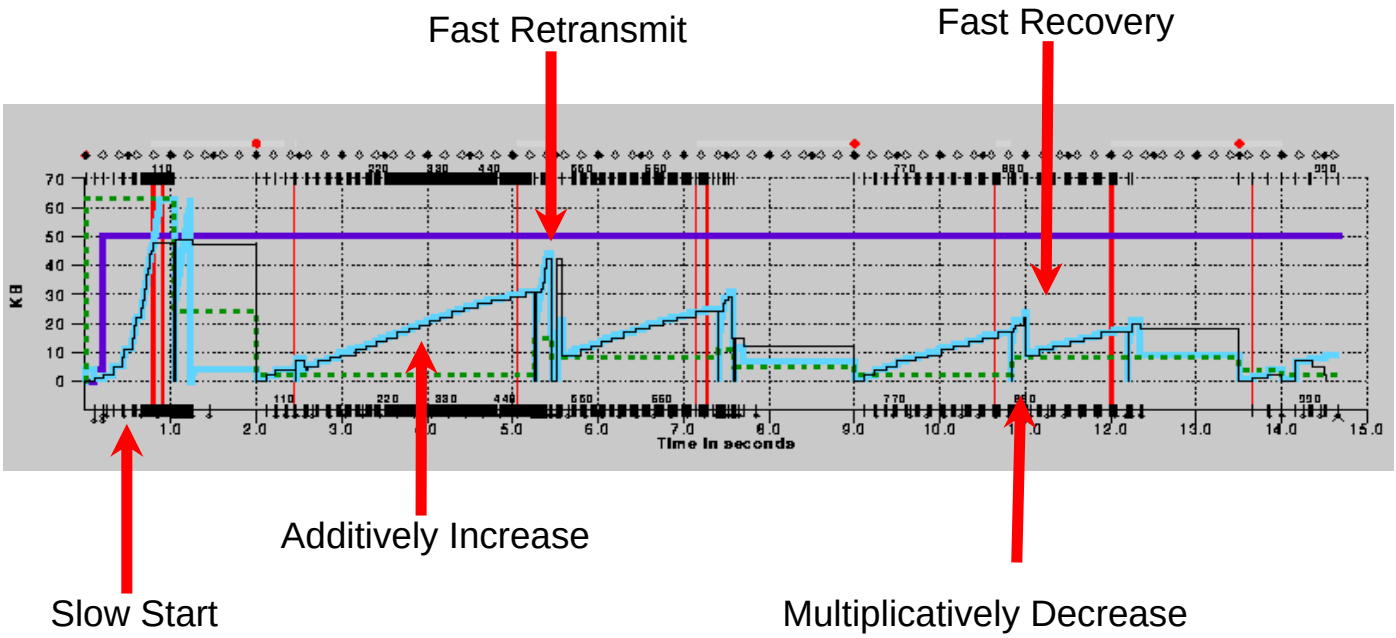
- Csomagvesztésnél, MD: multiplicative decreasing

  $x \leftarrow x/2$

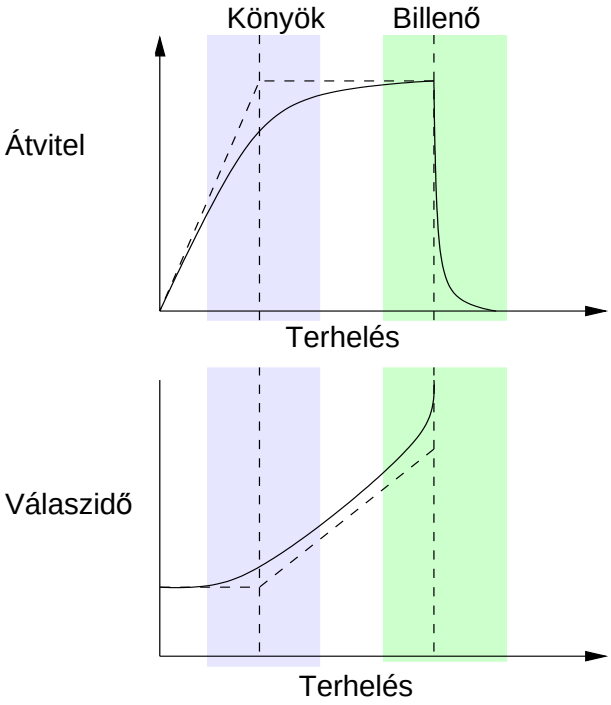
- Nyugtázott szegmenseknél, AD: additive increasing

  $x \leftarrow x + 1$

Példa: TCP Reno „in aktion”



Additive Increase Multiplicative Decrease (AIMD): Fairness és Hatékonyság



A hálózati terhelés az átvitellel és a válasziővel kölcsönösen hat egymásra.

