Tik-79.159: Tiedon salaus ja suojaus
Laskuharjoitusten mallivastaukset -97

$\boxed{\text{Laskuharjoitus 1}}$

Tehtävä 1

a) 10111101111000110001_2

b) 1111111111_2

c) 10000000000_2

d) 653

e) 3847

f) 2423

Tehtävä 2

a) $x \equiv 11 + 14 \pmod{2^4} \Rightarrow x \equiv 25 \pmod{16} \Rightarrow x = 25 + n \cdot 16$, josta pienin positiivinen x on $25 - 16 = 9$.

b) $x = (321 \cdot 123) \bmod 456 = 267$ (huomataan, että $267 + 86 \cdot 456 = 39483 = 321 \cdot 123$).

c) $x + 5 \equiv 4 \pmod{2^4} \Rightarrow x \equiv -1 \pmod{16} \Rightarrow x = -1 + n \cdot 16$, josta pienin positiivinen x on $-1 + 16 = 15$.

d) Todetaan poissulkevan tain ominaisuuksista: $a \oplus b = b \oplus a$, $a \oplus a = 0$, $a \oplus 0 = a$. Nyt $x \oplus 11100011_2 = 00100101_2 \Rightarrow x = 00100101_2 \oplus 11100011_2$, saadaan

$$\begin{aligned} &00100101_2 \\ \oplus &11100011_2 \\ x = &11000110_2 \end{aligned}$$

e) $114 = 01110010_2$, $211 = 11010011_2$. $114 \oplus 211 =$

$$\begin{aligned} &01110010_2 \\ \oplus &11010011_2 \\ = &10100001_2 = 161 \end{aligned}$$

f) $114 \text{ AND } 211 =$

$$\begin{aligned} &01110010_2 \\ \text{AND } &11010011_2 \\ = &01010010_2 = 82 \end{aligned}$$

Tehtävä 3

Lasketaan sekunnissa kokeiltavien avainten määrä:

$$100000\,\frac{keys}{ws \cdot s} \cdot 1000000\,ws = 10^{11}\,keys/s$$

n-bittisten avainten avainavaruus on kooltaan 2^n. Kun yhtä avainavaruudessa olevaa avainta haetaan, joudutaan keskimäärin hakemaan avaruuden puoleenväliin saakka, ts. kokeilemaan 2^{n-1} avainta.

avaimia	2^{40}	2^{56}	2^{72}	2^{128}
kokeillaan	2^{39}	2^{55}	2^{71}	2^{127}
10-kannassa	$10^{11.74}$	$10^{16.57}$	$10^{21.37}$	$10^{38.23}$
aika nyt	$10^{0.74}\ s \approx 5.5\ s$	$10^{5.57}\ s \approx 103\ h$	$10^{10.37}\ s \approx 743\ a$	$10^{27.23}\ s \approx 5.4 \cdot 10^{19}\ a$
aika 30 v. kuluttua	$10^{-5.26}\ s$	$10^{-0.43}\ s \approx 0.4\ s$	$10^{4.37}\ s \approx 6.5\ h$	$10^{21.23}\ s \approx 5.4 \cdot 10^{13}\ a$

Havaitaan, että 72-bittinen avain näyttää vielä olevan turvallinen, mutta 30 vuoden kuluttua (kun laskentatehon oletetaan olevan 10^6-kertainen), ei näin enää olekaan.

Tehtävä 4

Absoluuttinen informaatiosisältö $R = \log_2 4 = 2$.

a) Lasketaan yksittäisten merkkien informaatiosisällöt, eli kuinka monta bittiä jokin merkki sisältää informaatiota. Merkin sisältämä informaatio on sama, kuin sellaisen bittijonon pituus, jonka todennäköisyys satunnaisesti generoituna on sama kuin merkin. Esimerkiksi merkki, jonka todennäköisyys on 0.5, vastaa epäilemättä yhden bitin mittaista merkkijonoa, koska jokaisen tällaisen merkkijonon ("0" ja "1") todennäköisyys on 0.5. Samoin merkki, jonka todennäköisyys on 0.125 vastaa kolmen bitin mittaista merkkijonoa, koska minkä tahansa tietyn tällaisen kolmen bitin jonon todennäköisyys on $0.125 = \frac{1}{8}$. Yleisessä tapauksessa saadaan

$$H(X) = \log_2 \frac{1}{P}$$

josta

$$\begin{aligned} H(A) &= \log_2 \frac{1}{0.5} &= 1 \\ H(B) &= \log_2 \frac{1}{0.25} &= 2 \\ H(C) &= \log_2 \frac{1}{0.125} &= 3 \\ H(D) &= \log_2 \frac{1}{0.125} &= 3 \end{aligned}$$

Kielen *rate* r saadaan laskemalla

$$r = \lim_{|M| \to \infty} \frac{H(M)}{|M|}$$

Toisaalta taas pätee, että viestin kokonaisinformaatiosisältö on sama kuin sen sisältämien merkkien yksittäisten informaatiosisältöjen summa. (Huomaa, että merkin informaatiosisältö voi riippua edellisistä merkeistä—kts. kohta b).)

Pitkässä viestissä M on keskimäärin $0.5 \cdot |M|$ A-symbolia, $0.25 \cdot |M|$ B-symbolia, $0.125 \cdot |M|$ C-symbolia ja $0.125 \cdot |M|$ D-symbolia. Tästä saadaan raja-arvoksi

$$r = 0.5 \cdot 1 + 0.25 \cdot 2 + 2 \cdot 0.125 \cdot 3 = 1.75$$

Kielen informaatiosisältö on siis $r = 1.75$ ja redundanssiksi saadaan $R - D = 2 - 1.75 = 0.25$.

b) Lasketaan ensin keskimääräiset ehdolliset informaatiosisällöt kielen merkeille (kuinka paljon satunnainen X sisältää keskimäärin informaatiota kun edellinen merkki tunnetaan ja se on y).

$$H(X|y) = \sum_{m=A}^{D} P(m|y) \log_2 \frac{1}{P(m|y)}$$

(jos edellinen symboli on y, on seuraava symboli m todennäköisyydellä $P(m|y)$. Täten merkin m informaatiosisältö on y:n jälkeen $\log_2 \frac{1}{P(m|y)}$. Koska m esiintyy y:n jälkeen todennäköisyydellä $P(m|y)$, saadaan pitkässä viestissä ehdolliset todennäköisyydet kuten yllä.) Tulokset ovat

$$\begin{aligned} H(X|A) &= 1.75 \\ H(X|B) &= 2 \\ H(X|C) &= 2 \\ H(X|D) &= 1 \end{aligned}$$

Nyt kielen rate voidaan laskea kuten a)-kohdassa

$$r = \sum_{m=A}^{D} P(m) \cdot H(X|m) \tag{1}$$

Tätä varten on tunnettava todennäköisyydet $P(m)$. On ratkaistava yhtälöryhmä

$$\begin{cases} P(A) = P(A|A)P(A) + P(A|B)P(B) + P(A|C)P(C) + P(A|D)P(D) \\ P(B) = P(B|A)P(A) + P(B|B)P(B) + P(B|C)P(C) + P(B|D)P(D) \\ P(C) = P(C|A)P(A) + P(C|B)P(B) + P(C|C)P(C) + P(C|D)P(D) \\ P(D) = P(D|A)P(A) + P(D|B)P(B) + P(D|C)P(C) + P(D|D)P(D) \\ \quad 1 = P(A) + P(B) + P(C) + P(D) \end{cases}$$

Ratkaisemalla yhtälöryhmä esim. Gaussin eliminaatiolla saadaan

$$\begin{aligned} P(A) &\approx 0.276 \\ P(B) &\approx 0.293 \\ P(C) &\approx 0.259 \\ P(D) &\approx 0.172 \end{aligned}$$

josta (1):sta saadaan $r \approx 1.759$. Kielen redundanssi on $D = R - r \approx 2 - 1.759 = 0.241$.

Tehtävä 5

Salausfunktiot ovat bijektiivisiä funktioita 64-bittisiltä lohkoilta 64-bittisille lohkoille. Koska lähtö- ja maaliavaruudet ovat samat, ovat kyseessä permutaatiot. Joukon S permutaatioita on täsmälleen $|S|!$, joten salausfunktioita on $2^{64}!$.

128-bittisiä avaimia käyttävä salausmenetelmä kykenee tuottamaan näistä permutaatioista 2^{128}. Käytössä oleva osa on siis $\frac{2^{128}}{2^{64}!}$.

$$2^{64}! = 2^{64}(2^{64}-1)! \approx 2^{64}2^{64}(2^{64}-2)! = 2^{128}(2^{64}-2)!$$

joten käytössä on noin $\frac{1}{(2^{64}-2)!}$:s osa kaikista salausfunktioista, mikä on tietenkin mitättömän vähän.

Tehtävä 6

Tehtävässä käytetään Stirlingin kaavaa, joka antaa kertomafunktion approksimaation suurilla n:n arvoilla:

$$\boxed{n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n} \tag{2}$$

Todennäköisyys sille, että kaikki valitut alkiot ovat eri on

$$\begin{aligned} P_d &= \frac{n}{n} \cdot \frac{n-1}{n} \cdot \frac{n-2}{n} \cdots \frac{n-k+1}{n} \\ &= \frac{n!}{n^k(n-k)!} \end{aligned} \tag{3}$$

a) Sijoittamalla kaavaan (3) saadaan $P_d = 0.302$, ja haluttu todennäköisyys (vähintään kaksi alkiota samoja) $P = 1 - P_d = 0.698$.

b) Sijoitetaan (2) kaavaan (3), ja saadaan

$$P_d \approx \frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n}{n^k\sqrt{2\pi(n-k)}\left(\frac{n-k}{e}\right)^{n-k}}$$

Ottamalla tästä luonnollinen logaritmi saadaan

$$\ln P_d = (\ln n - \ln(n-k))(1/2 + (n-k)) - k \tag{4}$$

johon sijoittamalla $n = 10^6$ ja $k = 1000$ saadaan $\ln P_d \approx 0.49970005$, josta $P_d \approx 0.6067$ ja lopulta $P \approx 0.3933$.

