Kriptográfia

Hetedik előadás
Nyilvános kulcsú kriptográfia II.
Kulcsgondozás és további nyilvános kulcsú rendszerek

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

Kulcsgondozás (Key Management)

- a nyilvános kulcsú titkosítás segít megoldani a kulcselosztás problémáját

Ennek két aspektusa van:

- I. a nyilvános kulcsok szétosztása
  - azaz eljuttatása mindazokhoz, akik üzenetet küldhetnek a titkos kulcs birtokosának
- II. a nyilvános kulcsú titkosítás használata titkos kulcsok cseréjére
  - hogy az így biztonságosan eljuttatott titkos kulccsal gyors szimmetrikus titkosítással lehessen kommunikálni
    $\Rightarrow$ **hibrid kriptorendszerek**, ld. később.

A nyilvános kulcsok szétosztása (Distribution of Public Keys)

- **A nyilvános kulcsú titkosítás, semmit sem ér, ha a küldő nem győződik meg arról, hogy a nyilvános kulcs hiteles**, azaz valóban a címzetté.
- Mert egy támadó a címzett nyilvános kulcsát a saját nyilvános kulcsára hamisítva elolvashatja az üzenetet.
- Sőt, ha elfogja és megfejti az üzenetet, utána még a címzett valódi kulcsával titkosítani is tudja, majd továbbítani a címzettnek. Így még észrevétlen is maradhat.
- De pont azért szeretnénk a nyilvános kulcsú rendszert használni, hogy ne kelljen a nyilvános kulcsot előre titokban kommunikálni.
- Hogyan oldható ez meg mégis? Kemény dió.

A nyilvános kulcsok szétosztása (Distribution of Public Keys)

Különböző technikák vannak használatban, melyek az alábbi 4 csoportba sorolhatók:

- nyilvános kihirdetés (public announcement)
- nyilvános hozzáférésű katalógus (publicly available directory)
- nyilvános kulcsszolgáltató (public-key authority)
- nyilvános kulcs tanúsítványok (public-key certificates)

1. Nyilvános kihirdetés

- a felhasználók közhírré teszik a nyilvános kulcsukat a várható feladóknak, vagy nagyobb közösségeknek pl.
  - e-mailhez csatolva
  - hírcsoportokra, levelezési listákra küldve
  - (személyes) weblapra feltéve, stb.
- a legnagyobb hátránya, hogy **könnyen hamisítható**:
  - bárki könnyen készíthet nyilvános-magán kulcspárt, amit más nevében adhat ki
  - amíg a csalást a címzett észre nem veszi, és a feladókat nem figyelmezteti, a támadó szabadon olvashatja a neki küldött üzeneteket