- Az átvitel maximális, ha a terhelés a hálózat kapacitását majdnem eléri.
- Ha a terhelés tovább nő, túlcsordulnak a pufferek, csomagok vesznek el, újra kell küldeni, drasztikusan nő a válaszidő. Ezt a toródást congestion-nak nevezzük.
- Ezért a maximális terhelés helyett, ajánlatos a hálózat terhelését a könyök közelében beállítani. Itt a válaszidő csak lassan emelkedik, míg az adatátvitel már a maximum közelében van.
- Egy jó torlódáselkerülési (*congestion avoidance*) stratégia a hálózat terkelését a könyök közlében tartja: hatékonyság. Emellett fontos, hogy minden résztvevőt egyforma rátával szolgáljunk ki: fairness.

AIMD Fairness és Hatékonyság – Egy egyszerű modell

- n résztvevő, forduló-modell
- résztvevő i adatrátája a t-eik fordulóban $x_i(t)$
- Kezedeti adatráták: $x_1(0)$, …, $x_n(0)$
- A visszacsatolás (feedback) forduló t után: y(t) = 0, ha $\sum_{i=1}^{n} x_i(t) \leq K$

  y(t) = 1, ha $\sum_{i=1}^{n} x_i(t) > K$
- Minden résztvevő aktualizálja az adatrátáját a t+1-edik fordulóban:

$$x_i(t+1) = f(x_i(t),y(t))$$

  - Increase-stratégia $f_0(x) = f(x,0)$
  - Decrease-stratégia $f_1(x) = f(x,1)$

- Tekintsük a következő lineáris függvényeket :

$$f_0(x) = a_I + b_I x, \qquad f_1(x) = a_D + b_D x$$

AIMD Fairness és Hatékonyság– A Modell

- A következő lineáris függvényeket vizsgáljuk:

$$f_0(x) = a_I + b_I x, \qquad f_1(x) = a_D + b_D x$$

- Érdekes speciális esetek:
  - MIMD: Multiplicative Increase/Multiplicative Decrease

    $f_0(x) = b_I x, \quad f_1(x) = b_D x, \quad$ ahol $\quad b_I > 1,\ b_D < 1.$
  - AIAD: Additive Increase/Additive Decrease

    $f_0(x) = a_I + x, \quad f_1(x) = a_D + x, \quad$ ahol $\quad a_I > 0,\ a_D < 0.$
  - AIMD: Additive Increase/Multiplicative Decrease

    $f_0(x) = a_I + x, \quad f_1(x) = b_D x, \quad$ ahol $\quad a_I > 0,\ b_D < 1.$

AIMD Fairness és Hatékonyság

- Hatékonyság
  - Terhelés: $X(t) := \sum_{i=1}^{n} x_i(t)$
  - Mérték: $|X(t) - K|$
- Fairness: x=(x_1, …, x_n) esetén:

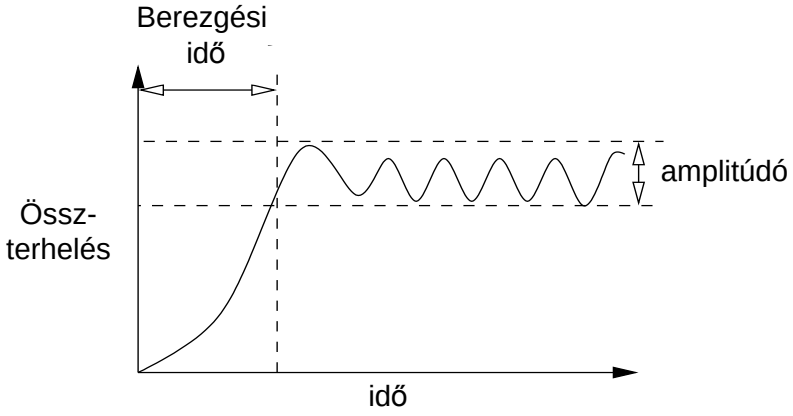
$$F(x) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} (x_i)^2} .$$

