Adatbiztonság pótZH Név:
2012. május 17. Neptun kód:

1. Egy kommunikációs csatorna 10^{-6} bithiba-aránnyal működik. Hogyan alakul a bithiba-arány rejtjelezett esetben, ha a rejtjelezés
a.) 128 bites kódolás CBC módban? (2p)
b.) 64 bites kódolás OFB byte alapú folyamrejtjelezéssel? (2p)

A véletlen csatornahibák ellen hibajavító kódolást alkalmazunk és a hibajavító kódolást a rejtjelezést megelőzően végezzük. Helyesen járunk-e el a hibák javításával kapcsolatosan, ha
c.) CBC módban rejtjelezünk? (2p)
d.) OFB módban rejtjelezünk? (2p)

2. Tekintsük az alábbi integritásvédelmi kódolást, ahol rejtjelezés és MAC kombinációját használjuk: $E_k(m \mid MAC_{k'}(m))$
Jelölések: $E_k(.)$ CBC módú rejtjelezés, az m üzenet N blokkból áll, $MAC_{k'}(.)$ egy CBC-MAC és egyben az N+1-dik rejtjelezendő blokk, továbbá a rejtjelezésre (IV; k), míg a MAC számításra (IV'; k') (inicializáló vektor, kulcs) pár kerül alkalmazásra.
a.) Elfogadható-e, ha $IV = IV'$; $k = k'$ egyszerűsítő választással élünk? Indokoljon! (10p)
b.) Milyen tanulságot tud levonni ebből? (2p)

3. Egy böngésző és egy webszerver a TLS Handshake protokollt használják. A szerver aláírás ellenőrző kulcsot tartalmazó tanúsítvánnyal rendelkezik, a kliens nem rendelkezik semmilyen tanúsítvánnyal. Mi lesz a client-key-exchange üzenet tartalma, ha
a) RSA alapú kulcscserét használnak? (2p)
b) fix Diffie-Hellman kulcscserét használnak? (2p)
c) egyszeri (ephemeral) Diffie-Hellman kulcscserét használnak? (2p)
d) anonim Diffie-Hellman kulcscserét használnak? (2p)

4. Olyan jelszavas hitelesítő rendszerünk van, ahol a felhasználó jelszavát négy, a felhasználó által választott w_1, w_2, w_3, w_4 szó alkotja. A rendszer a felhasználó minden w_i szavához véletlen módon választ 63 ún. álca szót, legyenek ezek $d_{i,1}$, $d_{i,2}$, …, $d_{i,63}$, és a négy álca halmazt tárolja a jelszóval együtt. A hitelesítés négy körben történik. Az ellenőrző rendszer az i. körben megjeleníti w_i-t és a $d_{i,1}$, $d_{i,2}$, …, $d_{i,63}$ szavakat, de nem sorrendben, hanem valamilyen véletlen permutációban (pl. 8x8-as elrendezésben). A hitelesítés akkor sikeres, ha a felhasználó minden körben sikeresen kiválasztja a megjelenített szavak közül a saját szavát. Mekkora az on-line próbálgatás támadás átlagos komplexitása, ha
a) az ellenőrző egy sikertelen kör után azonnal hibát jelez és megszakítja a hitelesítést? (4 p)
b) az ellenőrző csak a negyedik kör végén ad információt a hitelesítés eredményéről? (4 p)
c) Gyengébb vagy erősebb lenne a rendszer, ha az álca halmazok nem fixek lennének, hanem minden hitelesítésnél frissen generált véletlen szóhalmazokat használnánk? Miért? (4 p)

5. Unix/Linux hozzáférésvédelem az órán előadott módon. Tekintsük az alábbi `/etc/passwd` file részletet:

```
u1:x:1003:1004:,,,:/home/u1:/bin/bash
u2:x:1004:1005:,,,:/home/u2:/bin/bash
u3:x:1005:1006:,,,:/home/u3:/bin/bash
u4:x:1006:1007:,,,:/home/u4:/bin/bash
```

Az `/etc/group` file releváns része:

```
u1:x:1004:
u2:x:1005:
u3:x:1006:
u4:x:1007:
g1:x:1008:u1,u2
g2:x:1009:u2,u3,u4
g3:x:1010:u2,u3
```