$\boxed{\text{LASKUHARJOITUS 2}}$

Tehtävä 1

Kahden kierroksen Feistel-salausta kuvaavat yhtälöt

$$\begin{aligned} L_1 &= R_0 \qquad & R_1 &= L_0 \oplus f(R_0, K_0) \\ L_2 &= R_1 \qquad & R_2 &= L_1 \oplus f(R_1, K_1) \end{aligned}$$

jotka sieventyvät vasemmanpuoleisten yhtäsuuruuksien perusteella muotoon

$$\begin{aligned} L_2 &= L_0 \oplus f(R_0, K_0) \\ R_2 &= R_0 \oplus f(L_2, K_1) \end{aligned}$$

Oletetaan nyt, että hyökkääjä kykenee arvaamaan R_0:n. Tehdään valitun selväkielen hyökkäys: hyökkääjä lähettää viestin (X, R_0) kryptattavaksi (X, lohkon vasen puolisko, voi olla mitä tahansa). Kryptattu teksti on (L_2', R_2'), jolle pätee yllä olevan perusteella

$$L_2' = X \oplus f(R_0, K_0)$$

Koska vihollinen tuntee X:n, tuntee hän myös $X \oplus X \oplus f(R_0, K_0) = f(R_0, K_0)$. Mutta $L_2 = L_0 \oplus f(R_0, K_0)$, joten hyökkääjä kykenee laskemaan myös L_0:n tuntiessaan L_2:n (joka on cipherteksti ja joka oletetaan tunnetuksi). Näin ollen myös alunperin salaisen viestin vasen puolisko L_0 on paljastunut.

Tehtävä 2

Käytetään operaattoria yhteenlasku modulo 2^{32}. Merkitään tätä merkillä $\boxplus$. $\boxplus$ toteuttaa ryhmäaksioomat, nimittäin liitännäisyyden: $(a \boxplus b) \boxplus c = a \boxplus (b \boxplus c)$, neutraalialkion olemassaolon: $a \boxplus 0 = a$, sekä käänteisalkion olemassaolon: $a \boxplus (2^{32} - a) = 0$. Salakirjoituksen sekoitusfunktion ei tarvitse toteuttaa mitään ominaisuuksia, joita vastaavassa XOR-operaattoriin perustuvassa Feistel-salakirjoituksessa sen ei tarvitsisi täyttää. Huomaa, ettei sen enempää XOR- kuin $\boxplus$-operaattoriin perustuvan salauksen turvallisuus johdu käytetyn operaattorin luonteesta. Ryhmäominaisuus on välttämätön mutta myös riittävä. Jos ryhmäominaisuus puuttuu, ei salausta välttämättä kykene purkamaan.

Tehtävä 3

Lohkon oikeassa puoliskossa kolmen kierroksen ja lohkon vasemmassa puoliskossa neljän kierroksen jälkeen.

Tehtävä 4

Jos salakirjoitus on ryhmä, niin on olemassa avain K siten, että

$$E_{K_2}(E_{K_1}(P)) = E_K(P)$$

joten

$$E_{K_3}(E_{K_2}(E_{K_1}(P))) = E_{K_3}(E_K(P))$$

Uudelleen ryhmäominaisuudesta seuraa, että on olemassa avain K' siten, että

$$E_{K_3}(E_K(P)) = E_{K'}(P) = E_{K_3}(E_{K_2}(E_{K_1}(P)))$$

joten jos salaus on ryhmä on se myös *pure*.

Jos salakirjoitus on pure, ei se kuitenkaan välttämättä ole ryhmä. Esimerkiksi salakirjoitus, joka sisältää vain yhden avaimen K, ja jolle pätee

$$E_K(P) \neq P \qquad \text{sekä} \qquad E_K(E_K(P)) = P$$

(tällainen olisi esimerkiksi yksi one-time pad), on kyllä pure:

$$E_K(E_K(E_K(P))) = E_K(P)$$

mutta ei ryhmä, koska

$$E_K(E_K(P)) = P$$

eikä P:n kryptaaminen ainoalla avaimella K anna tulokseksi P:tä.

Tehtävä 5

CFB-moodia kuvaa yhtälö

$$C_i = P_i \oplus E(C_{i-1}) \tag{5}$$

Lisätään tähän ECB-moodin kryptaus lopuksi:

$$C_i' = E(C_i)$$

missä C_i':t ovat salakielilohkot ECB-moodin kryptauksen jälkeen, joten $C_i = D(C_i')$. Sijoittamalla nämä (5):een saadaan

$$C_i' = E(P_i \oplus E(D(C_{i-1}'))) = E(P_i \oplus C_{i-1}')$$

jonka havaitaan olevan CBC-moodin. Tehtävä kannattaa yrittää ratkaista myös graafisesti—piirrä kuva.

Tehtävä 6

Ratkaistaan inverssit käyttäen laajennettua Euklideen algoritmia:

a)

$$\begin{aligned}
19 &= 3(6) + 1 \Rightarrow \\
1 &= 19 - 3(6) \Rightarrow \\
3(6) &= 19 - 1 \Rightarrow \\
3(6) &\equiv -1 \pmod{19} \Rightarrow \\
-3 \cdot 6 &\equiv 1 \pmod{19} \Rightarrow \\
6^{-1} &\equiv -3 \pmod{19}
\end{aligned}$$

b)

$$\begin{aligned}
65537 &= 655(100) + 37 \\
100 &= 2(37) + 26 \\
37 &= 1(26) + 11 \\
26 &= 2(11) + 4 \\
11 &= 2(4) + 3 \\
4 &= 1(3) + 1 \Rightarrow
\end{aligned}$$

$$\begin{aligned}
1 &= 4 - 1(3) \\
&= 4 - (11 - 2(4)) = -1(11) + 3(4) \\
&= -1(11) + 3(26 - 2(11)) = 3(26) - 7(11) \\
&= 3(26) - 7(37 - 1(26)) = -7(37) + 10(26) \\
&= -7(37) + 10(100 - 2(37)) = 10(100) - 27(37) \\
&= 10(100) - 27(65537 - 655(100)) = -27(65537) + 17695(100)
\end{aligned}$$

$$\Rightarrow 100^{-1} \equiv 17695 \pmod{65537}$$

$\boxed{\text{Laskuharjoitus 3}}$

Tehtävä 1

a)

$$\begin{aligned}
65537 &= 64303 + 1234 \\
64303 &= 52(1234) + 135 \\
1234 &= 9(135) + 19 \\
135 &= 7(19) + 2 \\
19 &= 9(2) + 1 \Rightarrow
\end{aligned}$$

$$\begin{aligned}
1 &= 19 - 9(2) \\
&= -9(135) + 64(19) \\
&= 64(1234) - 585(135) \\
&= -585(64303) + 30484(1234) \\
&= 30484(65537) - 31069(64303)
\end{aligned}$$

$$\Rightarrow 65303^{-1} \equiv -31069 \equiv 34468 \pmod{65537}$$

b)

$$\begin{aligned}
\gcd(2679, 2397) &= \gcd(2397, 282) \\
&= \gcd(282, 141) \\
&= 141 \neq 1
\end{aligned}$$

joten ratkaisua ei ole.

Tehtävä 3

Kahden avaimen EDE-moodissa on kaksi avainta x ja y. P kryptattuna on

$$E_x(D_y(E_x(P)))$$

Käydään läpi kaikki 2^{56} single-DES -avainta, ja tallennetaan taulukkoon $D_z(0)$ kaikille single-DES -avaimille z (taulukon koko 2^{56}). Salautetaan sen jälkeen kaikki nämä lohkot hyökkäyksen kohteena olevalla taholla, joka tuntee siis avaimet x ja y. Ciphertekstit ovat muotoa

$$E_x(D_y(E_x(D_z(0))))$$

Dekryptataan jokainen saatu ciphertekstin lohko avaimella z; saadaan

$$D_z(E_x(D_y(E_x(D_z(0))))) \tag{6}$$

Nyt, *jos* tietty z on sama kuin x, niin $D_z \hat{=} D_x$ ja $E_z \hat{=} E_x$, jolloin (6) sievenee muotoon

$$D_x(E_x(D_y(E_x(D_x(0))))) = D_y(0)$$

Mutta alussa oli luotu taulukko, jossa ovat $D_k(0)$ kaikille k. Etsimällä taulukosta rivi, jolla $D_k(0) = D_y(0)$ saadaan selville, että ensinnäkin $z = x$ ja toisaalta, että $y = k$, joten molemmat EDE-moodin avaimet x ja y on ehkä saatu selville (voi olla, että löytyy pari, joka toteuttaa kyllä tämän mutta on silti "väärä hälytys". Pari täytyy testata myös jollakin toisella tunnetulla selväkieli-salakieli -parilla, ja jatkaa etsintää, jos tämä osoittaa parin virheelliseksi).

Tehtävä 4

Joka toisella kierroksella differenssi menee lävitse todennäköisyydellä yksi ja joka toisella todennäköisyydellä 0.034. Todennäköisyyden, että differenssi näkyy salauksen tuloksessa, tulee olla pieni, koska muutoin voidaan differenssin perusteella tehdä päätelmiä viimeisten salauskierrosten aliavaimista. Todennäköisyyden sille, että differenssi säilyy salauksen viimeisille kierroksille tulee olla samaa suuruusluokkaa kuin satunnaisesti valitun avaimen todennäköisyyden, jotta differentiaalinen kryptoanalyysi ei olsi brute-force -hyökkäystä parempi.

Saadaan $0.034^{k-2} = \frac{1}{2^{128}}$ josta $k = -256\frac{\ln 2}{\ln 0.034} \Rightarrow k \geq 53$.

Tehtävä 5

On laskettava xy mod 65537, kun $1 \leq x, y \leq 65536$.