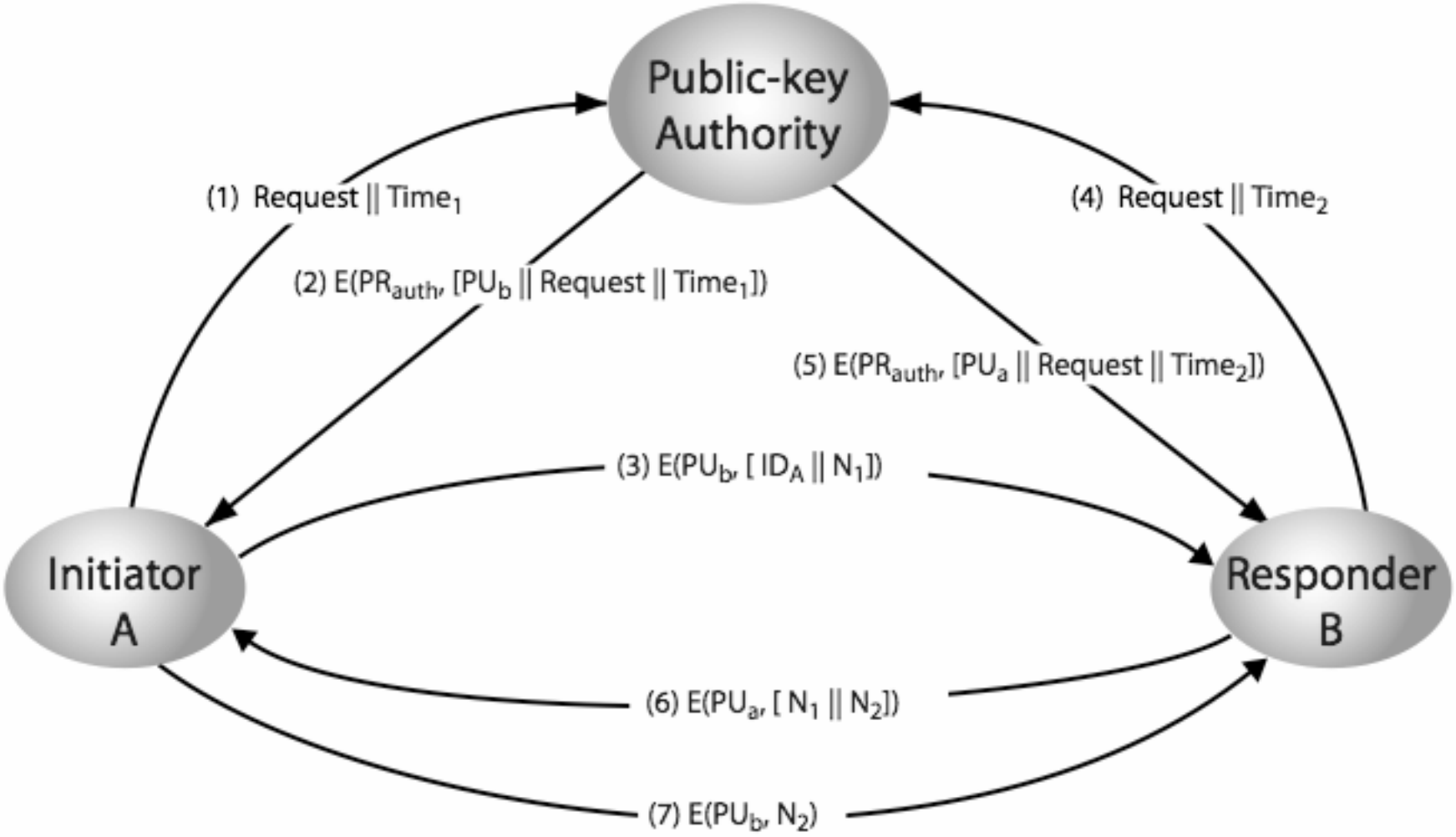
2. Nyilvános hozzáférésű katalógus (Publicly Available Directory)

- nagyobb biztonság érhető el, a kulcsok nyilvános katalógusba történő regisztrációjával
- a katalógust működtető egyénben/szervezetben a résztvevőknek meg kell bízni
- a katalógus:
  - **{név, nyilvános kulcs}** bejegyzéseket tartalmaz
  - a résztvevők személyesen v. titkosan regisztrálják kulcsaikat a katalógusba
  - a katalógusban tárolt kulcsot bármikor ki lehet cserélni
  - a katalógust időnként közzéteszik
  - a katalógus elektronikusan is biztonságos kommunikációval elérhető
- hátrány: lehetséges hamisítás és a katalógus feltörése

3. Nyilvános kulcsszolgáltató (Public-Key Authority)

- a biztonságosság növelhető a kulcsok katalógusból történő kiadásának szigorúbb ellenőrzésével
- tulajdonságai ua.-ok mint a katalógus esetében
- a felhasználóknak ismerni kell a katalógus nyilvános kulcsát
- a kért nyilvános kulcsok a katalógusból biztonságos interaktív kapcsolattal kérhetők le
  - de csak akkor szükséges valós idejű hozzáférés a katalógushoz, ha a kulcsok még nem állnak rendelkezésünkre,
  - azaz (1)-(2)-t és (4)-(5)-öt csak ekkor kell végrehajtani

