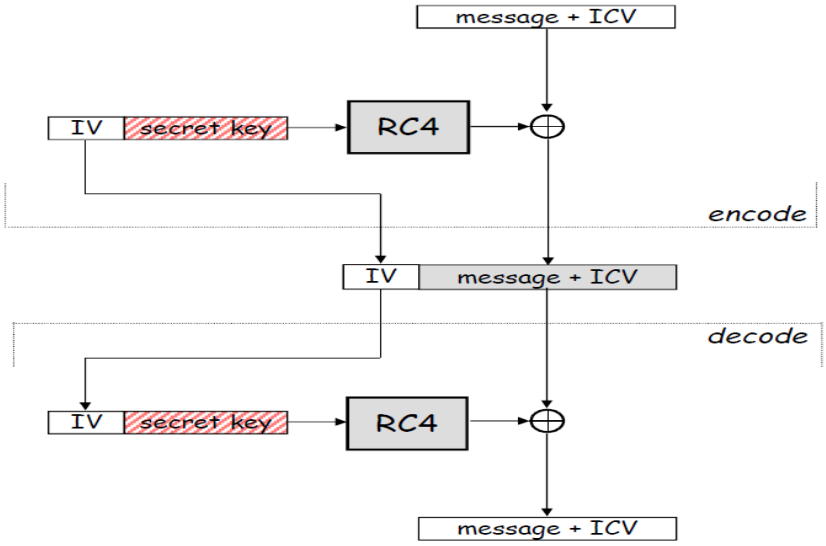
Válaszoljon a következő kérdésekre!

a) Hogyan működik az integritásvédelem a WEP protokollban? (4 pont)
b) Mutassa meg, hogyan tud a támadó tetszőleges módosítást végezni a WEP üzeneteken az integritásvédelem ellenére! (4 pont)
c) Hogyan védekezik a WEP az üzenet-visszajátszás ellen? (4 pont)
d) Tegyen javaslatot a visszajátszás elleni mechnizmus javítására! (4 pont)

Megoldás:

a)



b)

attacker can manipulate messages despite the ICV mechanism and encryption

- CRC is a linear function wrt to XOR:

$$CRC(X \oplus Y) = CRC(X) \oplus CRC(Y)$$

- attacker observes $(M \mid CRC(M)) \oplus K$ where K is the RC4 output
- for any ΔM, the attacker can compute $CRC(\Delta M)$
- hence, the attacker can compute:

$$((M \mid CRC(M)) \oplus K) \oplus (\Delta M \mid CRC(\Delta M)) =$$
$$((M \oplus \Delta M) \mid (CRC(M) \oplus CRC(\Delta M))) \oplus K =$$
$$((M \oplus \Delta M) \mid CRC(M \oplus \Delta M)) \oplus K$$

c) Sehogy, nincs visszajátszás elleni védelem.

d) Ezeket kell kijavítani:

- IV is not mandated to be incremented after each message
- receiver is not mandated to check the freshness of received IVs

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)

Megoldás:

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).

Háttérismeret:

RSA alapú
- a titkos kulcs (pre-master secret) titkosítva van a szerver RSA kulcsával

fix Diffie-Hellman
- szerver fix DH paraméterekkel (CA által aláírva)
- kliens lehet fix DH params vagy egyszeri DH a client_key_exchange

rövid életű Diffie-Hellman
- szerver és kliens is egyszeri DH params
- szerver aláírja a privát RSA kulcsával; kliens küldi a client_key_exchange-ben
- kliens auth. lehetséges

anonym Diffie-Hellman
- szerver és kliens is egyszeri DH params
- elküldik egymásnak auth nélkül

1. Egy webszerver és egy böngésző az SSL protokollt használja a HTTP forgalom védelmére. A handshake során Diffie-Hellman alapú kulcscserét használnak, és a szervernek egy DSA digitális aláírás ellenőrző kulcsot tartalmazó tanúsítványa van. A szerver nem kéri, hogy a kliens hitelesítse magát.
Adja meg, hogy ebben az esetben mely handshake üzenetek kerülnek átvitelre, és vázlatosan adja meg azok tartalmát! **(10p)**