Lasketaan ensin $a = xy$ mod 65536. Koska $65536 = 2^{16}$, voi tämän toteuttaa tehokkaasti. Samoin lasketaan $k = \left\lfloor \frac{xy}{65536} \right\rfloor$. Tämän voi toteuttaa myös tehokkaasti shiftaamalla. Nyt pätee

$$a - k \equiv xy \pmod{65537} \tag{7}$$

mikä on tietenkin todistettava. Ylläolevan perusteella pätee, että

$$xy = a + k \cdot 65536$$

joten (7) voidaan kirjoittaa muotoon

$$a - k \equiv (a + k \cdot 65536) \pmod{65537} \tag{8}$$

Mutta $65536 \equiv -1 \pmod{65537}$, joten (8) voidaan kirjoittaa muotoon

$$a - k \equiv (a + k \cdot -1) \equiv a - k \pmod{65537}$$

—identiteetti, josta väite seuraa. Tehokkaasti laskenta tapahtuu siis esimerkiksi näin:

```
unsigned int mul_mod_65537(unsigned int x, unsigned int y)
{
  unsigned int k, m;
  int a;
  m = x*y;
  a = m & 65535;    /* m mod 65536 */
  k = m >> 16;           /* m div 65536 */
  if (k == 0) k = 65536; /* XXX */
  a -= k;
  return(a < 0 ? a + 65537 : a);
}
```

Tässä joudutaan XXX:llä merkityllä rivillä tarkastamaan erikoistapaus: jos $x = y = 2^{16}$, on $xy = m = 2^{32}$ joka on nolla ylivuodon johdosta 32-bittisillä suorittimilla. Tällöin k saa myös virheellisesti arvon nolla, vaikka tulisi olla $k = 65536$. Viimeisellä rivillä tiedetään vasta, että $a - k \equiv xy \pmod{65537}$, mutta $a - k$ voisi olla negatiivinen. Koska halutaan palauttaa positiivinen tulos, lisätään negatiiviseen $a - k$:hon moduli yhden kerran.

Tehtävä 6

Tunnetaan selväkieli-salakielipari P_0, C_0, ts. joillekin salaisille DES-avaimille x ja y pätee $E_y(E_x(P_0)) = C_0$ ja $D_x(D_y(C_0)) = P_0$. Hyökkäyksessä kryptataan P_0 eri avaimilla K_1 ja C_0 eri avaimilla K_2, pyrkien löytämään avainpareja K_1, K_2 joille pätisi

$$E_{K_1}(P_0) = D_{K_2}(C_0) \Rightarrow E_{K_2}(E_{K_1}(P_0)) = C_0 \tag{9}$$

Kun tällainen avainpari löytyy, voi olla että $K_1 = x$ ja $K_2 = y$ (jos näin on, (9) toteutuu ainakin, mutta tällaisia pareja on muitakin).

Kysymys on: kuinka monta sellaista paria K_1, K_2 on olemassa, jotka toteuttavat yhtälön (9)?

C_0 dekryptautuu 2^{56} eri viestiksi ja P_0 kryptautuu 2^{56} eri viestiksi (koska avainavaruus on kooltaan 2^{56}). Voidaan olettaa, että nämä kaksi viestien osajoukkoa ovat satunnaisia osajoukkoja kaikkien 2^{64} eri 64-bittisten lohkojen joukolta. Haettujen parien määrä on sama, kuin viestien lukumäärä näiden kahden osajoukon leikkauksessa. Leikkauksen kooksi saadaan

$$\left(\frac{2^{56}}{2^{64}}\right)^2 \cdot 2^{64} = 2^{48}$$

Näistä pareista käydään keskimäärin puolet lävitse ennen oikeaa ratkaisua, joten tehtävän vastaus on 2^{47}.

$\boxed{\text{Laskuharjoitus 4}}$

Tehtävä 1

Piirretään kuva:

```
                               shift left
        +---+---+---+---+---+---+---+---+
<-----  | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |<-\
        +-+-+---+---+---+-+-+-+-+-+-+---+  |
          |               |   |   |        |
          \--------------(X)-(X)-(X)-------/
```

Olkoon alussa rekisterin arvo `11111111`. Rekisterin seuraaviksi arvoiksi saadaan nyt

```
11111111
11111110
11111100
11111001
11110010
11100100
11001000
10010000
00100000
01000000
10000000
00000001
00000010
00000101
00001011
...
```

Generoitu bittivirta syntyy rekisteristä vasemmalta poistuvista biteistä, joten virran ensimmäiset bitit `111111110010000`.

Polynomin asteluku on 8, joten, koska polynomi on primitiivinen, on virran jaksonpituus $2^8 - 1 = 255$. Tämä on hyvin lyhyt jaksonpituus. Lineaarisen siirtorekisterin generoimaa bittivirtaa ei voi yleensäkään käyttää sellaisenaan salaukseen XORaamalla sitä selvätekstin kanssa. Syy tähän selviää tehtävässä 3. Muihin kryptografisiin sovellutuksiin rekisteri ei luultavasti sovi lyhyen jaksonpituuden vuoksi.

Tehtävä 2

A5:ssä on kolme siirtorekisteriä, joiden pituudet ovat 19, 22 ja 23 bittiä. Täten rekistereillä yhteensä on korkeintaan

$$(2^{19}-1)(2^{22}-1)(2^{23}-1) = 18446702292280803327 \approx 1.84 \cdot 10^{19}$$

tilaa, joten syklien pituus ei voi ylittää $1.84 \cdot 10^{19}$:ta. Tämä on vain yläraja, lyhyempiä syklejä voi hyvin löytyä. 112-bittisten lohkojen salaamiseen tällainen jaksonpituus on varmasti riittävä.

Sen sijaan lohkonumeroiden toistuminen kolmen ja puolen tunnin välein aiheuttaa ongelmia, koska jokaisen lohkon alussa siirtorekisterit alustetaan uudelleen arvolla, joka riippuu lohkonumerosta. Täten yli kolme ja puoli tuntia kestävien yhteyksien aikana voivat samat bittivirrat toistua. Bittivirtasalauksesta tiedetään, että saman virran käyttäminen kahteen kertaan salauksessa on vaarallista (kuten one-time -padeilla).

Tehtävä 3

Merkkiä `A` vastaa ASCII-arvo 65 ja merkkiä `B` ASCII-arvo 66. Kun salasanoma ja vastaavan selväkielen alku tunnetaan, saadaan bittivirran alku selville:

```
salasanoma 10111000 11100110
selväkieli 01000001 01000010 (65, 66)
bittivirta 11111001 10100100
```

Nyt tiedetään, että siirtorekisterissä oli aluksi `11111`, ja sen jälkeen arvot kehittyvät selvitetyn bittivirran mukaisesti:

```
11111 00110100100
11110 0110100100
11100 110100100
11001 10100100
10011 0100100
00110 100100
01101 00100
11010 0100
10100 100
01001 00
10010 0
...
```


Rekisteristä tiedetään, että sen pituus on viisi (koska polynomin asteluvun ilmoitettiin olevan viisi). Mallinnetaan rekisterin oikeanpuoleisen, uuden bitin generointia yhtälöllä

$$newbit = a_5 \cdot p_5 \oplus a_4 \cdot a_4 \oplus a_3 \cdot a_3 \oplus a_2 \cdot a_2 \oplus a_1 \cdot a_1$$

missä p_n ovat rekisterin bittien arvot ja a_n on yksi jos bittiä n käytetään uuden bitin laskemiseen, muutoin nolla. Rekisterin ensimmäisiä arvoja tunnetaan. Otetaan niistä viisi ensimmäistä, ja muodostetaan niiden perusteella yhtälöryhmä:

$$\begin{cases} (a_5 \cdot 1) \oplus (a_4 \cdot 1) \oplus (a_3 \cdot 1) \oplus (a_2 \cdot 1) \oplus (a_1 \cdot 1) = 0 \\ (a_5 \cdot 1) \oplus (a_4 \cdot 1) \oplus (a_3 \cdot 1) \oplus (a_2 \cdot 1) \oplus (a_1 \cdot 0) = 0 \\ (a_5 \cdot 1) \oplus (a_4 \cdot 1) \oplus (a_3 \cdot 1) \oplus (a_2 \cdot 0) \oplus (a_1 \cdot 0) = 1 \\ (a_5 \cdot 1) \oplus (a_4 \cdot 1) \oplus (a_3 \cdot 0) \oplus (a_2 \cdot 0) \oplus (a_1 \cdot 1) = 1 \\ (a_5 \cdot 1) \oplus (a_4 \cdot 0) \oplus (a_3 \cdot 0) \oplus (a_2 \cdot 1) \oplus (a_1 \cdot 1) = 0 \end{cases}$$

(Tässä yhtälön oikealla puolella on seuraavaksi rekisteriin generoitu uusi bitti). Ratkaisemalla yhtälöryhmä tuntemattomien $a_1, \ldots, a_5$ suhteen saadaan

$$a_5 = a_2 = 1 \qquad a_4 = a_3 = a_1 = 0$$

joten karakteristinen polynomi on (5, 2, 0). Nyt bittivirtaa voidaan generoida eteenpäin mielivaltaisen pitkälle, ja salakielinen sanoma voidaan murtaa. Selväkielinen teksti on `ABDUL`.

Tehtävä 4

a) Lukujonon $x, H(x), H(H(x)), \ldots$ tulee näyttää tilastollisesti satunnaiselta. Lisäksi on eduksi, jos funktio on yksisuuntainen (ts. $H(x)$:n tunteva ei voi helposti arvata x:ää).

b) Tässä on ajatuksena, että A todistaa tuntevansa salaisen avaimen K B:lle seuraavasti: B lähettää A:lle haasteen (satunnaisluvun) R. A vastaa B:lle $H(K, R)$, ja B tarkastaa että saatu arvo vastaa B:n itsensä laskemaa $H(K, R)$:n arvoa.