  - 1/n ≤ F(x) ≤ 1
  - F(x) = 1 ↔ absolut Fairness
  - skálázástól független
  - Folytonos, differenciálható
  - Ha n közül k fair, a többi 0, akkor F(x) = k/n

Konvergencia

- Konvergencia nem lehetséges
- Legjobb esetben oszcilláció az optimális érték körül
  - Az oszcilláció amplitúdója A

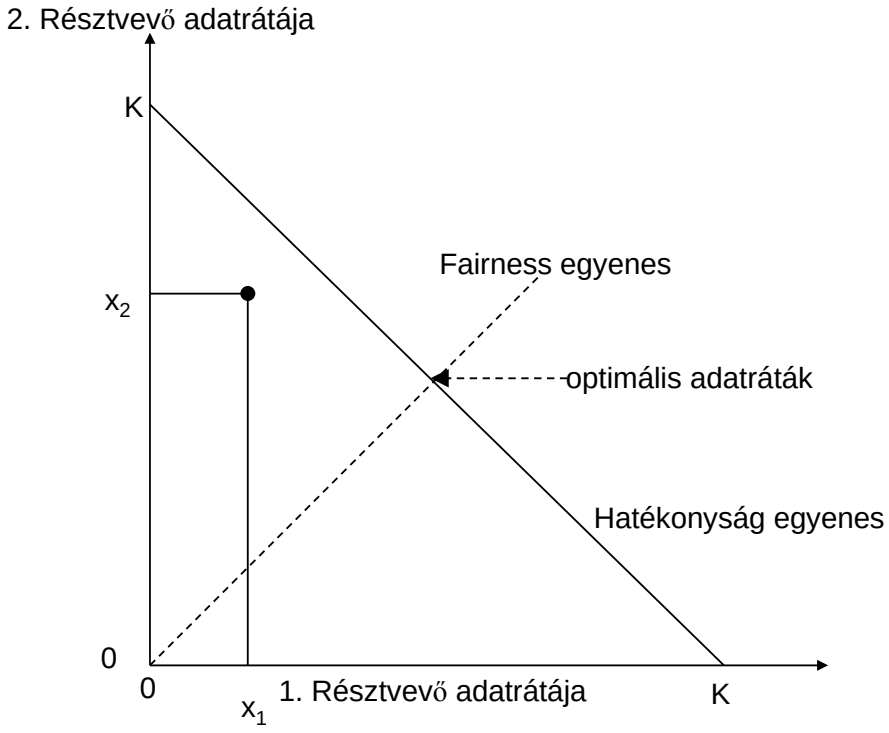
$$A = \inf_{t_0 \geq 0} \sup_{t \geq t_0} |X(t) - K| .$$



