# Kriptográfia

## Kilencedik előadás
## *A hitelesítésről általában*

Dr. Németh L. Zoltán
SZTE, Számítástudomány Alapjai Tanszék
2008 ősz

# Üzenet hitelesítés (Message Authentication)

- az üzenet hitesítésének összetevői:
  - az üzenet sértetlenségének biztosítása
  - az üzenet eredetének = feladójának ellenőrzése
  - letagadhatatlanság (viták eldöntése)
- három alternatív módszer használatos:
  - az üzenet titkosítása
  - üzenethitelesítő kód (message authentication code MAC)
  - hash-függvény (tördelő függvény, lenyomatképző függvény, hash function)

# Biztonságot kell nyújtani az alábbiakkal szemben

- tartalom felfedése (disclosure)
- forgalomelemzés (traffic analysis)
- megszemélyesítés (masquerade)
- tartalom módosítás (content modification)
- sorrend módosítás (sequence modification)
- üzenet idejének módosítása (timing modification)
- küldés letagadása (source repudiation)
- megkapás letagadása (destination repudiation)

# A hitelesítés alapszközei (primitívei)

1. Hitelesítés **titkosítással**
   - szimmetrikus kulcsú algoritmussal
   - nyílt kulcsú algoritmussal

2. **MAC** (üzenethitelesítő kód) algoritmusok

3. **Hash** (tördelő) függvények

# 1. Üzenet titkosítása I.

- maga az üzenet titkosítása is szolgálhat hitelesítésként
- ha **szimmetrikus titkosítást** használunk:
  - mivel a kulcsot csak a feladó és a címzett ismeri
  - a címzett biztos lehet a feladó személyében
  - és az üzenet sértetlenségében is (nem változtathatta senki se meg)
  - feltéve, hogy az üzenet elégséges struktúrával, vagy redundanciával rendelkezik, ahhoz hogy felismerjük ha megváltoztatták, pl. nem egyetlen véletlen kulcs önmagában

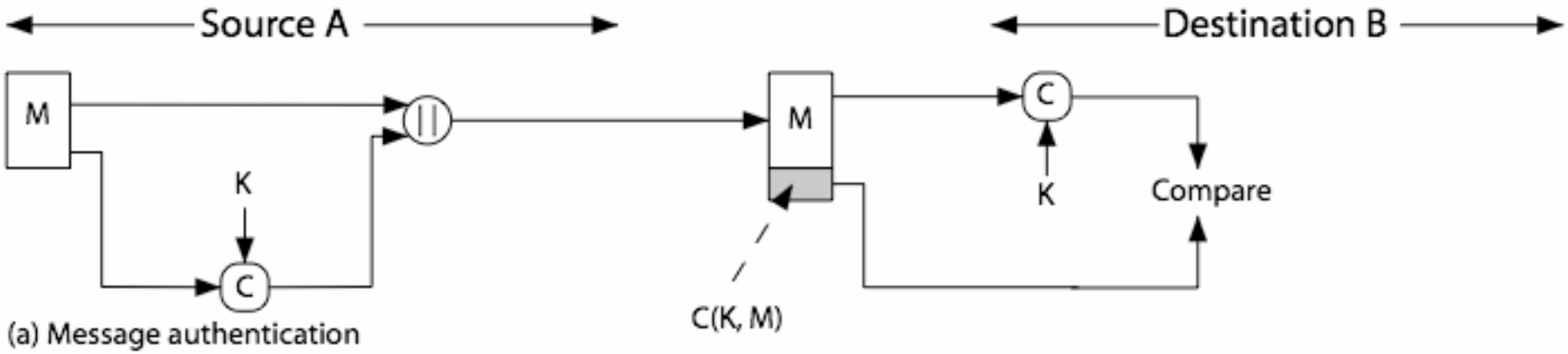
# 1. Üzenet titkosítása II.