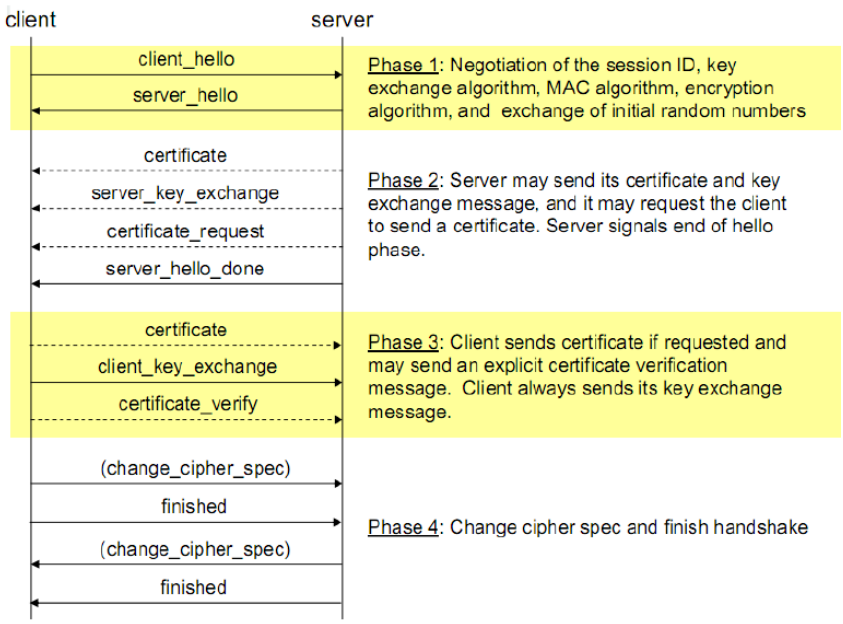
Megoldás:

$C \rightarrow S$:	client-hello	: kliens véletlenszáma, javasolt algoritmus-csokrok listája
$S \rightarrow C$:	server-hello	: szerver véletlenszáma, választott algoritmus-csokor, session ID
$S \rightarrow C$:	server-certificate	: szerver azonosító, szerver aláírás-ellenőrző kulcsa, CA aláírása
$S \rightarrow C$:	server-key-exchange	: szerver DH paraméterei: $p, g, g^x \bmod p$, szerver aláírása
$S \rightarrow C$:	server-hello-done	:
$C \rightarrow S$:	client-key-exchange	: kilens DH paramétere: $g^y \bmod p$
$C \rightarrow S$:	client-finished	: eddigi handshake üzeneteken és a mester titkon számolt MAC
$S \rightarrow C$:	server-finished	: eddigi handshake üzeneteken és a mester titkon számolt MAC

Háttérismeret:

TLS Handshake Protocol

client-hello: kliens verzió, véletlen szám, session id (ha új kapcsolatot akar akkor jelenlegi session id-t teszi bele, ha új session-t akar, akkor üres), lehetséges biztonsági algoritmusok felsorolása, tömörítő algoritmusok
server hello: szerver verzió, véletlen szám, session id (ha kliens kért, akkor kap új kapcsolatot [amennyiben nincs hely aktuális session-ben, új session-t kap], ha üres volt, új session-t), választott biztonsági és tömörítő algoritmus.