A fájl hozzáférési jogosultságok az alábbiak:

```
root@gotcha:/adatbizt# ls -la
total 16
drwxr-xr-x  4 root root 4096 2011-04-22 10:49 .
drwxr-xr-x 25 root root 4096 2011-04-22 10:51 ..
drwxrwsr-x  2 u1   g1   4096 2011-04-22 10:50 d1
drwxr-xr-x  2 u2   g1   4096 2011-04-22 10:50 d2

root@gotcha:/adatbizt# ls -la d1
total 20
drwxrwsr-x 2 u1   g1   4096 2011-04-22 10:50 .
drwxr-xr-x 4 root root 4096 2011-04-22 10:49 ..
-rw------- 1 u1   u4      4 2011-04-22 10:50 f1
-rw-rw---- 1 u1   g1     16 2011-04-22 10:50 f2
-rwxrwxrwx 1 u1   g2      8 2011-04-22 10:50 f3

root@gotcha:/adatbizt# ls -la d2
total 16
drwxr-xr-x 2 u2   g1   4096 2011-04-22 10:50 .
drwxr-xr-x 4 root root 4096 2011-04-22 10:49 ..
-rw-r--r-- 1 root g1      7 2011-04-22 10:50 f4
--w------- 1 root g1      6 2011-04-22 10:50 f5
```

a.) Mely felhasználók tudják olvasni a `d1/f1` fájlt és miért? (2p)
b.) Mely felhasználóknál fut le sikeresen a `cp d1/f2 d2/f6` parancs? (2p)
c.) Ki tudja módosítani az `f2` fájl jogosultságait (pl. `chmod o+w d1/f2`)? (3p)
d.) Ki tudja lefuttatni az `rm d1/f3; cp d1/f1 d1/f3` parancsokat mind sikeresen, azaz kitörölni `f3`-at és helyére `f1`-et másolni? (3p)

Pontozás: 1: 0-19, 2: 20-26, 3: 27-34, 4: 35-42, 5: 43-50

Adatbiztonság pót ZH megoldások

2012.május 17.

1.

a.) 65 E-6 b.) E-6

c.) Nem. A CBC mód hibaterjedés tulajdonsága szerint egy véletlen hiba esetén, hibázás utáni első blokk bitjeinek átlagosan fele hibás lesz, s még a rákövetkező blokk egy bitje. Ezt a nagymértékű meghibásodást csak igen költséges, komplex javító kóddal tudnánk eliminálni. A helyes megoldás a rejtjelezés utáni hibajavító kódolás alkalmazása.

d.) Igen. Nincs hibaterjedés a kulcsfolyamatos típusú rejtjelezés mód miatt. Ez esetben alkalmazhatjuk a hibajavítást a rejtjelezést megelőzően.

2.

a.) Nem. Tekintsünk pl. egy blokk hosszú *m* üzenetet. A CBC mód lépéseit végiggondolva látható, hogy a kódolás eredménye:

$$E_k(m+IV) \parallel E_k(0)$$

az üzenettől függetlenül, azaz semmiféle integritásvédelmet nem kapunk (itt E_k az ECB üzemű kódolást jelöli)

b.) A tanulság az, hogy ennél a megoldásnál, eltérő kulcsot kell alkalmazni a kétféle funkcióra Az IV megválasztásának szabadságát meghagyjuk az alkalmazásnak.

3.

a) Szerver RSA publikus kulcsával rejtjelezett pre-master titok, utóbbit a kliens generálja.
b) Kliens egyszer használatos DH publikus paramétere (g^x mod p).
c) Kliens egyszer használatos DH publikus paramétere (g^x mod p).
d) Kliens egyszer használatos DH publikus paramétere (g^x mod p).

4.

a) ½ x 4 x $2^6 = 2^7$
b) ½ x $(2^6)^4 = 2^{23}$
c) Gyengébb. A támadó több hitelesítést kezdeményez, az első körben megjelenő szóhalmazok metszete tartalmazza w_1-et, és ha nagy álca teret használunk, akkor pár futás után a metszet nagy valószínűséggel egy eleműre zsugorodik, azaz w_1 pár futás után megfejthető. Ha megvan w_1, akkor hasonlóképpen megfejthetjük w_2-t, majd ezek ismeretében w_3–at, és végül w_4–et.

5.

a.) csak u1, neki van olvasás joga és a directory-ba is be tud lépni
b.) u2, root, mert: g1 és u1 tudja olvasni f2-t, joguk van belépni d2-be, ott fájlt azonban u2 tud csak csinálni.
c.) csak a tulajdonos, u1 (és a root)
d.) a törlést csak u1,u2 tudja elvégezni f1-et viszont ebből csak u1 olvashatja f1-et, ő írni is tud az alkönyvtárba, tehát csak u1. (+root)