- ha azonban **nyilvános kulcsú titkosítás** történt:
  - maga a titkosítás, nem igazolja a feladó személyét
  - hiszen a nyilvános kulcshoz bárki hozzáférhet
  - ha azonban
    - a feladó **aláírja az üzenetet a magánkulcsával**
    - majd a címzett nyilvános kulcsával titkosítja azt, akkor
    - mind a bizalmasság, mind a hitelesség fennáll
  - szintén meg kell tudni különböztetni az értelmes üzenetet a véletlen jelsorozattól
  - a költsége a teljes üzenet kétszeri aszimmetrikus titkosítása
  - a feleknek persze biztosnak kell lenni, hogy valóban egymás nyilvános kódját birtokolják (kulcselosztás ! )

# 2. Üzenethitelesítő kód
## Message Authentication Code (MAC)

- egy **MAC** algoritmussal (**C**) egy rövid, rögzített hosszú blokkot generálunk, mely függ
  - az üzenettől: **M** és
  - **egy közös titkos kulcstól**: **K**
- hasonló a titkosításhoz, de nem kell visszaállíthatónak lennie
- egyfajta kriptográfiai ellenőrző összeg
- az üzenet végére illesztjük és **MAC kódnak** (aláírásnak) hívjuk: **MAC = C(K,M)**
- a címzett a titkos kulcs és az üzenet birtokában újra kiszámítva a MAC kód értékét
- ellenőrizheti a feladó hitelességét és az üzenet sértetlenségét, beleértve az üzenet sorszámát (sequence number), stb.

# 2. MAC alkalmazása titkosítás nélkül




Itt II az egymás után illesztés jele.

# A MAC tulajdonságai

- a MAC kriptográfiai ellenőrző összeg (cryptographic checksum) $\mathbf{MAC = C_K(M)}$
  - összesűríti a tetszőleges hosszú M üzenetet
  - egy K titkos kulcs használatával
  - egy rögzített hosszú hitelesítő „pecsétre”
- **nem lehet egyértelmű függvény**
  - potenciálisan végtelen sok üzenet fog tartozni ugyanahhoz a MAC értékhez
  - de azt szeretnénk, hogy két ilyet nagyon nehéz legyen találni

# Követelmények

- figyelembe véve a támadási típusokat
- a **MAC**-nek telejítenie kell:
  1. az üzenet és a MAC ismeretében keresztülvihetetlen egy másik azonos MAC-kel rendelkező üzenetet találni
  2. a MAC-ek egyenletes eloszlásúak
  3. a MAC egyenlő mértékben függ az üzenet minden bitjétől