3. Nyilvános kulcs-csere kulcsszolgáltatón keresztül

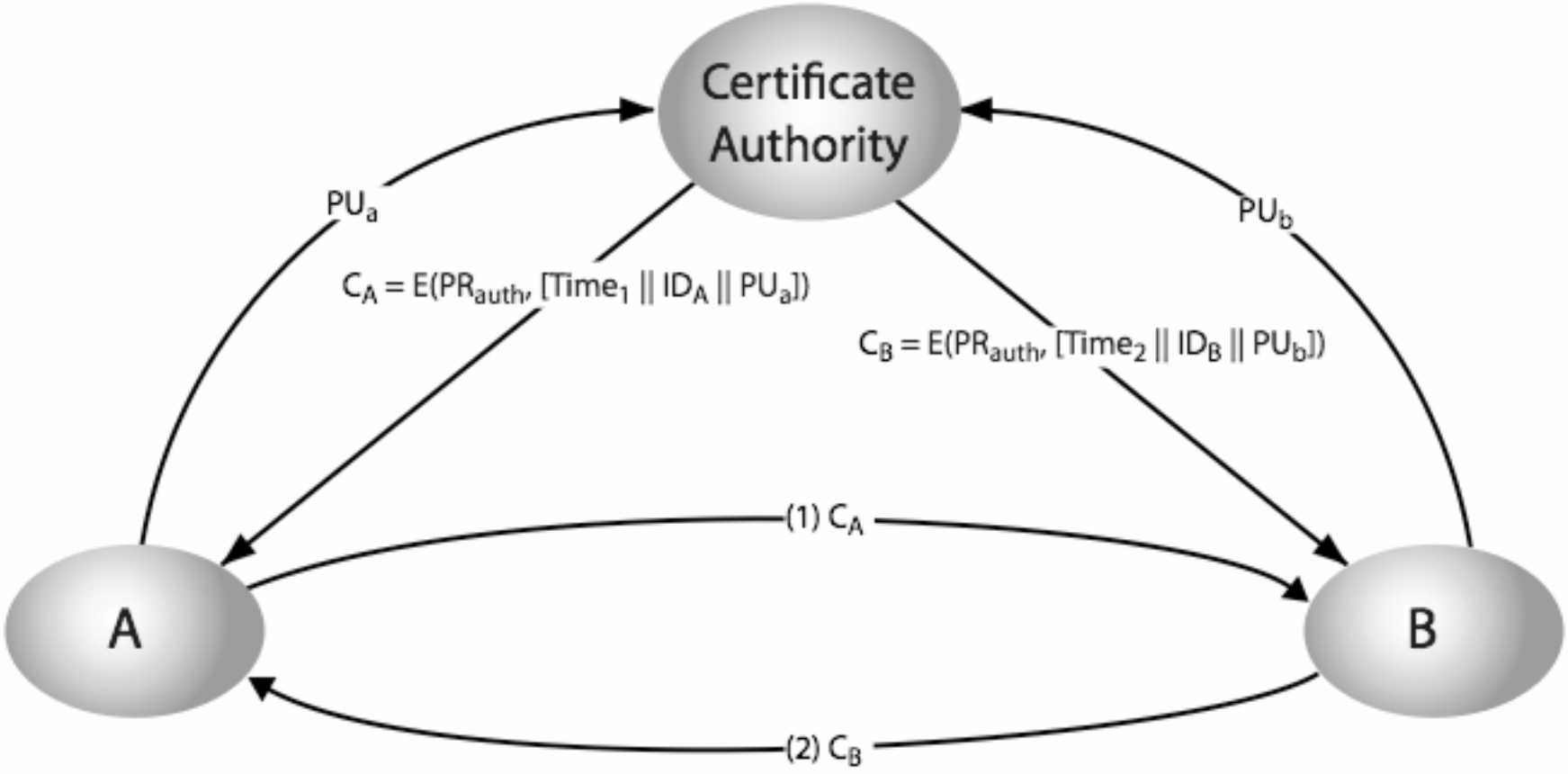


N_1 és **N_2** „nonce” = véletlen bitsorozat, a kapcsolat egyedi azonosítója

4. Nyilvános kulcs tanúsítványok (Public-Key Certificates)

- a kulcsszolgáltató használata szűk keresztmetszet lehet: elérhetőség, feltörhetőség
- a **tanúsítvány** biztosítja a kulcs-cserét a kulcsszolgáltató valós idejű elérése nélkül
- a tanúsítvány a **felhasználói azonosító** és a **nyilvános kulcs** összetartozását igazolja
  - általában segédinformációkat is tartalmaz, mint érvényességi idő, használati jogkör stb.
- a tanúsítványt **aláírja** a nyilvános kulcs szolgáltató vagy hitelesítés-szolgáltató Certificate Authority (**CA**)
- bárki ellenőrizheti, aki ismeri a szolgáltató nyilvános kulcsát

4. Nyilvános kulcs tanúsítványok használata



II. A nyilvános kulcsú titkosítás használata titkos kulcsok cseréjére

- az előző módszerekkel megszerzett nyilvános kulcsot felhasználhatjuk titkosításra vagy hitelesítésre
- de a nyilvános kulcsú algoritmusok lassúak
- ezért inkább a gyors szimmetrikus módszerekkel akarjuk magát az üzenetet titkosítani
- ehhez egy **kapcsolatkulcsra** (session key) **van szükség**
- ezt cserélik ki a felek egymás közt a nyilvános kulcsú titkosítás használatával