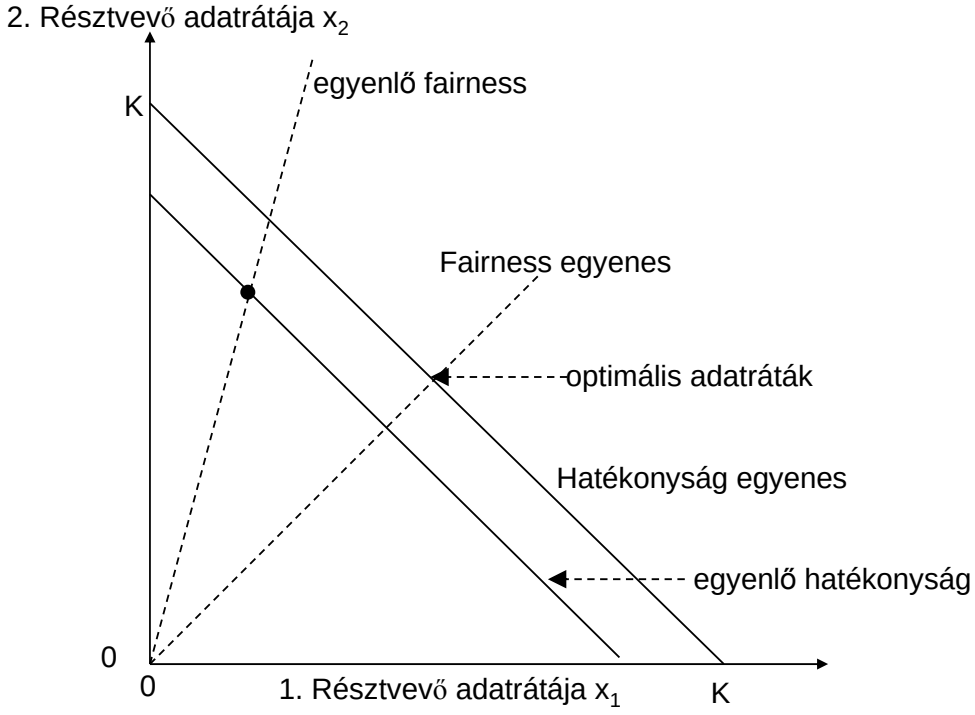
  - Berezgési idő T

$$T = \min\{t_0 \mid \forall t \geq t_0 \ : \ |X(t) - K| \leq A\} .$$

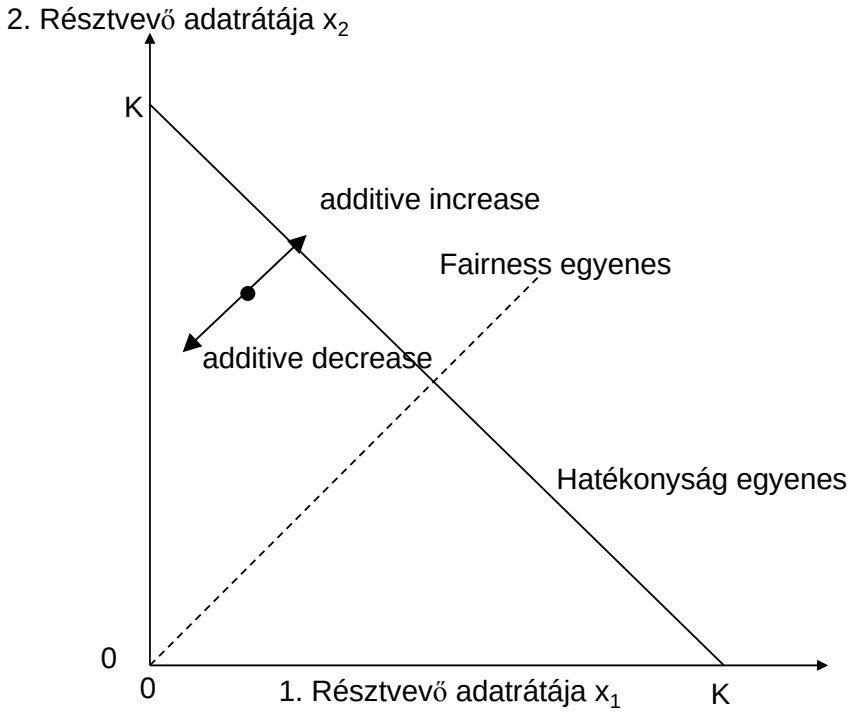
Vektor Ábrázolás (I)



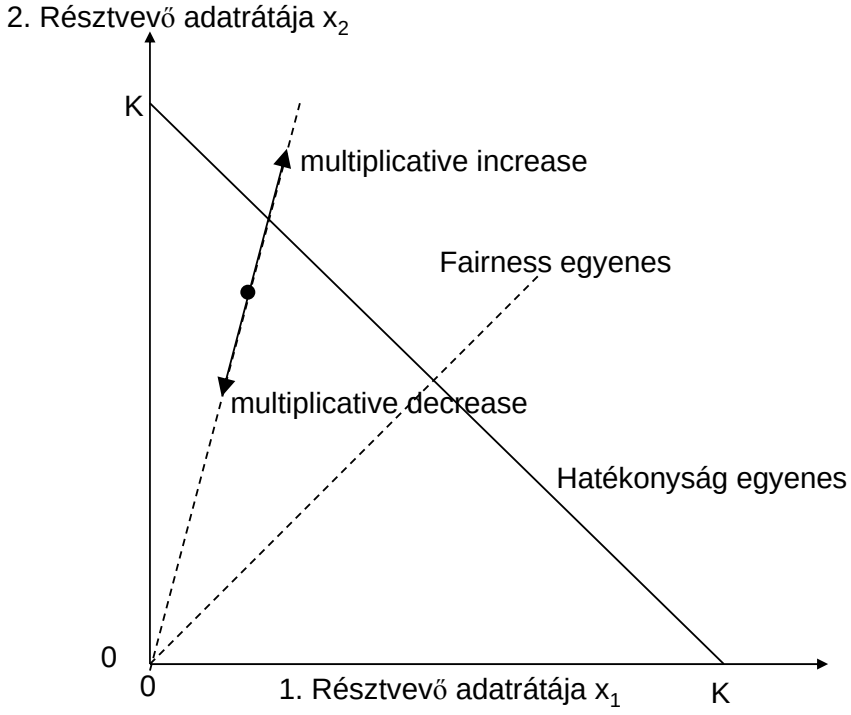
Vektor Ábrázolás (I)