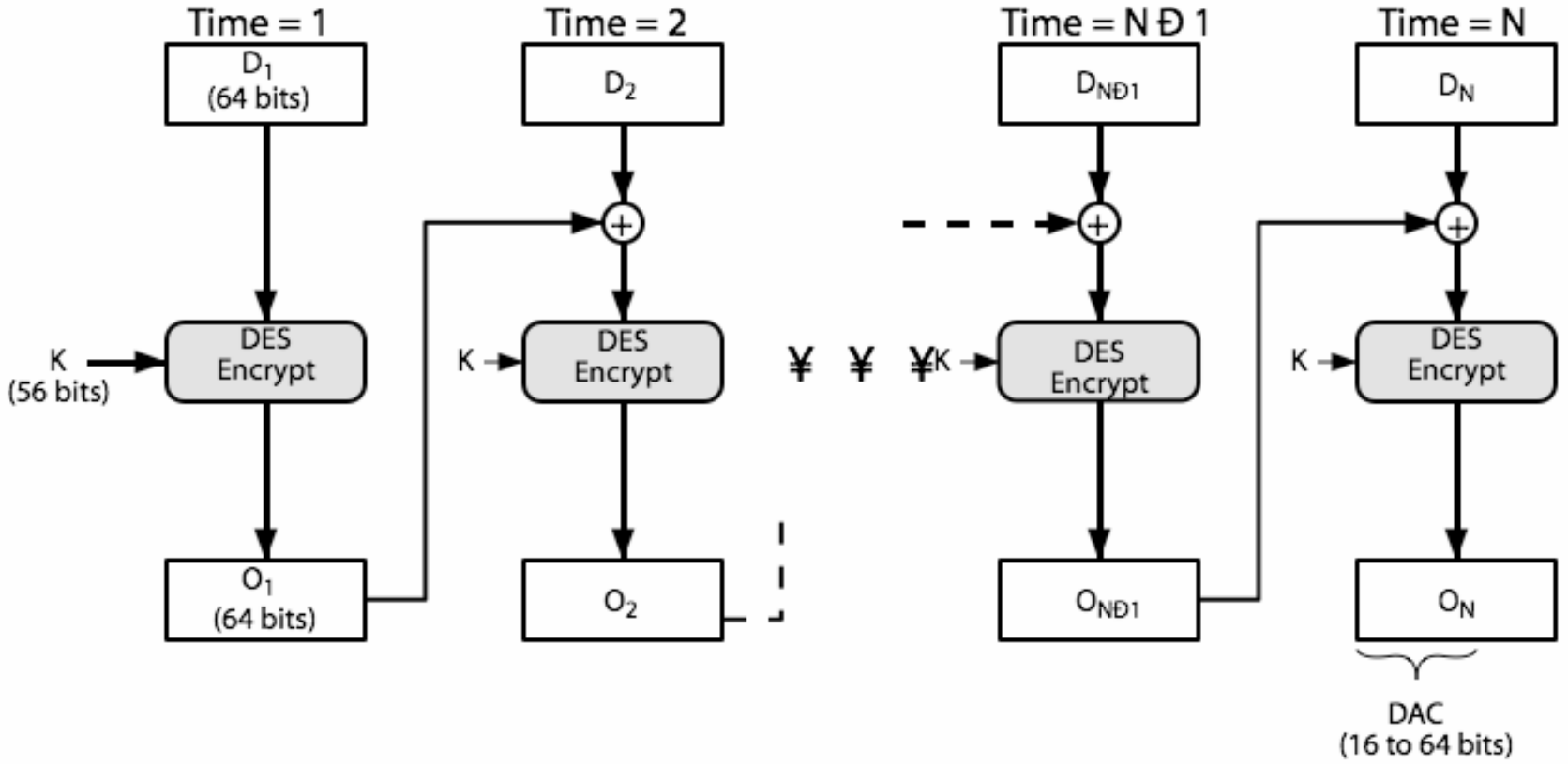
# A MAC alkalmazása

- mint láttuk a MAC-kód csak hitelesítést biztosít
- de titkosítás mellett is használhatjuk
  - általában külön kulcsokat használva
  - a MAC kód a titkosítás előtt és után is számítható
  - de jobb a titkosítás előtt alkalmazni
- A MAC **nem digitális aláírás** (nem letagadhatatlan, mindkét fél elkészítheti)
- Mikor érdemes használni?
  (a titkosítással történő hitelesítéssel szemben)
  - érdemes szétválasztani a bizalmasságot és a hitelesítést
  - néha csak hitelesítés szükséges, titkosítás nem
    pl. alkalomszerű ellenőrzés (gyorsításért), programkód hitelesítés, hálózat menedzselés (SNMPv3)
  - a MAC hitelességet hosszabb időre tudja biztosítani, mint a titkosítást, a megfejtés után a nyílt szöveg tárolható a MAC kóddal együtt

# Szimmetrikus titkosítók használata MAC készítésére

- bármely szimmetrikus titkosítóval készíthetünk MAC-et, ha a  CBC (cipher block chaining) módban az utolsó blokkot használjuk MAC-ként
- a **Data Authentication Algorithm (DAA)** széles körben volt használatban, ez egy MAC algoritmus, mely a DES-t használja CBC módban
  - a kezdővektor:  IV=00….0
  - ha szükséges az üzenetet 0-kal kell kiegészíteni, hogy egész számú 64 bites blokkból álljon
  - titkosítsuk az üzenetet DES-sel CBC módban
  - az utolsó 64 bites eredmény a MAC érték
    - vagy csak annak a legelső M bites blokkját (16≤M≤64)
- de a végső MAC mérete **túl rövid** a biztonsághoz
  Majd látjuk: a brute force ellen hosszabb MAC kell mint szimmetrikus titkos kulcs!