II. a nyilvános kulcsú titkosítás használata titkos kulcsok cseréjére

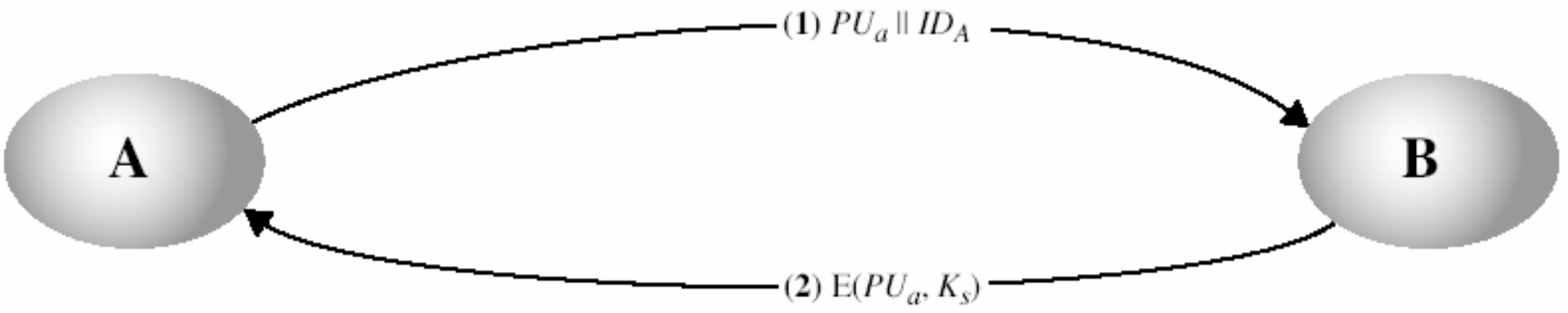
Több alternatívája van a kapcsolatkulcsban való megegyezésnek és így a biztonságos kapcsolat megteremtésének. *Pl:*

1) egyszerű titkos kulcs szétosztás
2) titkos kulcs szétosztás nyilvános kulcsú kriptográfiával
3) hibrid kulcs szétosztás

1. Egyszerű titkos kulcs szétosztás (Simple Secret Key Distribution)

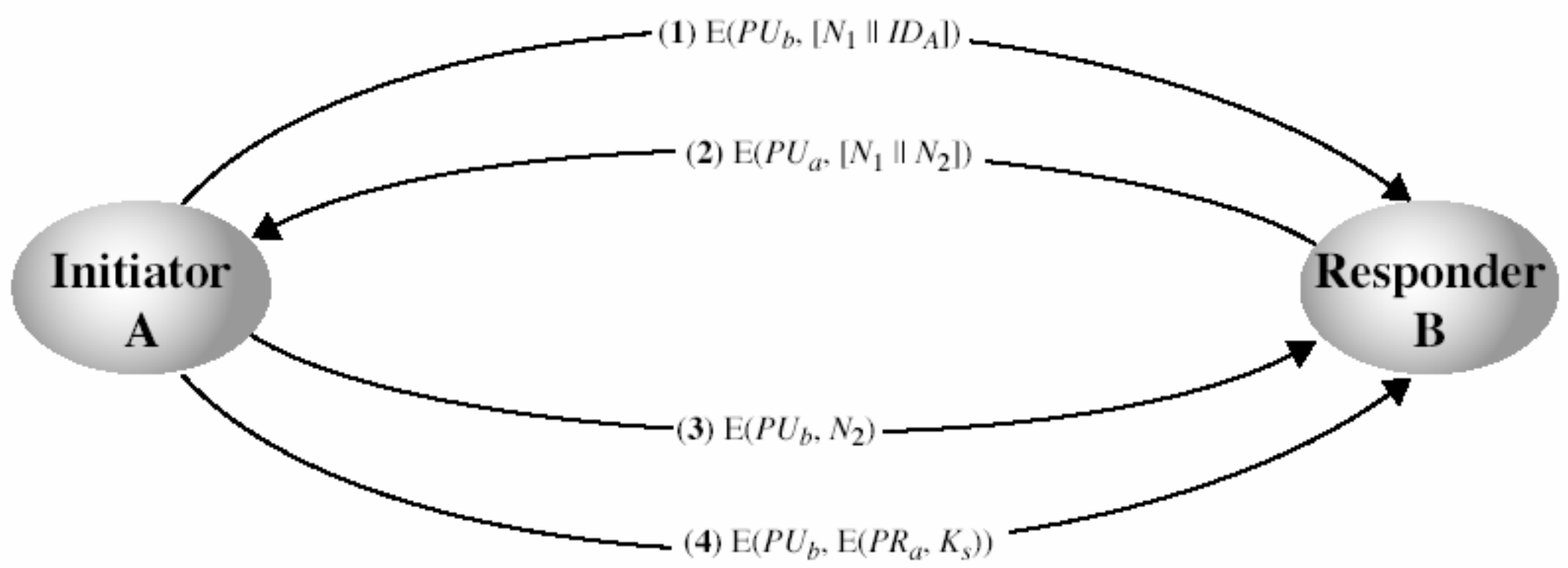
- Merkle javasolta 1979-ben
  1. A generál egy új ideiglenes nyilv./magán kulcspárt
  2. A elküldi B-nek a nyilvános kulcsot és az azonosítóját
  3. B generál egy új **K** kapcsolatkulcsot és elküldi A-nak a kapott nyilvános kulccsal titkosítva
  4. A megfejti **K**-t a magánkulccsal
- Biztonságos passzív támadás (csak lehallgatás) ellen.
- Nem biztonságos aktív támadás ellen, a támadó a nyilvános kulcsot a sajátjára cserélve megszemélyesítheti A-t és hozzájuthat **K**-hoz.