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)

Megoldás:

a) Alprotokollok:

- TLS Record: fragmentáció, tömörítés, rejtjelezés, üzenet-hitelesítés és integritásvédelem, visszajátszás elleni védelem
- TLS Handshake: algoritmusok egyeztetése, kulcscsere, partner-hitelesítés
- TLS Alert: hibaüzenetek
- 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	eddigi handshake üzeneteken és a mestertitkon számolt MAC (a korábban megegyezésre került algoritmusokkal, de új kulcsokkal védve)
S→C	server-finished	eddigi handshake üzeneteken és a mestertitkon számolt MAC (a korábban megegyezésre került algoritmusokkal, de új kulcsokkal védve)

3. Egy webszerver és egy böngésző a TLS protokollt használja a HTTP forgalom védelmére. A handshake során RSA alapú kulcscserében egyeznek meg. A szerver digitális aláírás ellenőrző kulcsot tartalmazó tanúsítvánnyal rendelkezik, és nem kéri, hogy a kliens hitelesítse magát. Adja meg, hogy ebben az esetben mely handshake üzenetek kerülnek átvitelre, és vázlatosan adja meg azok tartalmát! (8 p)

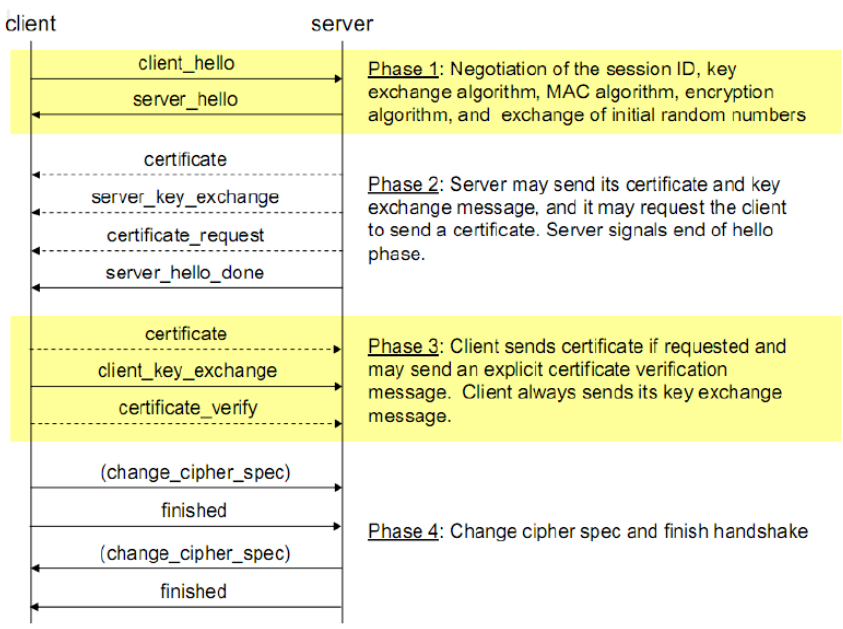
Megoldás:

A következő handshake üzenetek kerülnek átvitelre:
C → S: client-hello : kliens véletlenszáma, javasolt algoritmus-csokrok listája \\
S → C: server-hello : szerver véletlenszáma, választott algoritmus-csokor, session ID \\
S → C: server-certificate : szerver azonosító, szerver aláírás ellenőrző kulcsa, CA aláírása\\
S → C: server-key-exchange : szerver frissen generált RSA rejtjelező kulcsa, szerver aláírása \\
S → C: server-hello-done : \\
C → S: client-key-exchange : szerver RSA kulcsával rejtjelezett pre-master secret \\
C → S: (change cipher spec) \\
C → S: client-finished : eddigi handshake üzeneteken és a mester titkon számolt MAC \\
S → C: (change cipher spec) \\
S → C: server-finished : eddigi handshake üzeneteken és a mester titkon számolt MAC \\

Háttérismeret:

TLS Handshake Protocol

client-hello: kliens verzió, véletlen szám, session id (ha új kapcsolatot akar akkor jelenlegi session id-t teszi bele, ha új session-t akar, akkor üres), lehetséges biztonsági algoritmusok felsorolása, tömörítő algoritmusok
server hello: szerver verzió, véletlen szám, session id (ha kliens kért, akkor kap új kapcsolatot [amennyiben nincs hely aktuális session-ben, új session-t kap], ha üres volt, új session-t), választott biztonsági és tömörítő algoritmus.