# Az adat hitelesítő algoritmus (Data Authentication Algorithm DAA)

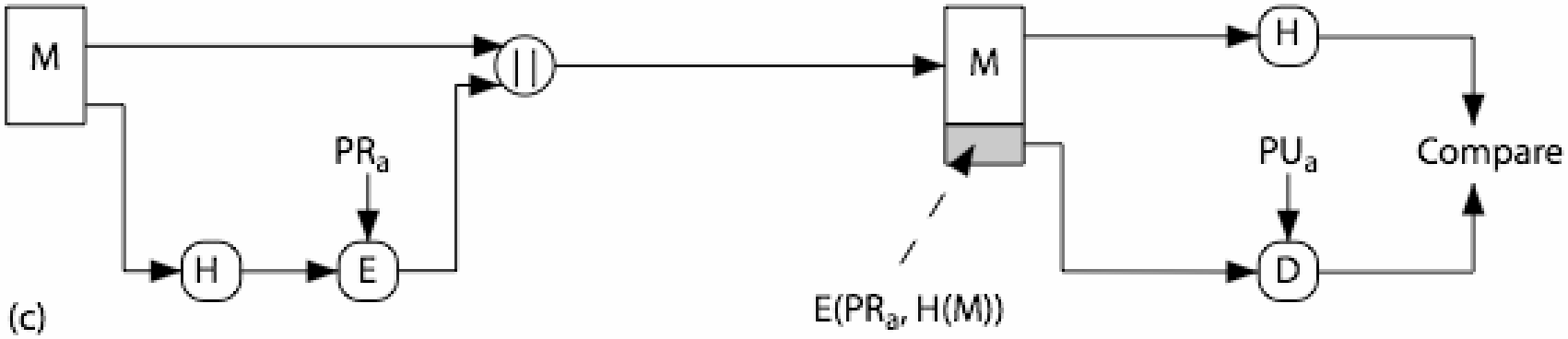



Nem más mint a DES CBC módban.

# 3. Hash-függvények

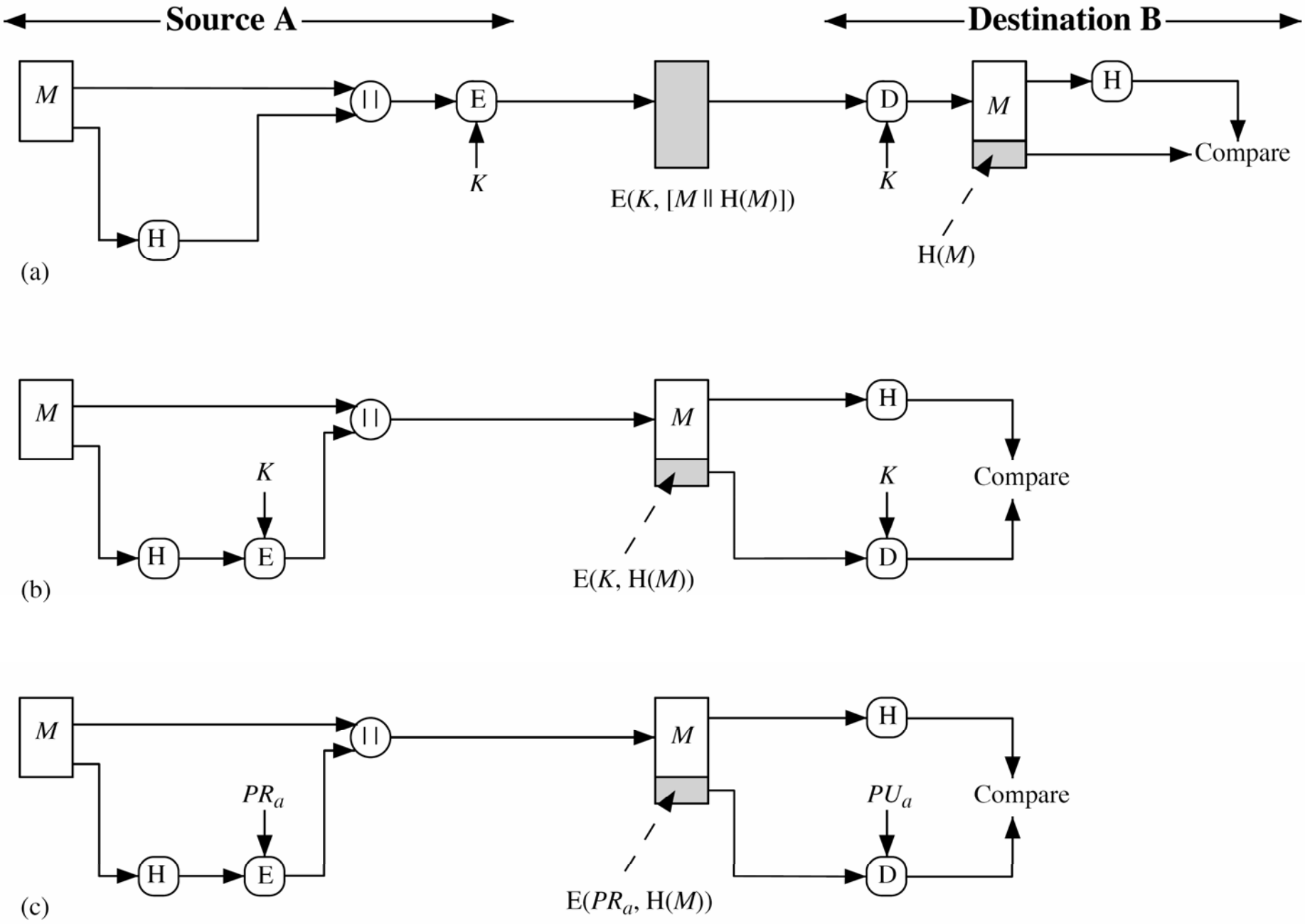
- tetszőleges üzenetet rögzített hosszra sűrítenek
  - **`h = H(M)`**
- nincs szükség kulcsra
  - szemben a MAC-kel
- ,,egyirányúak'' = a jó hash függvényt reménytelenül nehéz invertálni
- nyilvános algoritmusok
- jelzik az üzenet megváltozását
- többféleképpen használhatóak
- leggyakrabban elektronikus aláírás (digital signature) készítéséhez