1. Egyszerű titkos kulcs szétosztás vázlat



2. Titkos kulcs szétosztás nyilvános kulcsú kriptográfiával

Ha már $\mathbf{PU_A}$ és $\mathbf{PU_B}$ cseréje megtörtént:



$\mathbf{N_1}$ és $\mathbf{N_2}$ „nonce” = véletlen bitsorozat, a kapcsolat egyedi azonosítója

3. Hibrid kulcs szétosztás (IBM)

- szükséges egy kulcselosztó központ
  (key distribution center: **KDC**)
- itt minden felhasználónak egy titkos (szimmetrikus) főkulcsa (master key) van
- a KDC szimmetrikus titkosítással továbbítja a kapcsolatkulcsokat a felekhez
- nyilvános kulcsú titkosítás csak a főkulcsok szétosztásához, váltásához szükséges
- előnyei:
  - teljesítmény (a szimmetrikus titkosítás gyorsabb)
  - hátrafele kompatibilitás (backward compatibility)
- különösen egyetlen KDC és széles felhasználói kör esetén hasznos

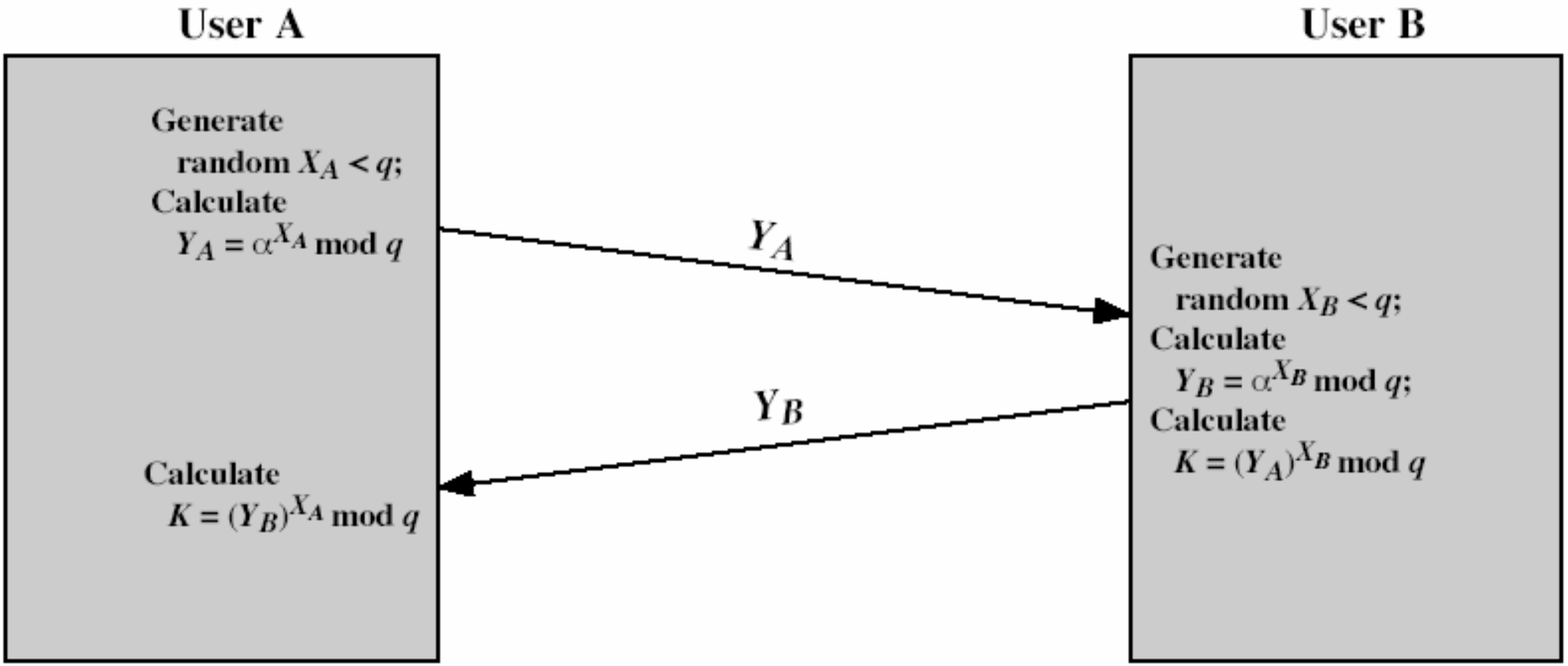
Diffie-Hellman kulcs-csere (Diffie-Hellman Key Exchange)