   H:n tulee olla sellainen, että $H(K, R)$:n laskeminen on hyvin hankalaa mikäli K:ta ei tunne, myös, vaikka suuri joukko pareja $H(K, X)$ tunnettaisiinkin vastaavien X:n arvojen kanssa (mutta K olisi edelleen salainen). Lisäksi funktion tulee olla siinä mielessä yksisuuntainen, että samasta joukosta $H(K, X)$:n arvoja salaisen avaimen K:n laskeminen on hyvin hankalaa.

c) Tässä funktiolle tulevat samat vaatimukset kuin edellisessä kohdassa. Tehtävässä mainitun “yhteisen salaisuuden” oletetaan pysyvän vakiona pitkään, joten se ei saa paljastua vaikka moni vanha istuntoavain murrettaisiinkin.

d) Tässä tarkoitetaan funktiota H, joka lasketaan viestistä M ennenkuin se allekirjoitetaan. Samoin perustein kuin kohdassa b) on funktion oltava yksisuuntainen. Funktion on oltava vahvasti törmäyksetön, jotta allekirjoitusta ei voida siirtää toiseen dokumenttiin. Lisäksi funktion on hyvä olla kompressoiva (pienempi maali- kuin lähtöavaruus), jotta se olisi käytännöllinen. (Esim. n-bittisellä RSA-avaimella voi allekirjoittaa vain alle n-bittisiä viestejä. Jos $H(M)$ on kooltaan samaa luokkaa kuin M, tarvitaan megatavun mittaisen viestin allekirjoittamiseen joko yksi miljoonabittinen avain tai sitten se täytyy allekirjoittaa lohkoissa, mikä on Huono Asia.)

e) Funktion täytyy olla heikosti törmäyksetön, jotta ohjelmakoodia ei voi myöhemmin muuttaa ilman, että sormenjälki muuttuu. Vahvasti törmäyksetön sen ei tarvitse olla (miksi?) Yksisuuntaisuutta ei sinänsä tarvita, esimerkiksi ohjelmakoodin kopioiminen toiseen paikkaan voisi toimia hyvin tällaisena turvatoimena, joskin se olisi tilankäytön suhteen tarpeettoman kallista—kompressoivuus olisikin funktiolle eduksi.

f) Funktion täytyy olla yksisuuntainen, mikäli itse dokumenttia ei haluta (vielä) paljastaa. Joka tapauksessa on funktion oltava heikosti törmäyksetön, jotta aikaleimalla olisi katetta (muutenhan voitaisiin julkaista mikä tahansa aikaleima, ja myöhemmin luoda siihen sopiva dokumentti). Vahvasti törmäyksetön funktion ei tarvitse olla, koska vahva törmäyksettömyys tarkoittaa sitä, että on hyvin vaikeaa löytää mitään satunnaista paria M, M' s.e. $H(M) = H(M')$. Nyt jos joku luo tällaisen parin, ja aikaleimaa parin toisen puolen M, on toinenkin viesti M' jo olemassa, ja se voitaisiin yhtä hyvin leimata samantien.

Tehtävä 5

Todennäköisyys sille, että kaikki $\sqrt{n}$ satunnaisesti valittua alkiota olisivat eri, on

$$1(1-\frac{1}{n})(1-\frac{2}{n})\cdots(1-\frac{\sqrt{n}-1}{n})$$

Approksimoidaan todennäköisyyttä käyttämällä approksimaatiota

$$1-a \approx e^{-a} \quad \text{pienille } a \tag{10}$$

Saadaan:

$$e^0(e^{-\frac{1}{n}})(e^{-\frac{2}{n}})\cdots(e^{-\frac{\sqrt{n}-1}{n}})$$

joka on

$$\begin{aligned} e^{-\frac{1+2+\cdots+(\sqrt{n}-1)}{n}} &= e^{-\frac{\sqrt{n}/2\cdot(\sqrt{n}-1)}{n}} \\ &= e^{-\frac{(n-\sqrt{n})/2}{n}} \\ &= e^{-\frac{1}{2}-\frac{1}{2\sqrt{n}}} \end{aligned}$$

mutta

$$\lim_{n\to\infty}\frac{1}{\sqrt{n}} = 0$$

joten suurilla n:n arvoilla lausekkeen voi edelleen sieventää muotoon

$$e^{-\frac{1}{2}}$$

joka on n. 0.607, joten todennäköisyys sille, että alkio tulee vähintään vähintään kahdesti, tämän komplementtitodennäköisyys, on n. 0.393. Todennäköisyys on hiukan liian pieni, koska käytetty approksimaatio (10) suurensi ensimmäistä todennäköisyyttä.

Tehtävä 6

Salauslohkon hyvälle pituudelle antavat alarajan sanakirjahyökkäyksen mahdottomuus (liian pieni lohko aiheuttaa sen, että kaikki mahdolliset selväkieli-salakieli -parit voidaan kerätä ja tallettaa) ja ylärajan diffuusionopeus lohkosalauksessa (kun lohkon pituus kasvaa, kasvaa myös tarvittavan laskennan määrä, jotta lohko olisi hyvin sekoittunut).

Hajautusfunktion tuloksen pituuden alarajan antaa syntymäpäivähyökkäyksen mahdottomuus (n. $\sqrt{n}$ alkiota täytyy käydä lävitse ennenkuin yksi törmäys on löytynyt) ja ylärajan käytännön vaatimukset (liian pitkä sormenjälki ei ole käytännöllinen).

Jos lohkosalauksen lohko on n-bittinen, täytyy lävitse käydä $O(2^n)$ lohkoa ennenkuin löytyy haluttua salakieltä vastaava selväkielipari (jos avainta ei tunneta, mutta salakirjoitus tarjotaan—valitun selväkielen hyökkäys), ja jos hash-funktion maaliavaruus on m-bittinen, täytyy lävitse käydä $O(\sqrt{2^m})$ yritettä, ennenkuin törmäys on löytynyt. Ongelmat tulevat yhtä raskaiksi, kun $m \approx n^2$. Näin on esim. kun $n = 64$ ja $m = 128$. Toisaalta hash-funktiota vastaan hyökkääminen voi olla helpompaa kuin lohkosalausta, koska hash-funktiota vastaan voi hyökätä ilman, että hyökkäyksen kohteena oleva osapuoli osallistuu mitenkään (esim. valitun selväkielen hyökkäys vaatii kohteen aktiivista toimintaa), ja joillakin uusilla hash-funktioilla (esim. SHA) onkin 160-bittinen maaliavaruus.

$\boxed{\text{Laskuharjoitus 5}}$

Tehtävä 1

CBC-salaus:

$$C_i = E(P_i \oplus C_{i-1}) \tag{11}$$
$$P_i = C_{i-1} \oplus D(C_i) \tag{12}$$

Olkoon olemassa jokin selväkielinen viesti $P_1, P_2, \ldots, P_k$. Lasketaan sille CBC-MAC samalla avaimella, jolla varsinainen kryptaus tullaan suorittamaan. Tämä MAC on, tehtävän kuvauksen mukaisesti, viimeinen salakielinen lohko, eli tässä tapauksessa C_k.

C_k liitetään selväkielen perään eheystarkastusta varten: $P_{k+1} = C_k$. Tämän jälkeen selväkielinen viesti MACin kanssa $(P_1, \ldots, P_{k+1})$ salataan CBC-moodissa, ja saadaan salakielinen viesti $C_1, \ldots, C_{k+1}$.

Välittömästi havaitaan, että

$$\begin{aligned} C_{k+1} &= E(P_{k+1} \oplus C_k) \\ &= E(C_k \oplus C_k) \\ &= E(0) \end{aligned}$$

joten salakielen viimeinen lohko on aina $E(0)$! Tämä kielii jo ongelmista.

Vastaanottajan on tarkoitus dekryptata salakielinen viesti ja sen jälkeen tarkistaa MAC. Kun salakieli dekryptataan, saa vastaanottaja MAC-lohkon arvoksi

$$\begin{aligned} P_{k+1} &= C_k \oplus D(C_{k+1}) \\ &= C_k \oplus D(E(0)) \\ &= C_k \end{aligned}$$

joka on oikein.

Ongelma on tämä: MACin sisältävän salakielen viimeinen lohko, jonka pitäisi välittää MAC, ei sisällä bittiäkään informaatiota, koska se on vakio: $E(0)$. Tämä johtuu saman avaimen käytöstä sekä salaukseen että integriteetin suojaamisen näillä algoritmeilla kuten yllä on esitetty. Jos vihollinen muuttaa salakielen jotakin lohkoa, tietää hän, että viimeisen lohkon pysyessä samana havaitsee vastaanottaja edelleen oikean MAC-tarkistussumman ja hyväksyy muutellun viestin.

Tehtävä 2

CRC-eheystarkastuksen idea on, että viesti, jonka perässä on CRC-tarkistussumma, on tarkistussumman kanssa jaollinen CRC-polynomilla modulo 2.[1] Käytännössä tämä tarkoittaa sitä, että kun viestin ja tarkistussumman päälle XORataan sopiviin kohtiin CRC-polynomia, saadaan tulokseksi pelkkiä nollia.

Tästä seuraa, että

> Jos A ja B ovat viestejä, joilla ovat validit CRC-tarkistussummat viestin lopussa, on myös $A \oplus B$ sellainen.

Tämän suora sovellus on seuraava: oletetaan, että meillä on jokin viesti M (esimerkiksi tehtävän pankkitransaktio) jonka perässä on tarkastussumma S. Jos viestiä halutaan muuttaa XORaamalla se M:n mittaisen bittijonon Δ kanssa, saadaan uuden viestin CRC-tarkistussumma pysymään validina laskemalla Δ:lle tarkistussumma C', ja liittämällä $M \oplus \Delta$:n perään tarkistussumma $C \oplus C'$.

Erityisesti tässä tehtävässä voitaisiin muuttaa rahasumma 790 markasta 5000 markkaan laskemalla ensin tarvittava Δ:

$$\begin{aligned} 790 &= 00000000\ 00000000\ 00000011\ 00010110_2 \\ 5000 &= 00000000\ 00000000\ 00010011\ 10001000_2 \\ \Delta &= 00000000\ 00000000\ 00010000\ 10011110_2 \end{aligned}$$

laskemalla sen jälkeen Δ:lle tarkistussumma ja XORaamalla Δ tarkistussummineen paketin päälle.