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)

c) Hasonlítsa az a) és b) esetek erősségét egy 8 karakterből álló átlagos felhasználó által választott jelszó erősségéhez? **(2p)**

Megoldás:

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.

c) Átlagos felhasználó által választott jelszó erőssége: 4 + 7*2 = 18 bit, tehát erősebb mint a) és gyengébb mint b)

Háttérismeret:

Jelszó erőssége
Mérése
$H = L * \log_2 N$
H a jelszó entrópiája
felhasználók által választott jelszavak entrópiája nem számítható ezzel
1 -> 4bit; 2-8 ->2bit; 9-20 -> 1.5bit; 21-... -> 1bit

Tekintsük a következő interaktív grafikus jelszó sémát:

Jelszó-választás: A felhasználó jelszava k db ikonból áll, melyeket k lépésben választ ki a következő módon: A rendszer a felhasználói névből, mint magból generál n db különböző álvéletlen egész számot az $[1, m]$ intervallumban, majd ezeket indexként használva kiválaszt n db különböző ikont egy m méretű fix ikon halmazból, és véletlen elrendezésben egyszerre megjeleníti ezeket a felhasználó számára. A megjelenített n db ikonból a felhasználó kiválasztja a jelszó első ikonját. A kiválasztott ikon a következő iterációban magként szolgál, amiből a rendszer ismét generál n db különböző álvéletlen indexet, véletlen elrendezésben megjeleníti az ezekhez tartozó különböző ikonokat, s a felhasználó ezek közül kiválasztja a jelszó második ikonját. Ez a harmadik lépés magja, és így tovább, amíg a k db ikont ki nem választotta a felhasználó.

Normál használat: A felhasználó hitelesítése a jelszó-választáshoz hasonlóan k lépésben történik. A rendszer a felhasználói névből mint magból legenerálja ugyanazt az n db különböző álvéletlen egész számot, mint a jelszó-választás első lépésében, véletlen elrendezésben (ami lehet más mint a jelszó-választáskor) megjeleníti az ezekehez tartozó ikonokat, s a felhasználónak ezek közül ki kell választania a jelszava első ikonját. A kiválasztott ikonból, mint magból, a rendszer generálja a második kör különböző álvéletlen számait, és megjeleníti az ezekhez tartozó ikonokat, melyek közül a felhasználónak ki kell választani a jelszó második ikonját, és így tovább.

a) Ez a séma felismerés (recognition) vagy emlékezet (recall) alapú? [0.2 pont]

b) Mekkora egy adott felhasználó potenciális jelszavai halmazának mérete? [0.4 pont]

c) Mekkora jelszó-erősséget jelent ez bitekben mérve? [0.4 pont]

d) Tekintsük az alábbi két variánst:

(i) Hitelesítéskor, ha egy adott körben a felhasználó hibásan választ, a rendszer azonnal megszakítja a hitelesítést és "Failed login attempt." üzenetet küld.

(ii) Hitelesítéskor, ha egy adott körben a felhasználó hibásan választ, a rendszer nem szakítja meg a hitelesítést, hanem legenerálja a hibás választáshoz tartozó következő kört. Az így megjelenő ikonok természetesen különbözhetnek a helyes válasz esetén megjelenő ikonoktól. A hitelesítés így mindig pontosan k lépésből áll, s ha közben valahol nem jól választott a felhasználó, arról csak a k. kör végén értesül egy "Failed login attempt." üzenet által.

Melyik variáns nyújt nagyobb biztonságot, és miért? [1 pont]

Megoldás:

a) felismerés (recognition)

b) n^k

c) $k \cdot \log n$

d) A (ii) variáns biztonságosabb, mert annál a találgatásos támadás komplexitása n^k, míg az (i) variáns esetében ez a komplexitás csak nk: a támadó az i. ikont átlagosan $n/2$ próbálgatásból tudja megfejteni, miután az $i-1$. ikont már megfejtette.