- az első publikált nyilvános kulcsú módszer
- Diffie és Hellman (1976), a nyilvános kulcsú rendszer ismertetésével
  - *megj: ma már tudjuk, hogy Williamson (UK CESG) 1970-ben ismerte ezt, de eredményét titokként kezelték*
- egy praktikus eljárás titkos kulcsok nyilvános cseréjére
- számos kereskedelmi termékben használják

Diffie-Hellman kulcs-csere

- egy közös titkos érték kialakítását valósítja meg
  - nem használható tetszőleges üzenet titkosítására
  - de jó egy közös kulcsban való megállapodásra
  - melyet csak a két fél ismer
- a végső közös titok függ mindkét résztvevőtől (a felek nyilvános és titkos információiktól)
- ,,könnyű irány'': moduláris (v. véges test feletti) hatványozás
- ,,nehéz irány'': diszkrét logaritmus számítása

Diffie-Hellman kulcs-csere vázlat



Diffie-Hellman kulcs generálás

- a felhasználók (többen is lehetnek, nem csak ketten) megegyeznek a globális paraméterekben:
  - egy **q** nagy prímszám
  - **a** egész mely **q primitív gyöke**
    azaz **a** hatványiként minden 0<y<q egész előáll **mod q**
- minden felhasználó (pl. A) elkészíti a kulcsait:
  - generál egy titkos $x_A < q$ számot, a magánkulcsát
  - kiszámítja a nyilvános kulcsát: $y_A = a^{x_A} \bmod q$
- a nyilvános kulcsát mindenki közzéteszi

Diffie-Hellman kulcs-csere

- az A és B közös titka K_{AB} lesz:

  $K_{AB} = a^{x_A \cdot x_B} \bmod q$

  $= y_A^{x_B} \bmod q$ (ahogy B számítja ki)

  $= y_B^{x_A} \bmod q$ (ahogy A számítja ki)

- K_{AB} ezután használható titkos kulcsként **szimmetrikus** titkosításra **A** és **B** között
- a támadónak a titkos x_A vagy x_B értékek valamelyikének kiszámítására van szüksége, ami a diszkrét logaritmus probléma megoldását jelenti
- hátrány, hogy A és B a következő kapcsolatban is ugyanazzal a kulccsal kommunikál, ha csak nem generálnak újra nyilvános/magán kulcspárt

Példa Diffie-Hellman kulcs-cserére

- A és B közötti titok kialakítása:
- közös paraméterek: **q=353** és **a=3**
- választott magánkulcsok:
  - A választása x_A**=97**, B választása x_B**=233**
- a nyilvános kulcsok kiszámítása:
  - $y_A = 3^{97} \bmod 353 = 40$ (A-é)
  - $y_B = 3^{233} \bmod 353 = 248$ (B-é)
- a közös titok kiszámítása:
  - $K_{AB} = y_B^{x_A} \bmod 353 = 248^{97} = 160$ (A)
  - $K_{AB} = y_A^{x_B} \bmod 353 = 40^{233} = 160$ (B)

Kulcs-csere protokollok (Key Exchange Protocols)

- minden pár minden kommunikáció előtt új nyilvános/magán kulcsot generálva egyezik meg a kapcsolatkulcsban
- a felhasználók csak egyszer készítik el a kulcsaikat, melyeket egy közös katalógusban tesznek közzé, mellyel biztonságosan kommunikálnak
- **sajnos mindkét módszer sérülékeny az aktív (men-in-the-middle) támadásra** (HF!)
- **a kulcsok hitelesítésére van szükség**

Elliptikus görbéken alapuló kriptográfia
Elliptic Curve Cryptography

- a nyilvános kulcsú kriptográfia jórésze (RSA, D-H) hatalmas prímszámokra vagy polinomokra épül
- ez jelentős terhet jelent a kulcsok használatában, cseréjében és tárolásában
- az elliptikus görbék erre kínálnak alternatívát
- **ugyanazt a biztonságot kisebb kulcsméret mellett nyújtják**
- új módszerek, még nem annyira kiforrott az elemzésük (mint pl. az RSA-é)