Tehtävä 3

[1] Tulkitaan esim. CRC-polynomi `1101` polynomiksi x^4+x^3+x+1 ja viesti `11100` polynomiksi $x^5+x^4+x^3+1$. Tulkitaan polynomien jaollisuus kuten tavallista, mutta lasketaan laskut modulo 2: esimerkiksi

$$(x^2+x+1)(x^3+x) = x^5+2x^3+x^2+x \equiv x^5+x^2+x$$

$$
\begin{aligned}
4109^{3171} &\equiv 4109 \cdot 4109^{3170} \pmod{4897} \\
&\equiv 4109 \cdot 3922^{1585} \quad (3922 \equiv 4109^2) \\
&\equiv 4109 \cdot 3922 \cdot 3922^{1584} \\
&\equiv 4368 \cdot 3922^{1584} \quad (4109 \cdot 3922 \equiv 4368) \\
&\equiv 4368 \cdot 607^{792} \\
&\equiv 4368 \cdot 1174^{396} \\
&\equiv 4368 \cdot 2219^{198} \\
&\equiv 4368 \cdot 2476^{99} \\
&\equiv 4368 \cdot 2476 \cdot 2476^{98} \\
&\equiv 2592 \cdot 2476^{98} \\
&\equiv 2592 \cdot 4429^{49} \\
&\equiv 2592 \cdot 4429 \cdot 4429^{48} \\
&\equiv 1400 \cdot 4429^{48} \\
&\equiv 1400 \cdot 3556^{24} \\
&\equiv 1400 \cdot 1082^{12} \\
&\equiv 1400 \cdot 341^{6} \\
&\equiv 1400 \cdot 3650^{3} \\
&\equiv 1400 \cdot 3650 \cdot 3650^{2} \\
&\equiv 2429 \cdot 2660 \\
&\equiv 1997
\end{aligned}
$$

Tehtävä 4

a) Salaiset tekijät ovat $p = 29$ ja $q = 41$. Salainen eksponentti d toteuttaa

$$ed \equiv 1 \pmod{(p-1)(q-1)}$$

ts. d on e:n inverssi modulo $(p-1)(q-1)$. Nyt $e = 17$ ja d:n voi ratkaista Euklideen algoritmilla kuten aiemmin on tehty. Saadaan $e^{-1} \equiv 593 \pmod{(p-1)(q-1)}$, joten $d = 593$. Salainen allekirjoitusavain on siis (593, 1189).

b) Allekirjoitetaan sormenjälki:

$$S(321) = 321^{593} \bmod 1189 = 873$$

Sanoma kokonaisuudessaan on sitten

`Moi, mitä kuuluu?` 873

Tehtävä 5

Ensiksi yhden avaimen murtuminen murtaa välittömästi toisenkin. Toiseksi, saman viestin kryptaaminen kahdella eri eksponentilla mutta samalla modulilla paljastaa välittömästi viestin: oletetaan, että vihollinen tuntee x^e mod M:n ja $x^{e'}$ mod M:n samoin kuin julkiset eksponentit e ja e'. Vihollinen etsii ratkaisun yhtälölle

$$a \cdot e + b \cdot e' = 1$$

joka löytyy jos ja vain jos e ja e' ovat suhteellisia alkulukuja. Jos ratkaisu löytyy (useimmiten näin on, salauseksponentit ovat yleensä pieniä alkulukuja ja siten myös keskenään suhteellisia alkulukuja), paljastuu x:

$$(x^e)^a \cdot \left(x^{e'}\right)^b \equiv x^{ea+e'b} \equiv x^1 \equiv x \pmod{M}$$

On huomattava, että joko a tai b on negatiivinen. Negatiiviseen potenssiin $-x$ korottaminen täytyy tehdä laskemalla ensin kannan inverssi $\pmod{M}$ ja sen jälkeen korottamalla tämä inverssi potenssiin $|x|$.

Olisi mahdollista, että esim. x^e:llä ei olisi inverssiä modulo M, jos nimittäin x^e ja M jakaisivat yhteisiä tekijöitä. Tällöin yllä oleva hyökkäys ei onnistuisi, mutta sen sijaan hyökkääjä olisi onnistunut osumaan M:n tekijään, mikä tietenkin paljastaisi koko avaimen (todennäköisyys olematon)!

Tehtävä 6

(Mitä tapahtuu, jos viesti m on jomman kumman modulin tekijän monikerta?) Ei mitään; RSA toimii kuten normaalistikin, vaikka Eulerin lause ei tässä kohden sitä takaakaan.

$\boxed{\text{Laskuharjoitus 6}}$

Tehtävä 1

$$693^3 \equiv 2345 \pmod{4897}$$

joten allekirjoitus on aito.

Tehtävä 2

a) Protokolla ei takaa luottamuksellisuutta eikä aitoutta. Vihollinen voi tehdä man-in-the-middle -hyökkäyksen, koska sanomien autenttisuutta ei voida tarkastaa (A ei tiedä mitenkään, kuka lähetti hänen näkemänsä g^y:n; samoin toisin päin).

b) Viestin luottamuksellisuus säilyy, koska avaintenvaihto on salakirjoituksin varmennettu ja avainten lähettäjien henkilöllisyydet voidaan tarkastaa. Viestin M aitous ei säily, ainakaan, mikäli se ei sisällä lainkaan redundanssia. Jos M sisältää redundanssia, esim. sisäisen, kryptografisesti vahvan tarkistussumman, $E_K(M)$:n muutteleminen havaitaan, ja tällöin myös viestin aitous säilyy.

c) Protokolla ei takaa viestin M luottamuksellisuutta, mutta takaa sen aitouden. M ei säily luottamuksellisena, koska vihollisen tekemä man-in-the-middle -hyökkäys paljastuu vasta, kun B vastaanottaa A:n lähettämän viestin, ja välissä on vihollinen jo ehtinyt lukemaan M:n. A ei voi varmistua avaintenvaihdon autenttisuudesta ennen viestin lähettämistä. Aitous säilyy (B huomaa, jos $E_K(\ldots)$:a on muuteltu), koska se sisältää paljon redundanssia, nimittäin A:n allekirjoituksen istuntoavaimelle.

d) Viestin luottamuksellisuus säilyy, koska M:n lähettäjä A varmistuu B:n julkisen avaimen aitoudesta. Myös aitous säilyy, koska kolmannessa viestissä on kylliksi redundanssia.

e) Protokolla ei takaa mitään, koska vihollinen voi tehdä man-in-the-middle -hyökkäyksen olettaen, että B ei tunne A:n julkista avainta K_A^+ valmiiksi. Jos B ei tunne sitä, ei B voi millään varmistua siitä, kenen julkinen avain lähetettiin hänelle todellisuudessa. Protokolla on [ominaisuuksiltaan] hyvin samankaltainen kuin Diffie-Hellman -avaintenvaihto

Tehtävä 3

Olet itse Bob (B). Seuraavan viestin lähetät itse:

$$B \longrightarrow A \quad g^y \bmod n = 1848$$

Avainmateriaaliksi tulee

$$g^{xy} = X^y = 1872^{521} \bmod 2207 = 530$$

(tässä $X = g^x$ jollekin tuntemattomalle x).

$\boxed{\text{Laskuharjoitus 7}}$

Tehtävä 1

p on julkinen moduli, g julkinen generaattori ja x salainen eksponentti. Lasketaan

$$y = g^x \bmod p = 2^{1751} \bmod 2357 = 1185$$

Julkinen avain on $\langle y, g, p\rangle = \langle 1185, 2, 2357\rangle$ ja salainen avain x.

Allekirjoitetaan $h(M) = 1463$. Valitaan satunnainen k siten, että $\gcd(k, p-1) = 1$. Valitaan k:ksi 101 (joka on alkuluku, joten ehto varmasti täyttyy. k:n ei kuitenkaan ole pakko olla alkuluku).

Lasketaan

$$a = g^k \bmod p = 2^{101} \bmod 2357 = 1037$$

Ratkaistaan sitten

$$h(M) \equiv (xa + kb) \pmod{(p-1)}$$

b:n suhteen (muut arvot jo tunnetaan). Koska $\gcd(k, p-1) = 1$, on yhtälöllä aina ratkaisu:

$$\begin{aligned} h(M) &\equiv (xa + kb) \pmod{(p-1)} \Rightarrow \\ kb &\equiv h(M) - xa \pmod{(p-1)} \Rightarrow \\ 101 \cdot b &\equiv 1463 - 1815787 \equiv 2152 \pmod{2356} \Rightarrow \\ b &\equiv 2152 \cdot 101^{-1} \pmod{2356} \end{aligned}$$

Lasketaan 101:n inverssi 2356:n suhteen esimerkiksi Euklideen algoritmilla:

$$101^{-1} \equiv 1213 \pmod{2356}$$

Sijoittamalla tämä edelliseen yhtälöön, saadaan

$$b \equiv 2152 \cdot 1213 \equiv 2284 \pmod{2356}$$

Allekirjoitus on nyt pari

$$\langle a, b\rangle = \langle 1037, 2284\rangle$$

Tarkastetaan allekirjoitus $\langle a = 1037, b = 2284\rangle$: tutkitaan, onko $y^a a^b \equiv g^{h(M)} \pmod p$. Nyt $y = 1185$, $g = 2$, $p = 2357$, $a = 1037$, $b = 2284$ ja $h(M) = 1463$. Saadaan

$$y^a a^b \equiv 1185^{1037} 1037^{2284} \equiv 1072 \equiv 2^{1463} \equiv g^{h(M)} \pmod{2357}$$

Tarkastetaan vielä että

$$1 \leq a \leq p - 1 \Rightarrow 1 \leq 1037 \leq 2356$$

Koska näin on, on allekirjoitus aito.