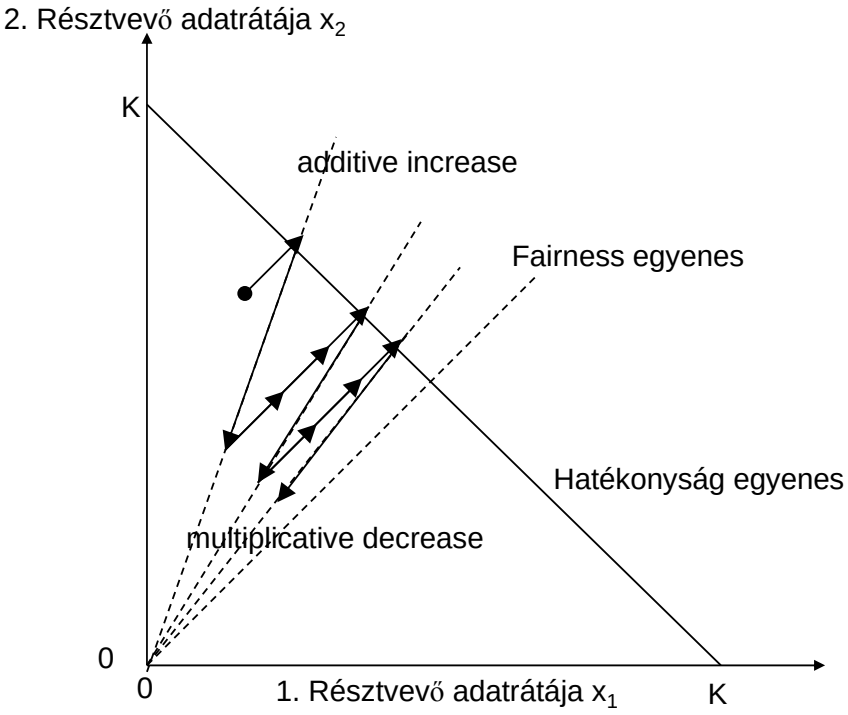
Vektor Ábrázolás (I)



Vektor Ábrázolás (I)



Vektor Ábrázolás (I)



TCP összefoglalás

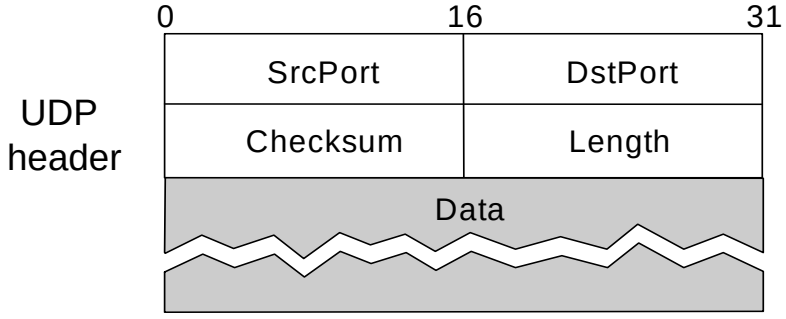
- TCP egy megbízható byte-folyamot hoz létre
  - Hibafelügyelet “Go-Back-N” által
- Congestion control
  - Ablak alapú
  - AIMD, Slow start, *Congestion Threshold*
  - Folyamfelügyelet *Window* által
  - Kapcsolatfelépítés
  - Nagle algoritmusa

TCP fairness & TCP friendliness

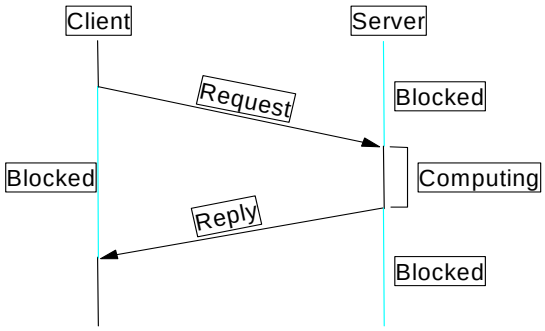
- TCP
  - Dinamikusan reagál a rendelkezésre álló sávszélességre
  - A sávszélesség fair felosztása
    - Idealis esetben: n TCP-kapcsolat mindegyike 1/n részt kap
- TCP más protokollokkal
  - Reakció más szállítói protokollok terhelésétől függ
    - pl. UDP-ben nincs congestion control
  - Más protokollok mindenkor felhasználhatók
  - UDP és más protokoll el tudja nyomni a TCP kapcsolatokat
- Végkövetkeztetés
  - A szállítói protokolloknak TCP-kompatibilisnek kell lenni (TCP friendly)