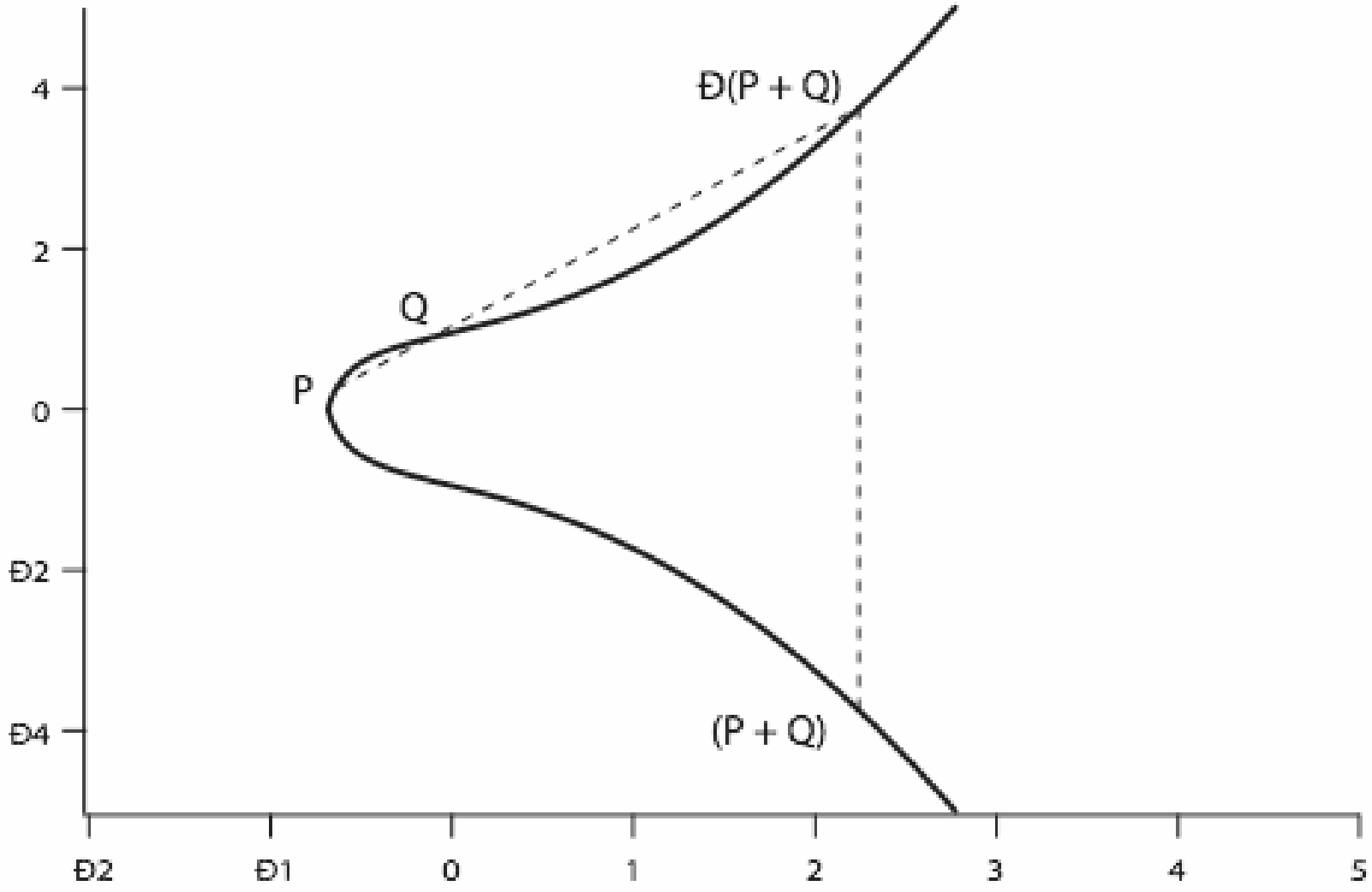
Valós elliptikus görbék

- egy kétváltozós egyenlet definiálja őket
- a számunkra érdekes harmadrendű elliptikus görbék az alábbi alakúak:
  - $y^2 = x^3 + ax + b$
  - ahol x,y,a,b valós számok
  - feltétel: nincs többszörös gyök, azaz $\mathbf{4a^3 + 27b^2 \neq 0}$
  - és még egy speciális pont: **O** (végtelen távoli pont)
- egy ilyen göbe pontjain összeadást definiálhatunk:
- a P és Q pont összege: **P+Q** legyen a PQ egyenes és a görbe R metszéspontjának, az

  x tengelyre vonatkozó tükörképe

  (Ez szintén a görbén lesz, mert a görbe az x –tengelyre szimetrikus)