Tehtävä 2

Olkoon n alkion joukosta S valittu osajoukko A siten, että $|A| = k$. Valitaan sitten S:stä toinen k:n alkion osajoukko B satunnaisesti. Lasketaan todennäköisyys sille, että $A \cap B = \emptyset$.

Oletetaan $k \ll n$ ja approksimoidaan osajoukon B valintaa toistokokeella, jossa valitaan satunnainen S:n alkio k kertaa. Todennäköisyys sille, että yksi täten valittu alkio ei kuulu A:han on $1 - \frac{k}{n}$, joten todennäköisyys sille, että $A \cap B = \emptyset$ on *tällä approksimaatiolla*

$$P_d \approx \left(1 - \frac{k}{n}\right)^k$$

Approksimoidaan eksponentin alla olevaa termiä (10):n avulla, ja saadaan

$$P_d \approx \left(e^{-\frac{k}{n}}\right)^k = e^{-\frac{k^2}{n}}$$

Ratkaistaan yhtälö $P_d = 0.5$:

$$\begin{aligned} e^{-\frac{k^2}{n}} &= 0.5 \Rightarrow \\ -\frac{k^2}{n} &= \ln 0.5 \Rightarrow \\ k^2 &= -\ln 0.5 \cdot n \Rightarrow \\ k &= \sqrt{\ln 2 \cdot n} \end{aligned}$$

Ts. k:n on oltava lähellä $\sqrt{n}$:ää.

Tehtävä 3

Oletetaan, että mikrotietokone kykenee suorittamaan 700000 DES-iteraatiota sekunnissa, ja että DESin avaimen asettamiseen kuluva aika on mitättömän pieni niin, että se voidaan jättää laskuissa huomiotta.

UNIX-koneiden tapa toimia salasanojen kanssa on seuraava: olkoon p jonkin käyttäjän salasana. Valitaan satunnainen jono bittejä, *suola* s, ja lasketaan hash-arvo $H(p, s)$. Tämä arvo, sekä käytetty suola, talletetaan tietokoneen `/etc/passwd`-tiedostoon.

Myöhemmin kun käyttäjä pyytää pääsyä järjestelmään, järjestelmä kysyy salasanaa. Käyttäjä vastaa p'. Järjestelmä hakee `/etc/passwd`-tiedostosta käyttäjää vastaavan rivin, jolta löytyy s ja $H(p, s)$. Sen jälkeen järjestelmä tarkastaa, onko $H(p', s) = H(p, s)$. Jos näin on, järjestelmä päättelee, että $p = p'$ ja sallii käyttäjälle pääsyn.

Standardi hash-funktio H, pitkälti historiallisista syistä, on, että salasanasta p poimitaan 56 bittiä, jotka asetetaan DES-algoritmin salausavaimeksi (käytännössä huomioidaan salasanan kahdeksan ensimmäistä kirjainta, joista jokaisesta ylintä bittiä ei käytetä. Tästä syystä salasanojen todellinen maksimipituus on kahdeksan merkkiä). Suolaa s käytetään modifioimaan DES-algoritmia jonkin verran, siten, että eri suolan arvoilla s on kyseessä aina erilainen DESin variaatio. Suolan s mukaista algoritmia käytetään p:stä johdetulla avaimella kryptaamaan nolla (0) 25 kertaa peräkkäin. Viimeisen kryptauskerran tulos on $H(p, s)$:n arvo.

Tehtävässä oletetaan, että on saatu 100000 riviä eri kohdekoneiden `/etc/passwd`-tiedostoista. Suoritetaan nyt joitakin laskelmia.

Olkoon P kaikkien mahdollisten salasanojen, S kaikkien mahdollisten suolojen ja C tunnettujen, kryptattujen salasanojen joukko. On murrettava vähintään yksi salasana C:stä, ts. löydettävä p siten, että $H(p, s) = X$ jollekin parille $(X, s) \in C$.

Tehdään hyökkäys siten, että kokeillaan järjestyksessä salasanoja $p_1, p_2, \ldots, p_{|P|} \in P$ sekä jokaiselle salasanalle suoloja $s_1, s_2, \ldots, s_{|S|} \in S$. Lasketaan jokaiselle parille $H(p_k, s_l)$ ja tarkistetaan, onko C:ssä paria (X, s) siten, että $H(p_k, s_l) = X$ ja $s_l = s$. Jos näin on, tämän parin kätkemä salasana murrettu.

Oletetaan, että taulukon C tutkiminen ei vie merkittävää määrää aikaa, vaan että kaikki kuluva aika kuluu $H(p_k, s_l)$:n laskemiseen.

C:ssä on keskimäärin $\frac{|C|}{|S|}$ samalla suolalla salattua salasanaa. Todennäköisyys sille, että tietty pari p_k, s_l sopii vähintään yhteen salasanaan C:ssä on

$$1 - \left(1 - \frac{1}{|W|}\right)^{\frac{|C|}{|S|}}$$

Jos $|W| \gg |C|$, on n:n yrityksen jälkeen todennäköisyys sille, että vähintään yksi salasana on murtunut

$$1-\left(1-\frac{1}{|W|}\right)^{n\frac{|C|}{|S|}}$$

Asettamalla tämä $\frac{1}{2}$:een ja ratkaisemalla n:n suhteen saadaan

$$\begin{aligned} 1-\left(1-\frac{1}{|W|}\right)^{n\frac{|C|}{|S|}} &= \frac{1}{2} \Leftrightarrow \\ \left(1-\frac{1}{|W|}\right)^{n\frac{|C|}{|S|}} &= \frac{1}{2} \Leftrightarrow \\ n\frac{|C|}{|S|}\ln\left(1-\frac{1}{|W|}\right) &= -\ln 2 \Leftrightarrow \\ n &= \frac{(-\ln 2)|S|}{|C|\ln\left(1-\frac{1}{|W|}\right)} \qquad (13) \end{aligned}$$

Tietokone laskee 700000 DES-iteraatiota sekunnissa, joten salasanayritteitä voidaan kokeilla sekunnissa $\frac{700000}{25}$.

a) Suolaa ei käytetä; sama tilanne kuin että suola olisi aina vakio, joten $|S| = 1$. Mahdollisia salasanoja on $|W| = 26^8$. Tässä ja kaikissa muissa kohdissa on $|C| = 10^5$. Sijoittamalla yhtälöön (13) saadaan

$$n = \frac{(-\ln 2)1}{10^5 \ln\left(1-\frac{1}{26^8}\right)}$$

josta $n = 1447478$ ja hyökkäykseen kuluva aika $t = n \cdot \frac{25}{700000} = 52\ s$.

b) $|S| = 1$, $|W| = 8^{128}$, $n = 499465181453$, $t = 7\ kk$

c) $|S| = 2^{12}$, $|W| = 8^{128}$, $n = 2045809383232424$, $t = 2317\ a$

d) UNIXin salasanajärjestelmässä ei suola on 12-bittinen, mutta tässä oletetaan, että se voi olla suurempi. Samoin oletetaan, että kaikki 10 merkkiä vaikuttavat kryptaukseen, vaikka kymmenen seitsemänbittisen merkin informaatiosisältö on 2^{70} joka on suurempi kuin DES-avaimen koko 2^{56}.

Idea on se, että jokaisella käyttäjällä on suola, joka on varmasti eri kuin kenelläkään muulla maailmassa. Tämä aiheuttaa, että $|S| = |C|$ (ei kannata kokeilla sellaisilla suolan arvoilla, jotka eivät esiinny C:ssä kertaakaan), ja (13) sievenee muotoon

$$n = \frac{(-\ln 2)}{\ln\left(1-\frac{1}{|W|}\right)}$$

Huomataan, että kyseessä on tavallinen brute-force -hyökkäys, eikä esimerkiksi saadun aineiston C koko $|C|$ vaikuta tulokseen enää mitenkään. $|W| = 2^{70}$, $n = 818323753292969962226$, $t = 9,26 \cdot 10^8\ a$.

Tehtävä 4

(Kts. Schneier, s. 102) Victor ja Peggy kävelevät molemmat pisteeseen B. Sen jälkeen Peggy kulkee salaisen oven kautta ja palaa pisteeseen B. Nyt Victor uskoo, että Peggy tuntee taikasanan, mutta ei silti tiedä sanaa (olettaen, ettei Victor kuule, mitä Peggy puhuu oven luona).

Mikä on ero alkuperäiseen todistukseen? Oletetaan, että Victorilla on ystävä Alice, ja Alice on tarkkailemassa Victorin luona Peggyn ja Victorin toimintaa. Alkuperäissä todistuksessa Alice ei voi tietää, huijaavatko Victor ja Peggy häntä vaiko eivät, koska ei voi tietää, ovatko Victor ja Peggy sopineet etukäteen, missä järjestyksessä Victor käskee Peggyä tulemaan ulos luolan eri puolilta. Sen sijaan yllä esitetty vaihtoehto todistaa myös Alicelle Peggyn tietämyksen, eikä ainoastaan Victorille.

Tehtävä 5

a) Vanha A:n viesti voidaan soittaa uudelleen.

b) Ei hyökkäyksiä, A:n viestit hylätään vanhentuneina.

c) Ei hyökkäyksiä, A:n viestit hylätään vanhentuneina.

d) A:n lähettämät viestit voidaan nauhoittaa. B ei hyväksy niitä nykyhetkellä, koska aikaleima on edellä, mutta ne voidaan soittaa vihollisen toimesta myöhemmin B:lle, kun B:n kello on saavuttanut ajan jota A:n kello näyttää nyt.

Tehtävä 6

Koska kyseessä on nollatietotodistus, ei sitä voi esittää ei-interaktiivisesti. Todistuksen periaate esitetään kuitenkin tässä. Tehtävä perustuu oletukseen, että kahden graafin välisen isomorfismin löytäminen tuntematta sitä etukäteen on hankalaa suurilla graafeilla.