4. *Melyik biztonságosabb? Számítással indokoljon!* (6 pont)

i) (a) egy 6 számjegyből álló véletlen PIN kód vagy (b) egy 4 alfanumerikus (case sensitive) karakterből álló véletlen jelszó?
ii) (a) egy 6 számjegyből álló véletlen PIN kód vagy (b) egy 8 karakterből álló felhasználó által választott jelszó?

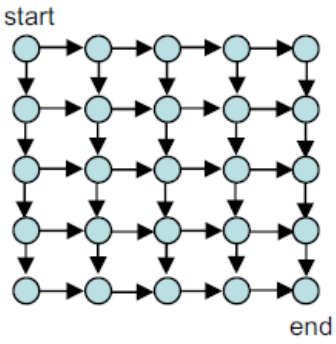
Megoldás:

i) (a) 6*log10 = 6*3.322 = 19.932 bit (b) 4*log62 = 4*5.954 = 23.816 bit
ii) (a) 19.932 bit (b) 4 + 7*2 = 18 bit

Háttérismeret:

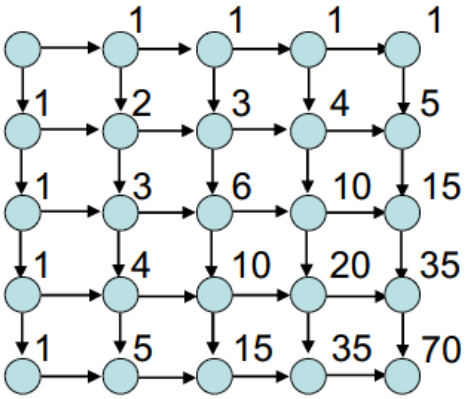
Jelszó erőssége
- Mérése
  - $H = L * \log_2 N$
  - H a jelszó entrópiája
  - felhasználók által választott jelszavak entrópiája nem számítható ezzel
    - 1 -> 4bit; 2-8 ->2bit; 9-20 -> 1.5bit; 21-... -> 1bit

4. Tekintsük a következő, mobiltelefonra vagy PDA-ra tervezett grafikus felhasználó-hitelesítési módszert: A képernyőn megjelenik az alábbi ábrán látható irányított gráf. A felhasználó jelszava ebben a gráfban egy általa korábban választott útvonal mely a bal felső sarokból indul és a jobb alsó sarokba érkezik. Belépéskor ezt az útvonalat kell a felhasználónak emlékezetből újra megrajzolni. *Hány számjegyből álló PIN kód erősségével ekvivalens erősségű ez a módszer?* **(10 p)**



Megoldás:

4. A bal felső sarokból az egyes csúcsokba vezető lehetséges útvonalak számát rekurzíven tudjuk kiszámolni: adott csúcsba vezető utak száma egyenlő a csúcsba mutató élek kezdőpontjába vezető utak számának összegével. Ez alapján a jelszótér mérete 70. Ezzel ekvivalens erősségű PIN kód hossza: $\log 70 / \log 10 = 1 + \log 7 / \log 10 < 2$



Háttérismeret:

Jelszó erőssége
- Mérése
  - $H = L * \log_2 N$
  - H a jelszó entrópiája
  - felhasználók által választott jelszavak entrópiája nem számítható ezzel
    - 1 -> 4bit; 2-8 ->2bit; 9-20 -> 1.5bit; 21-... -> 1bit

Biometriai azonosítás

a) Milyen tulajdonságokkal kell rendelkeznie egy emberi fiziológiai vagy viselkedési jellemzőnek ahhoz, hogy alkalmas legyen biometriai azonosításra? (4 p)
b) Soroljon fel legalább 5 biometriai azonosítási módszert! (3 p)
c) Milyen típusú ujjlenyomat minutiák léteznek és milyen paraméterekkel reprezentálják ezeket? (3 p)
d) Milyen makroszkopikus ujjlenyomat mintázatok léteznek? Soroljon fel legalább hármat! (3 p)
e) Ismertesse a minutia alapú matching eljárást! (4 p)
f) Milyen típusú hibákkal kell számolni a matching során? Mit jelentenek ezek a hiba fogalmak? (3 p)

