

Név:

Neptun kód:

1.	2.	3.	4.	5.	6.	Σ

ADATBIZTONSÁG PÓTPÓTZÁRTHELYI

2010. május 21.

1. Nevezzük rossz kulcsnak az olyan kulcsot, amelyre a kódoló transzformáció egyben dekódoló transzformáció is!
 - a.) Keresse meg a $E_k(x)=x+k$; $x, k \in \mathbb{Z}_{26}$ shift rejtjelező rossz kulcsait! (10 pont)
 - b.) Mutassa meg, hogy $E_{a,b}(x)=ax+b$, $a, b, x \in \mathbb{Z}_n$ affin rejtjelező esetén $k=(a,b)$ rossz kulcs akkor és csak akkor, ha $a^{-1}=a \pmod n$ és $b(a+1)=0 \pmod n$! (10 pont)
2. Diffie-Hellman kulcsmegegyezés protokoll
 - a.) Definiálja a protokollt! (7 pont)
 - b.) Nyújtja-e a protokoll az alábbi szolgáltatásokat (egy-egy mondatban definiálja is az alábbi fogalmakat)? (6 pont)
 - kulcskonfirmáció (igen/nem):
 - explicit kulcshitelesítés (igen/nem):
 - kulcsfrissesség (igen/nem):
 - partnerhitelesítés (igen/nem):
 - c.) Támadható-e a protokoll? Ha igen, hogyan, ha nem, miért nem? (6 pont)

Igen Nem Magyarázat:
3. TLS protokoll
 - a.) Milyen alprotokolljai vannak az TLS-nek, és hogyan helyezkednek ezek el a TCP/IP protokoll stack-ben? (3 pont) Milyen feladatokat látnak el az egyes alprotokollok? (7 pont)
 - b.) Egy webszerver és egy böngésző a TLS protokollt használja a HTTP forgalom védelmére. Már létrehoztak egy session-t, és a handshake során RSA alapú kulcscserét használtak. A szerver digitális aláírás ellenőrző kulcsot tartalmazó tanúsítvánnyal rendelkezik, és nem kérte, hogy a kliens hitelesítse magát. Most a kliens egy új kapcsolatot szeretne nyitni a már létező session-ben, és a szerver ebbe beleegyezik. Adja meg, hogy ebben az esetben mely handshake üzenetek kerülnek átvitelre, és vázlatosan adja meg azok tartalmát! (10 pont)

4. Tekintsünk egy szervert és két klienset: A -t és B -t. A kliensek egymást segítve próbálják anonimizálni a szervernek küldött kéréseiket a következő módon:

- A a kéréseit p_{AB} valószínűséggel B -n keresztül, $1-p_{AB}$ valószínűséggel közvetlenül küldi a szervernek,
- B a kéréseit p_{BA} valószínűséggel A -n keresztül, $1-p_{BA}$ valószínűséggel közvetlenül küldi a szervernek.

A kliensek nem azonos valószínűséggel bocsátanak ki kéréseket a szerver felé. Elérhető-e p_{AB} és p_{BA} megfelelő megválasztásával a „gyanún felüli” (beyond suspicion) küldő anonimitási szint a szerverrel szemben? Adjon intuitív választ (5 pont) és formális bizonyítást (15 pont)!

Intuitív válasz:

Formális bizonyítás:

5. Rövid kérdések:

a.) Mire használják a unix sticky bitet? (3 pont)

b.) Mi a spam elleni harcban a DKIM megoldás? (3 pont)

c.) Hogyan szinkronizálja az időt a Conficker bot? (3 pont)

6. Mutassa be a Bell-LaPadula modell három tulajdonságát (ss,ds,*) formálisan! (12 pont)

Pontozás: 1: ≤ 39 , 2: 40 – 54, 3: 55 – 69, 4: 70 – 84, 5: 85 – 100

MEGOLDÁS

1. a) $k=0, k=13: D_k(E_k(x))=(x+k)+k=x \rightarrow 2k=0 \pmod{26}$
- b.) $a(ax+b)+b=x \pmod{n}$, azaz $a^2x+ab+b=x \pmod{n}$ a.cs.a., ha $a^2=1 \pmod{n}$, $ab+b=0 \pmod{n}$
2. Tk. 169. és 189. oldal.
3. a) Alprotokollok:
 - TLS Record: fragmentáció, tömörítés, rejtjelezés, üzenet-hitelesítés és integritásvédelem, visszajátszás elleni védelem
 - TLS Handshake: algoritmusok egyeztetése, kulcscsere, partner-hitelesítés
 - TLS Alert: hibaiüzenetek
 - TLS Change Cipher Spec: Handshake végének jelzése, állapotváltás

Elhelyezkedés a protokoll stack-ben:

- TLS Record: TCP felett
- TLS Handshake, Alert, Change Cipher Spec, és alkalmazások (pl. HTTP): TLS Record felett

b) A következő handshake üzenetek kerülnek átvitelre:

C→S	client-hello	kliens véletlenszáma, létező session azonosítója (korábban megegyezésre került algoritmus-csokor)
S→C	server-hello	szerver véletlenszáma, létező session azonosítója (korábban megegyezésre került algoritmus-csokor)
C→S	client-finished	eddigyi handshake üzeneteken és a mestertitkon számolt MAC (a korábban megegyezésre került algoritmusokkal, de új kulcsokkal védve)
S→C	server-finished	eddigyi handshake üzeneteken és a mestertitkon számolt MAC (a korábban megegyezésre került algoritmusokkal, de új kulcsokkal védve)

4. Intuitív válasz: Nem érhető el a „gyanún felüli” anonimitás, ugyanis a kliensek nem azonos valószínűséggel küldenek kéréseket, s így minden szerverhez érkező kérés esetén (bármilyen utat jár be a kérés) az egyik kliens nagyobb valószínűséggel az eredeti forrás, mint a másik. Például, ha A 9-szer nagyobb valószínűséggel bocsát ki kéréseket, akkor átlagosan a szerverhez érkező kérések 1/10-e származik csak B-től, azaz egy beérkező kérés esetén A nagyobb valószínűséggel az eredeti küldő.

Bizonyítás: Jelöljük az eredeti küldőt α -val, és azt a hosztot akitől a szerver a kérést megkapja ω -val. Jelöljük továbbá p_A -val annak valószínűségét, hogy az eredeti küldő A, és p_B -vel annak valószínűségét, hogy az eredeti küldő B. Ekkor:

$$\begin{aligned} \Pr\{\alpha = A | \omega = A\} &= \frac{\Pr\{\omega = A | \alpha = A\} \Pr\{\alpha = A\}}{\sum_{X \in \{A, B\}} \Pr\{\omega = A | \alpha = X\} \Pr\{\alpha = X\}} \\ &= \frac{(1 - p_{AB})p_A}{(1 - p_{AB})p_A + p_{BAPB}} \end{aligned}$$

Hasonlóan:

$$\Pr\{\alpha = B|\omega = A\} = \frac{p_{BAPB}}{(1 - p_{AB})p_A + p_{BAPB}}$$

$$\Pr\{\alpha = B|\omega = B\} = \frac{(1 - p_{BA})p_B}{(1 - p_{BA})p_B + p_{ABP_A}}$$

$$\Pr\{\alpha = A|\omega = B\} = \frac{p_{ABP_A}}{(1 - p_{BA})p_B + p_{ABP_A}}$$

A gyanún felüli anonimitási szint elérésének feltétele a következő:

$$\Pr\{\alpha = A|\omega = A\} = \Pr\{\alpha = B|\omega = A\}$$

$$\Pr\{\alpha = B|\omega = B\} = \Pr\{\alpha = A|\omega = B\}$$

Felhasználva az előzőeket, ezek a feltételek így alakulnak:

$$(1 - p_{AB})p_A = p_{BAPB}$$

$$(1 - p_{BA})p_B = p_{ABP_A}$$

ahonnan:

$$p_A = p_{ABP_A} + p_{BAPB} = p_B$$

feltétel adódik. Mivel azonban $p_A \neq p_B$, ezért a feltétel nem teljesülhet.

5.

- a.) sticky bit: +t, csak a tulajdonos törölheti a fájlt vagy alkönyvtárat. Hiába van esetleg írási joga az adott felhasználónak, nem törölheti.
- b.) DKIM: Minden e-mail a DNS rekordban tárolt nyilvános kulcshoz tartozó titkos kulccsal aláírásra kerül a kiküldő szerverek által, így kevesebb a lehetőség hamisított címmel levelet feladni.
- c.) Conficker szinkron: Fontosabb weboldalak (Google, Yahoo, ...) lekérdezésével megszerzett HTTP fejlécben található Date mezőt használja fel a szinkronizáláshoz.

ss-property: for all (S_i, O_j, read) in b , $f_c(S_i) \geq f_o(O_j)$.

6.

a subject at a given security level may not read an object at a higher security level (no read-up).

***-property: for all $(S_i, O_j, \text{append})$ in b , $f_c(S_i) \leq f_o(O_j)$ and for all (S_i, O_j, write) in b , $f_c(S_i) = f_o(O_j)$**

a subject at a given security level must not write to any object at a lower security level (no write-down). The *-property is also known as the Confinement property.

ds-property: for all (S_i, O_j, A_x) in b , $A_x \in M[S_i, O_j]$

use of an access matrix <http://en.wikipedia.org/wiki/The_Access_Matrix> to specify the discretionary access control. An individual may grant to another individual access to a file based on the owner's discretion constrained by the MAC rules.