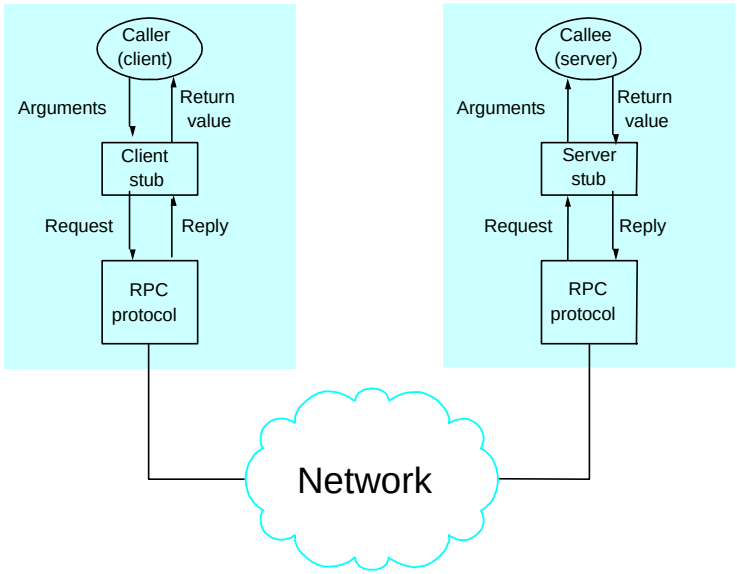
UDP

- User Datagram Protocol (UDP)
  - Egy nem megbízható kapcsolat nélküli szállítói réteg protokoll csomagoknak
- Fő funkció:
  - A hálózati réteg csomagjainak demultiplexálása
- Egyéb funkció (opcionális):
  - Checksum: UDP header + data



Remote procedure call – Struktúra





- Komplexebb interakcióhoz:
  - Egy függvény hívása egy másik számítógépen
- Cél: Transzparens protokoll a hívónak/hívottnak

Számítógéphálózatok biztonsága

- Szerepet játszik a következő rétegekben
  - Fizikai réteg
  - Adatkapcsolati réteg
  - Hálózati réteg
  - Szállítói réteg
  - Alkalmazási réteg
- Mi jelent fenyegetést (vagy támadást)
- Milyen módszerek vannak
  - Kryptográfia
- Hogyan védekezhetünk támadások ellen?
  - Példa: Firewalls

Számítógéphálózatok biztonsága

Fenyegetés, támadás

- Definíció:
  - Egy számítógéphálózat fenyegetése minden olyan lehetséges esemény vagy akciók sorozata, amely biztonsági célok megsértéséhez vezethet
  - A támadás a fenyegetés realizálása
- Példa:
  - Egy hacker betör egy zárt hálózatba
  - Az átfutó email-ek nyilvánosságra hozása
  - Idegen hozzáférés egy online bankszámlához
  - Egy hacker egy rendszer összeomlását okozza
  - Valaki autorizálatlanul tevékenykedik valaki más nevében (Identity Theft)

Biztonsági célok

- Bizalmaság (confidentiality):
  - Csak egy előre meghatározott publikum tudja írni vagy olvasni az átvitt vagy tárolt adatokat
  - A résztvevők azonosságának a bizalmassága: Anonimitás
- Adatintegritás (data integrity)
  - Adatok megváltoztatása kideríthető legyen
  - Az adatok szerzője felismerhető legyen
- Felelős hozzárendelhetősége (accountability)
  - Minden kommunikációs eseményhez hozzárendelhető legyen annak okozója
- Rendelkezésre állás (availability)
  - A szolgáltatások elérhetők legyenek és helyesen működjenek
- Kontrollált hozzáférés (controlled access)
  - A szolgáltatásokat és az információkat csak autorizált felhasználók érjék el