Tehtävässä on kaksi graafia, G_1 ja G_2, joiden välillä vallitsee jokin isomorfismi $\mathcal{P}_{12}$, s.e. $\mathcal{P}_{12}(G_1) = G_2$ ja $\mathcal{P}_{12}^{-1}(G_2) = G_1$. Graafien välisen isomorfismin tunteva assari haluaa nyt osoittaa, että tuntee tämän isomorfismin, mutta ei halua paljastaa sitä. $\mathcal{P}_{12}$:n paljastaminen sinällään ei ole siis kelvollinen ratkaisu.

Assari luo sen sijaan kolmannen graafin G_3, joka on isomorfinen G_1:n kanssa. Koska assari on luonut graafin itse, hän tuntee isomorfismin $\mathcal{P}_{13}$ (jolle $\mathcal{P}_{13}(G_1) = G_3$). Tuntiessaan $\mathcal{P}_{12}$:n ja $\mathcal{P}_{13}$:n kykenee hän laskemaan myös isomorfismin G_2:lta G_3:lle, nimittäin

$$\mathcal{P}_{23} = \mathcal{P}_{13} \circ \mathcal{P}_{12}^{-1}$$

Assari kysyy sen jälkeen, haluaako yleisö, että hän osoittaa G_3:n isomorfiseksi G_1:n vaiko G_2:n kanssa. Kumman tahansa yleisö valitseekin, kykenee assari osoittamaan isomorfismin, paljastamalla joko $\mathcal{P}_{13}$:n tai $\mathcal{P}_{23}$:n. Jomman kumman isomorfismin paljastaminen ei kuitenkaan kerro mitään isomorfismista $\mathcal{P}_{12}$.

Huijari voisi luoda G_3:n, joka olisi isomorfinen molempien graafien kanssa (jos se on isomorfinen toisen kanssa, on se myös toisen), mutta ei kykenisi *osoittamaan* isomorfismia kuin jomman kumman alkuperäisistä graafeista kanssa. Todennäköisyys huijauksen onnistumiseen yhdellä kerralla on $\frac{1}{2}$ ja n:llä kerralla $\frac{1}{2^n}$, joten suurilla n paljastuu huijaus käytännössä aina.

$\boxed{\text{Laskuharjoitus 8}}$

Tehtävä 1

a) PKCS #1:ssä jokaisen sanoman eteen lisätään mahdollisimman paljon satunnaista dataa. Täten, vaikka alkuperäinen viesti olisi kuinka lyhyt, on kuitenkin potenssiin korotettu data maksimaalisen pituinen, ja tällöin potenssiin korotettuna se numeroksi tulkittuna ylittää reilusti n:n.

b) Sama satunnaisen datan lisääminen korjaa myös tämän vian.

c) Ja tämän.

d) Sanomaa allekirjoitettaessa salaisella RSA-avaimella allekirjoitetaan viestin hash-arvo eikä viestiä itseään, joten tehtävässä mainittu hyökkäys vaatisi törmäyksen löytämistä hash-funktiosta.

Tehtävä 2

Esimerkiksi salaamalla kaikki liikenne normaalisti vaihtuvilla symmetrisillä avaimilla, mutta liittämällä jokaisen uuden yhteyden alkuun korkean tahon julkisella avaimella salattuna yhteydessä käytetty istuntoavain. Tällöin, halutessaan, voi korkean tahon edustaja dekryptata salatun istuntoavaimen ja lukea siirretyn datan, mutta muut eivät kykene tähän. Esimerkki (A siirtää B:lle salaisen sanoman M):

$$A \longrightarrow B \quad E_{K_{SES}}(M), E_B(K_{SES}), E_{VALVOJA}(K_{SES})$$

Tehtävä 3

Oletetaan, että A ja B haluavat keskustella keskenään ja saavuttaa pfs:n.

Ensin A generoi satunnaisen julkisen avaimen K_R ja lähettää sen B:lle:

$$A \longrightarrow B \quad K_R$$

B generoi istuntoavaimen K_{SES}, ja salaa sen ensin Alicen lähettämällä satunnaisella julkisella avaimella, ja sen jälkeen Alicen pysyvällä julkisella avaimella man-in-the-middle -hyökkäyksen välttämiseksi:

$$B \longrightarrow A \quad K_A(K_R(K_{SES}))$$

A dekryptaa viestin ensin pysyvällä salaisella avaimellaan ja sen jälkeen K_R:ää vastaavalla salaisella avaimella. Sen jälkeen Alice hävittää maailmankaikkeudesta K_R:n ja sitä vastaavan salaisen avaimen. Nyt vaikka myöhemmin sekä Alicen että Bobin julkiset avaimet paljastuisivat, säilyisi K_{SES} salaisena, koska se on siirretty nyttemmin hävitetyn K_R:n suojaamana.

Yhdensuuntaisessa viestinvälityksessä ei pfs:a voi saavuttaa.

Sähköpostissa ei pfs ole siksi mahdollista, mutta sen voisi jossain määrin saavuttaa siten, että vastaanottajalla olisi kaksi avainparia: vaihtuva ja pysyvä. Hän vaihtaisi vaihtuvaa julkista avaintaan säännöllisin väliajoin ja sertifioisi ne pysyvällä avaimella. PGP:ssä avainten vaihtaminen ja peruminen on hankalaa.

Pfs ei sovi avainten varmuuskopioinnin tai key recoveryn kanssa tietenkään yhteen, koska jos jotain tallennetaan, jolla viestiin päästään myöhemmin käsiksi, ei kyseessä ole pfs.

Tehtävä 4

Aivan samoin.

Tehtävä 5

a) Liitetään allekirjoitukseen aikaleima (jolloin vaaditaan synkronoituja kelloja) tai jotain tietoa, jota ei ole ollut olemassa kuin vasta äskettäin (esimerkiksi äsken generoitu nonssi). Kellonaika vakuuttaa kaikki, jos kellot ovat luotettavia. Nonssi vakuuttaa vain sen generoijat ja generoijiin luottavat tahot.

b) Julkaistaan ajanhetkellä T dokumentin sormenjälki sanomalehdessä, ja todetaan myöhemmin, että dokumentin sormenjälki vastaa julkaistua; tai pyydetään yleisesti luotettua tahoa allekirjoittamaan dokumentti ja sen hetkinen aikaleima, ja esitetään myöhemmin allekirjoitus todistuksena siitä, että dokumentti oli jo olemassa.

c) Ei mitenkään, ellei dokumentti sisällä jotain sellaista informaatiota, jota ei voinut ollut olemassa vielä hetkellä T, esimeriksi seuraavan vuoden uutisia tai jotain nonssia, joka on generoitu vasta myöhemmin, minkä todistaminen tietenkin on yhtä vaikeaa.

d) Itse allekirjoitus allekirjoitetaan jonkin luotetun tahon toimesta aikaleiman kanssa.

e) Salataan salaisuus salaisella avaimella K, ja julkaistaan K esimerkiksi sanomalehdessä ajanhetkellä T. Tai (hauska): salataan salaisuus salaisella avaimella K ja hävitetään kaikki tieto avaimesta. Aletaan murtamaan avainta brute forcella (oletetaan, että parempaa menetelmää ei tunneta). Avain paljastuu, kun avain on murrettu, ja tähän kuluva aika voidaan *jossain määrin* arvioida.

Tehtävä 6

a) Salaisen avaimen kanssa ekvivalentti (samoihin allekirjoituksin pystyvä) avain voidaan löytää brute forcella (miksi?). Myös eri viestien allekirjoitukset ovat usein samoja, joten saman allekirjoituksen voi siirtää laillisesta dokumentista laittomaan.

b) Yhdestä allekirjoituksesta $\langle a, b\rangle$ sormenjäljelle $h(M)$ voidaan ratkaista salainen eksponentti x ratkaisemalla

$$ax + kb - h(M) \equiv 0 \pmod{(p-1)}$$

jossa vain x on tuntematon.

c) Kahdesta allekirjoituksesta $\langle a, b\rangle$ ja $\langle a, b'\rangle$ (a riippuu vain k:sta, joten jos k pysyy samana pysyy a myös) viesteille $h(M)$ ja $h(M')$ voidaan ratkaista salainen eksponentti x:

$$\begin{cases} ax + kb - h(M) \equiv 0 \pmod{(p-1)} \\ ax + kb' - h(M') \equiv 0 \pmod{(p-1)} \end{cases} \Rightarrow k(b-b') + (h(M') - h(M)) \equiv 0 \pmod{(p-1)}$$

Oikeanpuoleinen yhtälö ratkaistaan ensin k:n suhteen ja sen jälkeen x ratkaistaan kuten edellä.

d) Satunnaisia ElGamal-allekirjoituksia pystyy generoimaan kuka tahansa, ts. generoimaan jonkin allekirjoituksen $\langle a, b\rangle$ ja sen jälkeen katsomaan, minkälaiseen sormenjälkeen se sopisi. Jos viestistä ei lasketa yksisuuntaista funktiota, on sormenjälki viesti itse, joten voidaan generoida satunnainen allekirjoitus, ja julkaista se siihen liittyvän, satunnaisen (roska)viestin kanssa muka aidosti allekirjoitettuna.

Satunnaisia allekirjoituksia generoidaan näin:

Valitaan satunnaiset u ja v, ja lasketaan

$$\begin{aligned} a &= y^u g^v \bmod p \\ b &= -au^{-1} \bmod (p-1) \\ M &= vb \bmod (p-1) \end{aligned}$$

jolloin $\langle a, b\rangle$ on M:n validi allekirjoitus, joka voidaan tarkastaa julkisella avaimella y:

$$\begin{aligned} y^a a^b &\equiv y^a \left(y^u g^v\right)^b \pmod p \\ &\equiv y^a y^{ub} g^{vb} \pmod p \\ &\equiv y^a y^{u(-au^{-1})} g^{vb} \pmod p \\ &\equiv y^a y^{-a} g^{vb} \pmod p \\ &\equiv g^{vb} \equiv g^M \pmod p \end{aligned}$$

e) Jos a saa olla suurempi kuin $p-1$, voi valideja allekirjoituksia tehdä kuka tahansa. Tässä esitetään eräs tapa: asetetaan

$$a = (g^{-1} \bmod p)((p-1)g)^2$$

jolloin

$$y^a \equiv \left(y^{(p-1)}\right)^{(g^{-1} \bmod p) g^2 (p-1)} \equiv 1^{(\dots)} \pmod p$$

Toisaalta taas

$$a \equiv (-g)^2 (g^{-1}) \equiv g \pmod p$$

koska $p-1 \equiv -1 \pmod p$. Nyt sormenjälki $h(M)$ voidaan allekirjoittaa asettamalla $b = h(M)$:

$$y^a a^b \equiv 1 \cdot g^{h(M)} \pmod p$$

kuten tulikin olla.