Összeadás elliptikus görbén



(b) $y^2 = x^3 + x + 1$

Összeadás elliptikus görbén

P +Q számításának speciális esetei:

- P + O = O +P = P
- ha nincs metszéspont $\Rightarrow$ PQ egyenes függőleges, ekkor P+Q := O,
  (vagyis Q = -P, az additív inverz = a tükörkép)
- ha P = Q, akkor 2P = P + P a P-beli érintő és a görbe metszetének tükörképe

- Belátható, hogy a görbe pontjai az így definiált összeadásra nézve Abel-csoportot alkotnak.
- Így pl.: (P + Q) + (-Q) = P /ezért kell tükrözni./
- Szemléltetése: CryptTool bemutató
  (Indiv. Procedures / legalsó menüpont)

Véges elliptikus görbék

- Az elliptikus görbék kriptográfiája olyan görbéket használ, melynek mind az együtthatói, mind a változóinak értékei végesek.
- két általánosan használt családjuk:
  - prím görbék $\mathbf{E_p(a,b)}$ a Z_p test felett
    - egész értékeket használ modulo p (ahol p prím)
    - feltétel, hogy $\mathbf{4a^3 + 27b^2 \neq 0}$
    - $\mathbf{E_p(a,b)= \{(x,y) \mid} \boldsymbol{y^2=x^3+ax+b \bmod p\}}$
    - szoftver alkalmazásokban jobb
  - bináris görbék $\mathbf{E_{2^m}(a,b)}$ a $GF(2^m)$ véges test felett
    - bináris együtthatójú polinomokat használ
    - hardver alkalmazásokban jobb

Elliptikus görbék kriptográfiája (Elliptic Curve Cryptography)

- **P+Q**: ECC összeadás = a hagyományos moduláris szorzás analógiája
- **kP**: ECC ismételt összeadás = a hagyományos moduláris hatványozás analógiája
- a “nehéz irány” a diszkrét log analógiája:
  - **Q=kP**, ahol Q és P a prím görbe pontjai
  - Q-t kiszámítani k és P ismeretében könnyű
  - de k-t meghatározni Q-ból és P-ből nehéz
  - ezt nevezzük elliptikus görbe logaritmus problémának (elliptic curve logarithm problem)
- Példa: $E_{23}(9,17)$

ECC Diffie-Hellman kulcs-csere

- a D-H-hoz hasonlóan állapodhatunk meg egy közös titokban
- a felek előre választanak egy alkalmas $E_p(a,b)$ görbét
- és egy alappontot (base point) $G=(x_1,y_1)$
  - melynek **rendje**: **n** nagy
    (rend = a legkisebb n melyre $nG=0$)
- A és B magánkulcsot választ: $n_A<n$, $n_B<n$
- a nyilv. kulcsok: $P_A=n_AG$, $P_B=n_BG$
- a közös titok: $K=n_AP_B$, $K=n_BP_A$
  - ez ugyanaz, mert $K=n_An_BG$

ECC Titkosítás/megfejtés

- több alternatíva van, az alábbi a legegyszerűbb:
- alkalmas görbét és **G** alappontot választunk
- az üzenet M blokkját a görbe egy $\mathbf{P_m}$ pontjával azonosítjuk
- a címzett generál egy magánkulcsot: $\mathbf{n_B < n}$
- és kiszámítja a nyilvános kulcsát: $\mathbf{P_B = n_B G}$
- $\mathbf{P_m}$ titkosítása: $\mathbf{C_m = \{kG,\ P_m + kP_B\}}$, ahol **k** véletlen érték
- $\mathbf{C_m}$ megfejtése:

$$\mathbf{P_m + kP_B - n_B(kG) = P_m + k(n_B G) - n_B(kG) = P_m}$$

Az ECC biztonsága

- a biztonsága az elliptikus görbe logaritmus probléma nehézségén alapszik
- a ma ismert leggyorsabb eljárás rá a
  “Pollard rho algoritmus”
- összehasonlítva a faktorizációval jóval kisebb kulcsmért elégséges, mint pl. az RSA esetén
- ugyanakkor egyező kulcsméret mellett a titkosítás számításigénye, kb. megegyezik az RSA-éval
- ezért hasonló biztonság szint megtartásával az ECC jelentős számítási előnnyel rendelkezik

A szükséges kulcsméretek összehasonlítása egyező biztonsági szintek mellett

Szimmetrikus titkosítások (kulcs bitekben)	**ECC-alapú (*n* mérete bitekben)**	**RSA/DSA (a modulus mérete bitekben)**
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

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 10)
- Lawrie Brown előadás fóliái (Chapter 10)