Támadások technikai definíciója

- Álarc (masquerade)
  - Valaki másnak adja ki magát
- Lehallgatás (eavesdroping)
  - Valaki olyan információt olvas, amit nem neki szántak
- Hozzáférési jog megsértése (Authorization Violation)
  - Valaki olyan szolgáltatást vagy erőforrást használ, ami nem neki szánt
- (Átvitt) Információ elvesztése vagy megváltoztatása
  - Az adatokat megváltoztatják vagy megsemmisítik
- Kommunikáció letagadása (denial of commmunication acts, repudiation)
  - Valaki (hamisan) letagadja a részvételét a kommunikációban
- Információ hamisítás (forgery of information)
  - Valaki más nevében állít elő (változtat) üzeneteket
- Sabotage
  - Minden olyan akció, amely a szolgáltatások vagy a rendszer helyes működését vagy rendelkezésre állását csökkenti

Fenyegetések és biztonsági célok

biztonsági cél	fenyegetés						
	álarc	lehallgatás	hozzáférési jog megsértése	információ elvesztése vagy megváltoztatása	kommunikáció letagadása	információ hamisítás	sabotage (pl. túlterhelés)
bizalmasság	x	x	x				
adatintegritás	x		x	x		x	
felelős hozzárendelhetőség	x		x		x	x	
rendelkezésre állás	x		x	x			x
kontrollált hozzáférés	x		x			x	

A kommunikációs biztonság terminológiája

- Biztonsági szolgáltatás
  - Egy absztrakt szolgáltatás, amely egy biztonsági tulajdonságot kíván biztosítani
  - Lehet kryptografikus protokollal vagy anélkül realizálni, pl.
    - Adatok titkosítása egy merev lemezen
    - CD a páncélszekrényben
- Kryptografikus algoritmus
  - matematikai transzformáció
  - kryptografikus (titkosító) protokollokban használt
- Kryptografikus protokoll
  - lépések és kicsrélendő üzenetek sora egy biztonsági cél eléréséhez

Biztonsági szolgáltatás

- Authentifikáció
  - Digitális aláírás: az adat bizonyíthatóan a létrehozótól származik
- Integritás
  - Biztosítja, hogy az adat ne legyen észrevétel nélkül megváltoztatható
- Bizalmasság
  - Az adat csak a fogadó által érthető
- Kontrollált hozzáférés
  - Biztosítja, hogy csak az arra jogosultak férjenek hozzá a szolgáltatásokhoz és információkhoz
- Letagadhatatlanság
  - Bizonyítja, hogy az üzenet letagadhatatlanul az előállítójától származik

Kryptológia

- Kryptológia
  - A titkos kommunikáció tudománya
  - A görög kryptós (rejtett) és lógos (szó) szavakból
  - Kryptológia részei:
    - Kryptográfia (gráphein = írás): titkos kommunikáció létrehozásának a tudománya
    - Krypto-analízis (analýein = megoldani, kibogozni): titkosított információ kibogozásának a tudománya

Titkosítási módszerek

- Szimetrikus titkosítási módszerek
  - pl. Caesar kód
  - Enigma
  - DES (Digital Encryption Standard)
  - AES (Advanced Encryption Standard)
- Kryptografikus Hash-függvények
  - SHA-1, SHA-2, MD5
- Aszimetrikus titkosítási módszerek
  - RSA (Rivest, Shamir, Adleman)
  - Diffie-Helman
- Digitális aláírás
  - PGP (Phil Zimmermann), RSA

Szimetrikus titkosítási módszerek

- pl. Caesar kód, DES, AES
- Léteznek f,g függvények, úgy hogy
  - titkosítás:
    - f(kulcs,szöveg) = kód
  - visszakódolás:
    - g(kulcs,kód) = szöveg
- A kulcsnak
  - titokban kell maradni
  - a küldő és a fogadó számára ismertnek kell lenni

Kryptografikus hash-függvények

- pl. SHA-1, SHA-2, MD5

- Egy kryptografikus hash-függvény h egy szöveget képez le egy fix hosszúságú kódra, úgy hogy
  - h(szöveg) = kód
  - és nincs olyan másik szöveg szöveg´, melyre:
    - h(szöveg´) = h(szöveg) és szöveg ≠ szöveg´
- Lehetséges megoldás:
  - Szimetrikus kryptográfikus módszerek felhasználása