Megoldás:

a) Szükséges tulajdonságok:

- minden ember rendelkezzen az adott jellemzővel (universality)
- az adott jellemző értéke különbözzön különböző emberekre (uniqueness)
- a jellemző ne változzon nagyon időben (permanence)
- a jellemző legyen jól mérhető (collectability)
- legyen nehéz hamisítani (circumvention)

b) ujjlenyomat, irisz minta, arc, fül geometria, recehártya (retina) képe, kéz geometria, hang, arc hőtérképe, billentyű-leütés dinamikája, kézi aláírás dinamikája és zaja

c) végződés (ending) és elágazás (bifurcation), paraméterek: a minutia típusa, pozíció X, Y koordinátája, és a minutia szöge

d) íves (arch), hurkos (loop), örvény (whorl)

e) Az ujjlenyomat matching azt próbálja meg eldönteni, hogy két ujjlenyomat azonos ujjról származik-e. A minutia alapú matching 4 lépésből áll:

- minutia párok közötti hasonlóság számolása
- két ujjlenyomat egymásra igazítása a leghasonlóbb minutia párok alapján
- összetartozó (matching) minutiák azonosítása
- globális hasonlósági mérték számolása a két ujjlenyomat között az összetartozó minutiák száma alapján. Ha a hasonlósági mérték egy küszöbszám felett van, akkor a két ujjlenyomat egyezik.

f) Hamis negatív / hamis visszautasítás / Type I: azonos ujjról származó két ujjlenyomat visszautasítása. Hamis pozitív / hamis elfogadás / Type II: különböző ujjakról származó két ujjlenyomat elfogadása egyezőnek

2. Tekintsünk egy szervert és két klienset A-t és B-t. A kliensek egymást segítve próbálják anonimizálni a szervernek küldött kéréseiket oly módon, hogy mindketten a kéréseiket 1/4 valószínűséggel a másik kliensen keresztül, 3/4 valószínűséggel közvetlenül küldik a szervernek. A továbbításra kapott kéréseket mindketten közvetlenül küldik ki. A kliens 1/4 valószínűséggel, B kliens 3/4 valószínűséggel bocsát ki kéréseket.
Melyik esetben nagyobb a szerver bizonytalansága (entrópia) az eredeti küldőre vonatkozóan: (a) ha A-tól vagy (b) ha B-től kap egy kérést? Számítással indokolja válaszát! **(15p)**

Megoldás:

2. feladat: Jelöljük az eredeti küldőt α-val, és azt a hosztot akitől a szerver a kérést megkapja ω-val. Továbbá jelöljük p_A-val annak valószínűségét, hogy az eredeti küldő A, p_B-vel annak valószínűségét, hogy az eredeti küldő B, p_{AB}-vel annak valószínűségét, hogy A a kérését B-n keresztül küldi, és p_{BA}-val annak valószínűségét, hogy B a kérését A-n keresztül küldi. Tudjuk, hogy $p_A = \frac{1}{4}$, $p_B = \frac{3}{4}$, $p_{AB} = \frac{1}{4}$, $p_{BA} = \frac{1}{4}$. Ekkor:

$$\begin{aligned}
\Pr\{\alpha = A|\omega = A\} &= \frac{\Pr\{\omega = A|\alpha = A\}\Pr\{\alpha = A\}}{\sum_{X\in\{A,B\}}\Pr\{\omega = A|\alpha = X\}\Pr\{\alpha = X\}} \\
&= \frac{(1-p_{AB})p_A}{(1-p_{AB})p_A + p_{BA}p_B} \\
&= \frac{\frac{3}{4}\frac{1}{4}}{\frac{3}{4}\frac{1}{4} + \frac{1}{4}\frac{3}{4}} \\
&= \frac{1}{2}
\end{aligned}$$