# Hash függvény egy használata csak hitelesítésre

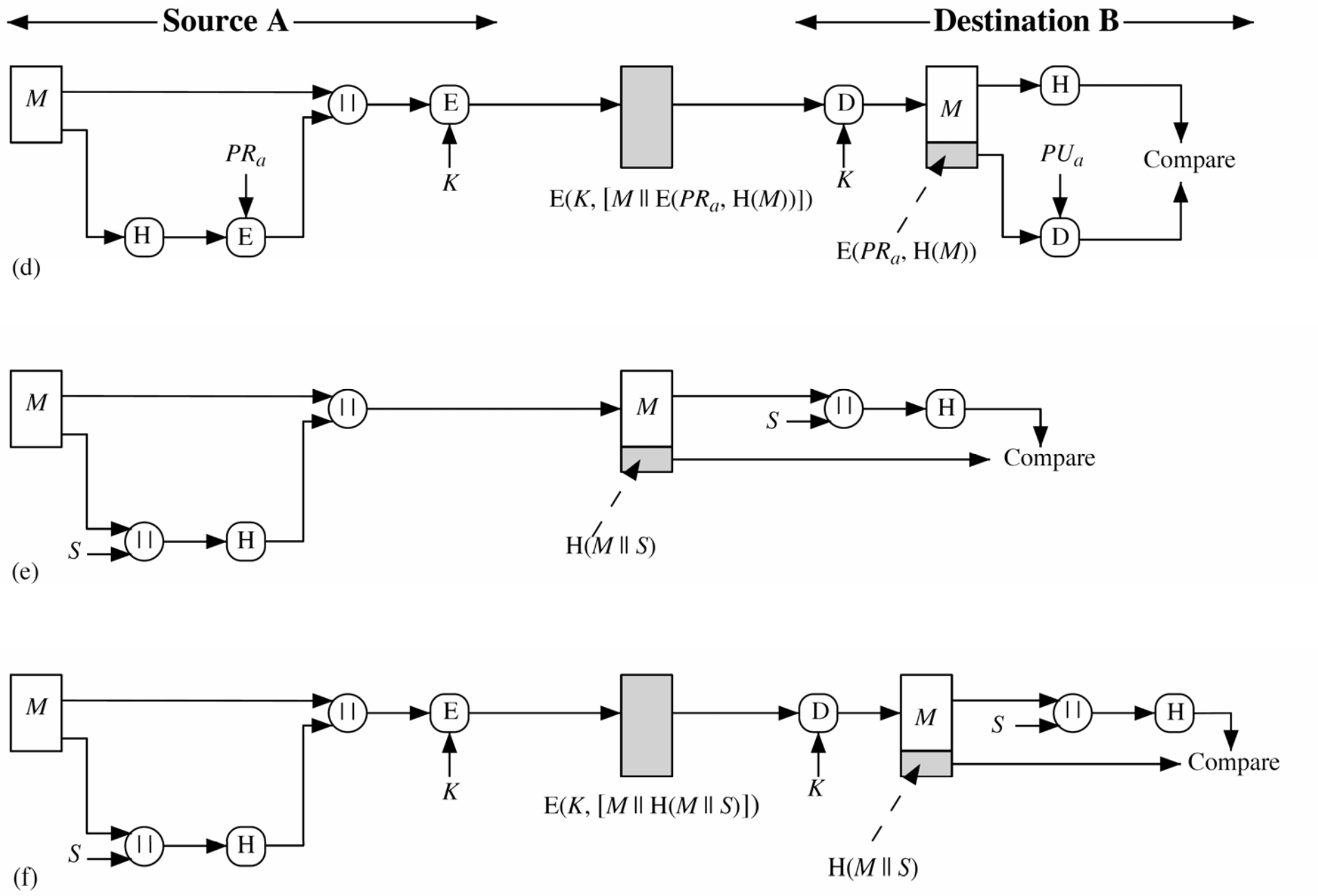



Az üzenet nincs titkosítva csak aláírva.

# Hash függvények alkalmazásai I



# Hash függvények alkalmazásai II



# Követelmények hash függvényekkel szemben

1. tetszőleg méretű **M** üzenetre alkalmazhatóság
2. rögzített méretű **h** kimenetet adjon
3. **h=H(M)**-et könnyű legyen kiszámítani
4. **h** ismeretében kivitelezhetetlen olyan **x**-et találni, melyre **H(x)=h**

**egyirányú tulajdonság** (one-way property)

5. ismert **x**-re kivitelezhetetlen olyan **y**-t találni, melyre **H(y)=H(x)**

**gyenge ütközésmentesség** (weak collision resistance)

6. kivitelezhetetlen két olyan **x,y**-t találni melyre **H(y)=H(x)**

**erős ütközésmentesség** (strong collision resistance)

# Egyszerű hash függvények

- több javaslat született egyszerű függvényekre
- az üzenet blokkjainak XOR-olásával
- nem biztonságosak, mert könnyű bármely üzenetet úgy módosítani, hogy a hash érték változatlan maradjon, vagy kívánt érték legyen
- erősebb kriptográfiai függvényekre van szükség

# Születésnap elvű támadások (Birthday Attacks)

- azt gondolnánk, hogy egy **64**-bites hash biztonságos
- de a **Születésnap paradoxon** (**Birthday Paradox)** miatt ez nincs így
- ahhoz hogy találjunk két azonos napon született embert legalább 50%-os valséggel csak **23** ember kell!
- ez általánosítható úgy is, ha két csoportból (mondjuk fiúk-lányok) akarunk egy olyan párt találni, akik véges sok érték közül közös tulajdonsággal rendelkeznek
- így pl. 64 bites hash esetén $2^{64}$ hash érték van, de csak két $2^{32}$ nagyságú csoportot kell képeznünk, ahhoz, hogy legalább 50%-os valséggel egyezést találjunk.
- **Műveletgény: $2^{33}$ hash érték számítás**