Aszimetrikus titkosítási módszerek

- pl. RSA, Ronald Rivest, Adi Shamir, Lenard Adleman, 1977
  - Diffie-Hellman, PGP
- Privát kulcs `privat`
  - titkos, csak az üzenet fogadója ismeri
- Nyilvános kulcs `public`
  - minden résztvevő ismeri
  - egy függvény állítja elő
    - keygen(`privat`) = `public`
- Titkosító függvény f és visszakódoló függvény g
  - mindenki számára ismert
- Titkosítás
  - f(`public`,`text`) = `code`
  - minden résztvevő ki tudja számítani
- Visszakódolás
  - g(`privat`,`code`) = `text`
  - csak a fogadó tudja kiszámítani

Példa: RSA

- Az eljárás a prím-faktor felbontás nehézségére alapul
- 1. példa: 15 = ? * ?
  - 15 = 3 * 5
- 2. példa:

3865818645841127319129567277348359557444790410289933586483552047443
=
12345678901234567890123456789002 09 *
31313131313131313131313131313002 27

- Máig nem ismert hatékony eljárás a prím-faktor felbontásra
  - De két prím szorzata hatékonyan kiszámítható
  - Prím számok hatékonyan meghatározhatók
  - Prím számok gyakoriak

Az RSA-séma

1. Legyen p,q két nagy prím szám (1024-2048 bit)
2. Számítsuk ki n = p q
3. Számítsuk ki $\phi(n) = (p-1)(q-1)$ (Euler ϕ függvény)
4. Legyen e egy szám, $1<e<\phi(n)$, úgy hogy e és $\phi(n)$ relatív prím
5. Legyen d egy szám, melyre $e\,d = 1 \bmod \phi(n)$

- Privát kulcs (n,d)
- Visszakódolás:
  - $\mathtt{message} = \mathtt{code}^d \bmod n$
- Nyilvános kulcs: (n,e)
- Titkosítás:
  - $\mathtt{code} = \mathtt{message}^e \bmod n$
- Euler tétele:
  - $\forall$ m egészre, m és n relatív prím: $m^{\phi(n)} = 1 \bmod n$
- Helyesség (ha message és n relatív prím):

$(\mathtt{message}^e \bmod n)^d \bmod n$
$= \mathtt{message}^{e\,d} \bmod n = \mathtt{message}^{e\,d \bmod \phi(n)} \bmod n$
$= \mathtt{message} \bmod n$

Az RSA-séma

- RSA-séma helyessége:

$code^d \bmod n$
$= (message^e \bmod n)^d \bmod n$
$= message^{e\,d} \bmod n$

Mivel $ed = 1 \bmod (p-1)(q-1)$ és így
$ed = 1 \bmod (p-1)$ és
$ed = 1 \bmod (q-1)$,
a kis Fermat tétel (változat) alpján
$message^{ed} = message \bmod p$
és
$message^{ed} = message \bmod q$

Ekkor a Kínai maradék tétel miatt
$message^{ed} = message \bmod pq$

- Kis Fermat tétel:
  - $\forall$ p prímre és m egészre: $m^p = m \bmod p$
- Változat:
  - $\forall$ p prímre, m egészre, ha i,j pozitív egészek, $i=j \bmod p-1$, akkor $m^i = m^j \bmod p$
- Kínai maradék tétel:
  - $\forall$ $n_1,n_2,...,n_k$ egészre, melyek páronként relatív prímek, és $\forall$ $a_1,a_2,...,a_k$ egészre $\exists$ x egész, melyre $x = a_i \bmod n_i$, i=1,...,k. Továbbá minden ilyen x kongruens moduló $n=n_1n_2...n_k$.

RSA példa

- Két nagy prím szám 7,11
- n=77
- $\phi(n) = (p-1)(q-1) = 60$

- Privát kulcs (n,d):
  - d = 43, amelyre
    - $e * d = 1 \bmod \phi(n)$

- Visszakódolás:
  - $message = code^{43} \bmod 77$

$47^{43} \bmod 77$
$= 47^{1+2+8+32} \bmod 77$
$= 47*47^2*47^8*47^{32} \bmod 77$
$= 47*47^2*((47^2)^2)^2*((((47^2)^2)^2)^2)^2 \bmod 77$
$= ...$
$= 5 = message$

- Nyilvános kulcs (n,e):
  - n = 77 és egy szám e = 7

- Titkosítás:
  - $code = message^7 \bmod 77$
  - message = 5
  - $code = 5^7 \bmod 77 = 47$