Hasonlóan:

$$\begin{aligned}
\Pr\{\alpha = B|\omega = A\} &= \frac{p_{BA}p_B}{(1-p_{AB})p_A + p_{BA}p_B} = \frac{1}{2} \\
\Pr\{\alpha = B|\omega = B\} &= \frac{(1-p_{BA})p_B}{(1-p_{BA})p_B + p_{AB}p_A} = \frac{9}{10} \\
\Pr\{\alpha = A|\omega = B\} &= \frac{p_{AB}p_A}{(1-p_{BA})p_B + p_{AB}p_A} = \frac{1}{10}
\end{aligned}$$

A szerver bizonytalansága az eredeti küldőre vonatkozóan ha A klienstől kapja a kérést:

$$\begin{aligned}
H &= -\sum_{X\in\{A,B\}}\Pr\{\alpha = X|\omega = A\}\log\Pr\{\alpha = X|\omega = A\} \\
&= -2\frac{1}{2}\log\frac{1}{2} = 1
\end{aligned}$$

A szerver bizonytalansága az eredeti küldőre vonatkozóan ha B klienstől kapja a kérést:

$$\begin{aligned}
H &= -\sum_{X\in\{A,B\}}\Pr\{\alpha = X|\omega = B\}\log\Pr\{\alpha = X|\omega = B\} \\
&= -\frac{1}{10}\log\frac{1}{10} - \frac{9}{10}\log\frac{9}{10} \approx 0.47
\end{aligned}$$

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)!

Megoldás:

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:

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

$$\begin{aligned}\Pr\{\alpha = B|\omega = A\} &= \frac{p_{BA}p_B}{(1-p_{AB})p_A + p_{BA}p_B} \\ \Pr\{\alpha = B|\omega = B\} &= \frac{(1-p_{BA})p_B}{(1-p_{BA})p_B + p_{AB}p_A} \\ \Pr\{\alpha = A|\omega = B\} &= \frac{p_{AB}p_A}{(1-p_{BA})p_B + p_{AB}p_A}\end{aligned}$$

A gyanún felüli anonimitási szint elérésének feltétele a következő:

$$\begin{aligned}\Pr\{\alpha = A|\omega = A\} &= \Pr\{\alpha = B|\omega = A\} \\ \Pr\{\alpha = B|\omega = B\} &= \Pr\{\alpha = A|\omega = B\}\end{aligned}$$

Felhasználva az előzőeket, ezek a feltételek így alakulnak:

$$\begin{aligned}(1-p_{AB})p_A &= p_{BA}p_B \\ (1-p_{BA})p_B &= p_{AB}p_A\end{aligned}$$

ahonnan:

$$p_A = p_{AB}p_A + p_{BA}p_B = p_B$$

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

4. Tekintsünk egy szervert és két klienst, *A*-t és *B*-t. A kliensek azonos valószínűséggel bocsátanak ki kéréseket a szerver felé. A kliensek egymást segítve próbálják anonimizálni a szervernek küldött kéréseiket a következő módon:
   - *A* a kéréseit p_A valószínűséggel *B*-n keresztül, 1-p_A valószínűséggel közvetlenül küldi a szervernek,
   - *B* a kéréseit p_B valószínűséggel *A*-n keresztül, 1-p_B valószínűséggel közvetlenül küldi a szervernek.

   a) Mi a feltétele annak, hogy a fenti rendszer elérje a „gyanún felüli" (beyond suspicion) küldő anonimitási szintet a szerverrel szemben? (10 p)
   b) Mekkora a szerverrel szembeni küldő anonimitás szintje bitekben mérve (entrópia) ha $p_A = p_B = ¼$ (10 p)

Megoldás:

4. a

Jelöljük az eredeti küldőt α-val, és azt a hosztot akitől a szerver a kérést megkapja ω-val.