# Születésnap elvű támadások (Birthday Attacks)

- **A születésnap elvű támadás:**
  - az ellenfél generál $2^{32}$ db variációját egy valós üzenetnek lényegében ugyanazzal a jelentéssel (pl. szinonimákkal vagy szóközökkel)
  - az ellenfél szintén generálja $2^{32}$ variációját a hamis üzenetnek, amire ki akarja cserélni
  - a születésnap paradoxon miatt legalább 50% annak a valószínűsége, hogy a két üzenetcsoportban talál két azonos hash értékkel rendelkezőt
  - ezután aláírathatja a valós üzenetet a küldővel, amit aztán kicserélhet az azonos hash értékű hamis párjával
- Nagyobb MAC/Hash értékek szükségesek !!!

# Egy levél $2^{37}$ változat ban

Dear Anthony,

{This letter is / I am writing} to introduce {you to / to you} {Mr. / --} Alfred {P. / --}

Barton, the {new / newly appointed} {chief / senior} jewellery buyer for {our / the}

Northern {European / Europe} {area / division} . He {will take / has taken} over {the / --}

responsibility for {all / the whole of} our interests in {watches and jewellery / jewellery and watches}

in the {area / region} . Please {afford / give} him {every / all the} help he {may need / needs}

to {seek out / find} the most {modern / up to date} lines for the {top / high} end of the

market. He is {empowered / authorized} to receive on our behalf {samples / specimens} of the

{latest / newest} {watch and jewellery / jewellery and watch} products, {up / subject} to a {limit / maximum}

of ten thousand dollars. He will {carry / hold} a signed copy of this {letter / document}

as proof of identity. An order with his signature, which is {appended / attached}

{authorizes / allows} you to charge the cost to this company at the {above / head office}

address. We {fully / --} expect that our {level / volume} of orders will increase in

the {following / next} year and {trust / hope} that the new appointment will {be / prove}

{advantageous / an advantage} to both our companies.

# Blokktitkosítók, mint hash függvények

- a blokktitkosítók hash függvényként is használhatók:
  - **$IV=H_0:=00\ldots0$**
  - az üzenet végének föltöltésse **0**-kal (padding)
  - számítás: $\mathbf{H_i = E_{M_i}\,[H_{i-1}]}$ , ahol $\mathbf{M_i}$ az **i**-dik blokk
  - az utolsó blokk értéke használható hashként
  - hasonlít a CBC módra, de kulcs nélküli
- az eredményként adott hash túl rövid (64, 128-bit)
  - mind a születésnap elvű támadás
  - mind a középen találkozó “meet-in-the-middle” támadás veszélyezteti
- Más variáció is sérülékenyek
- Jó hash függvényekhez más megközelítés kell.

# Hash függvények és MAC algoritmusok biztonsága

- támadástípusok mint a blokktitkosítóknál
- **teljes kipróbálás (brute-force)**
  - hash: erős ütközésmentesség ellen csak: $2^{m/2}$
    - javasoltak  hardvert MD5 törésére
    - 128-bit sérülékeny, 160-bit már jobb
  - MAC: nehezebb támadni, mert ismert üzenet - MAC érték párok kellenek
    - támadható a kulcstér vagy direkt a MAC érték
    - legalább 128-bites MAC kell a biztonsághoz

# Hash függvények és MAC algoritmusok biztonsága

- **kriptográfiai támadások**
  - a függvény egyirányúságát támadják
  - cél, hogy ne lehessen a brute force-nál jobbat adni
  - pl. 2004: **MD5 nem erősen ütközésmentes!!!**
  - 2006: MD5-höz egy percen belül egy laptopon két azonos hash értékű üzenet található

- számos analitikus támadás ismert iterált hash függvényekre is (a körfüggvényt kell visszafejteni elég sok körön keresztül).

# Felhasznált irodalom


- Virrasztó Tamás: Titkosítás és adatrejtés: Biztonságos kommunikáció és algoritmikus adatvédelem, NetAcademia Kft., Budapest, 2004. Online elérhető: http://www.netacademia.net/book.aspx?id=1#

- William Stallings: Cryptography and Network Security, 4th Edition, Prentice Hall, 2006. (Chapter 11)
- Lawrie Brown előadás fóliái (Chapter 11)