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

Hasonlóan:

$$\begin{aligned}\Pr\{\alpha = B|\omega = A\} &= \frac{p_B}{1-p_A+p_B} \\ \Pr\{\alpha = B|\omega = B\} &= \frac{1-p_B}{1-p_B+p_A} \\ \Pr\{\alpha = A|\omega = B\} &= \frac{p_A}{1-p_B+p_A}\end{aligned}$$

A gyanún felüli anonimitási szint elérésének feltétele a következő:

$$\begin{aligned}\Pr\{\alpha = A|\omega = A\} &= \Pr\{\alpha = B|\omega = A\} \\ \Pr\{\alpha = B|\omega = B\} &= \Pr\{\alpha = A|\omega = B\}\end{aligned}$$

Ez akkor és csak akkor teljesül, ha $p_A + p_B = 1$.

Ha $p_A = p_B = \frac{1}{4}$, akkor

$$\begin{aligned}\Pr\{\alpha = A|\omega = A\} &= 1-p_A = \frac{3}{4} \\ \Pr\{\alpha = B|\omega = A\} &= p_B = \frac{1}{4}\end{aligned}$$

Ebből az entrópia a következő módon számolható:

$$\begin{aligned}H &= -\sum_{X\in\{A,B\}}\Pr\{\alpha = X|\omega = A\}\log\Pr\{\alpha = X|\omega = A\} \\ &= -\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4} \\ &= 0.25*2 + 0.75*0.415 \\ &= 0.81125\end{aligned}$$

4. Tekintsünk egy MIX alapú anonim kommunikációs rendszert! Tegyük fel, hogy a MIX-nek van egy RSA kulcspárja, és minden felhasználó ismeri a MIX (e, n) publikus kulcsát. A rendszer úgy működik, hogy a felhasználó az m üzenetét a MIX publikus kulcsával kódolja, azaz a tankönyvi RSA segítségével előállítja a $c = m^e \bmod n$ rejtjeles üzenetet, és c-t küldi a MIX-nek. A MIX dekódolja c-t, és m-et nyíltan küldi tovább az m-ben található eredeti címzettnek. Vissza irányú kommunikáció nincs, ezért m nem tartalmaz információt a küldőről. A MIX kötegelve dolgozik: 100 bejövő üzenetet mindig megvár (tfh. ezek 100 különböző felhasználótól érkeznek), s csak utána kezdi kiküldeni az ezekhez tartozó nyílt üzeneteket az eredeti címzetteknek megkevert sorrendben.
a) Milyen privacy-vel kapcsolatos célt próbál elérni ez a protokoll? (1p)
b) Eléri-e a protokoll a célját? Indokoljon! (6p)

Megoldás:

a) unlinkability of sender and receiver (globális lehallgató ellen), sender anonymity (fogadóval szemben) (½ p – ½ p)

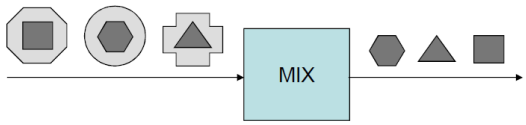
b) unlinkability: Nem. Bárki megfigyelheti a MIX-ből kimenő nyílt üzeneteket, azokat kódolhatja a MIX publikus kulcsával, és így megállapíthatja, hogy melyik kimenő üzenet melyik bemenő üzenethez tartozik. (4p)

Sender anonymity: igen, mert az üzenetek nem tartalmaznak információt a küldőre vonatkozóan. (2p)

Háttérismeret:

a MIX is a proxy that relays messages between communicating partners such that it

- changes encoding of messages
- batches incoming messages before outputting them
- changes order of messages when outputting them
- (may output dummy messages)



properties:

- sender anonymity w.r.t. communication partner
- unlinkability w.r.t. global (and hence local) eavesdroppers
- the MIX still needs to be trusted

defense against colluding compromised MIXes

- if a single MIX behaves correctly, unlinkability is still achieved