$\boxed{\text{Laskuharjoitus 9}}$

Tehtävä 1

k-kombinaatioita n:stä alkiosta on $C(n,k) = \frac{n!}{k!(n-k)!}$. Todennäköisyys siihen, että kaksi kombinaatiota ovat samat, on

$$\frac{1}{C(n,k)}$$

Todennäköisyys on pienin, kun $k = n/2$, koska tällöin kombinaatioita on eniten.

Olkoon siis $k = n/2$. Tällöin todennäköisyys on

$$\frac{1}{C(2k,k)} = C(2k,k)^{-1} = \frac{(k!)^2}{(2k)!}$$

Kirjoitetaan tämä (2):n avulla muotoon

$$\frac{\left[\sqrt{2\pi k}\,\left(\frac{k}{e}\right)^k\right]^2}{\sqrt{4\pi k}\,\left(\frac{2k}{e}\right)^{2k}} = \frac{2\pi k}{2\sqrt{\pi k}} \cdot (1/2)^{2k} = (1/2)^{2k}\sqrt{\pi k}$$

ja etsitään pienin k, jolle tämä lauseke $< 2^{-64}$. Ensimmäinen tällainen k on 34, jolloin $n = 2k = 68$.

Kaksi n bitin jonoa ovat samoja todennäköisyydellä 2^{-n}.

Tehtävä 2

Väärinkäytöstä on kahdenlaista: huijausta setelin luomisvaiheessa, ja setelin moninkertaisen käytön yritystä. Jos luomisvaiheessa haluaa kyetä huijaamaan, on luoduista lukupareista täsmälleen puolen oltava aitoja ja puolen väärennettyjä, ja pankin olisi valittava *cut-and-choose* -menettelyssä näistä pareista kaikki aidot mutta ei yhtään väärennettyä. Edellisen tehtävän perusteella kun pareja on 68, alittaa tämän tapahtuman todennäköisyys 2^{-64}:n. Näistä pareista paljastetaan menettelyssä puolet, eli $k = 34$.

Seteliä kahteen kertaan käytettäessä on maksun vastaanottajan valittava toisella käyttökerralla kaikista k parista samat puoliskot kuin ensimmäisellä kerralla. Tämän todennäköisyys on edellisen tehtävän nojalla 2^{-k}, joten tämän väärinkäytöksen estämiseksi olisi oltava $k = 64$, jolloin alunperin pareja tehtäisiin 128 (68 ei riittäisi).

Tehtävä 3

a) $M = 4321$. Valitaan alkuluku $p = 5939$, ja rakennetaan kvadraattinen polynomi modulo p: (LaGrangen kynnysrakenteen voisi tehdä myös vaikka reaalisella, ei-diskreetillä polynomilla, mutta yleensä diskreettejä polynomeja käytetään)

$$\boxed{ax^2 + bx + M \pmod p}$$

Valitaan a:lle ja b:lle satunnaiset arvot $a = 13$ ja $b = 4$, ja evaluoidaan sitten polynomi viidessä pisteessä:

$$\begin{aligned}P(1) &= 4338\\ P(2) &= 4381\\ P(3) &= 4450\\ P(4) &= 4545\\ P(5) &= 4666\end{aligned}$$

Varjot ovat nyt $(1, 4338)$, $(2, 4381)$, $(3, 4450)$, $(4, 4545)$ ja $(5, 4666)$.

b) Tiedetään, että polynomi on kvadraattinen kuten edellä (koska kynnysrakenne on (3, 5)-rakenne). Muodostetaan kolmen tunnetun varjon perusteella yhtälöryhmä

$$\begin{cases} a\cdot 2^2 + b\cdot 2 + M \equiv 19535 \pmod{65537}\\ a\cdot 4^2 + b\cdot 4 + M \equiv 1583 \pmod{65537}\\ a\cdot 5^2 + b\cdot 5 + M \equiv 44146 \pmod{65537}\end{cases}$$

ja ratkaistaan se:

$$\begin{aligned} 9a + b &\equiv 42563 \pmod{65537}\\ 12a + 2b &\equiv 47585 \pmod{65537}\\ 6a &\equiv 37541 \pmod{65537}\\ a &\equiv 6^{-1}\cdot 37541 \pmod{65537}\end{aligned}$$

Nyt on ratkaistava $6^{-1} \pmod{65537}$. Saadaan 10923, joten

$$a \equiv 60871$$

josta

$$b \equiv 42563 - 9\cdot 60871 \equiv 19020 \pmod{65537}$$

ja

$$M \equiv 19535 - 2\cdot 19020 - 4\cdot 60871 \equiv 159$$

Tehtävä 4

a) Muodostetaan yhtälöryhmä varjojen ja tunnetun modulin perusteella:

$$\begin{cases} s' \equiv (a_1 = 64) \pmod{(m_1 = 1013)}\\ s' \equiv (a_2 = 421) \pmod{(m_2 = 1019)}\\ s' \equiv (a_3 = 840) \pmod{(m_3 = 1021)}\end{cases}$$

Tämä yhtälöryhmä on nyt ratkaistava. Yhtälöryhmän voi ratkaista eri tavoin, mutta tässä esitetty on suoraviivainen ja elegantti.
Lasketaan $M = \prod_{i=1}^{3} m_i$ sekä $M_i = \frac{M}{m_i}$. Saadaan:

$$\begin{aligned} M &= 1053924187\\ M_1 &= 1040399\\ M_2 &= 1034273\\ M_3 &= 1032247\end{aligned}$$

Lasketaan sitten $M_i^{-1} \pmod{m_i}$ kaikille M_i:

$$\begin{aligned} M_1^{-1} &\equiv 401 \pmod{m_1}\\ M_2^{-1} &\equiv 934 \pmod{m_2}\\ M_3^{-1} &\equiv 702 \pmod{m_3}\end{aligned}$$

Todetaan, että

$$s' \equiv \left(\sum_{i=1}^{3} a_i M_i (M_i^{-1} \pmod{m_i})\right) \pmod M$$

on eräs ratkaisu yhtälöryhmälle: k:s yhteenlaskettava modulo m_n on a_k, jos $k = n$, muutoin nolla: jos $k = n$, on

$$a_k M_k (M_k^{-1} \pmod{m_k}) \equiv a_k \pmod{m_k}$$

koska M_k on kerrottu sen inverssillä. Toisaalta, jos $k \neq n$, on m_n M_k:n tekijä M_k:n määritelmän perusteella, joten yhteenlaskettava sisältää nollan modulo m_n ja on siten kokonaisuudessaan nolla.

Laskemalla saadaan $s' = 809737562$. Todetaan, että $s' < M$ (voisi olla ehkä, että $s' \geq M$. Tällöin s' olisi vielä redusoitava modulo M).

Lasketaan alkuperäinen viesti: $s = (s' \bmod p) = 159$.

b) Lisätään rakenteeseen jokin varjo. Valitaan seuraavan varjon moduliksi 1031 (ensimmäinen alkuluku 1021:n jälkeen): $m = 1031$. Lasketaan

$$s' \bmod m = 809737562 \bmod 1031 = 472$$

joten varjo on $(1031, 472)$.

Tehtävä 5

a) Kynnysrakenteen voi purkaa $m - 1$ tai vähemmällä varjolla, erityisesti jos tunnetaan varjoista ne, joilla on suurimmat modulit.

b) Koska kynnysrakenteen voi ratkaista m pienimmällä varjolla, on $s' < d_1 \cdot d_2 \cdots d_m = M$. Oletetaan, että tunnetaan $m - 1$ suurinta varjoa. Niiden tulo on $N = d_{n-m+2} \cdot d_{n-m+3} \cdots d_n$. Näistä varjoista voidaan nyt ratkaista yksikäsitteisesti $s' \pmod N$. Koska $s' < M$, niin kun $s' \pmod N$ tunnetaan, on eri s':n mahdollisia arvoja $\frac{M}{N}$. Jos on $M > Np$ (kuten vaaditaan), seuraa tästä että s':n arvoilla on p eri mahdollisuutta, ts. yhtä monta kuin eri viestejä s, koska $s < p$, jolloin $m - 1$:n varjon tunteminen ei paljasta mitään salatusta viestistä s. Mutta jos $M < Np$, kertovat $m - 1$ suurinta modulia jotakin viestistä s. Esim. jos $M = 2N$, saadaan $m - 1$ suurimman modulin avulla kaksi eri vaihtoehtoa s':n arvolle, ja toinen näistä vaihtoehdoista on oikea s':n arvo, josta s on laskettavissa kunhan oikea vaihtoehto arvataan (molemmat voidaan kokeilla, ja valita se, joka näyttää tuottavan "järkevän" s:n arvon).

c) Sekoituskertoimen on oltava väliltä $1 \leq r < \frac{M}{p}$, koska on oltava $s' < M$.

Tehtävä 6

Bob voi toistaa Alicen lähettämän paketin ja siten saada aina halutessaan lantinheiton tulokseksi nollabitin.

Keinoja vian korjaamiseksi:

- Bob lähettää oman bittinsä b' selväkielisenä (sen salaamisesta ei ole tässä mitään hyötyä), tai
- hash-arvoihin lasketaan mukaan satunnaislukujen ja bittien lisäksi lähettäjien nimet.

Laskuharjoitus 10

Tehtävä 1

Yhteisen modulin käytöstä ei ole haittaa. Voisi ajatella, että koska molemmat tuntevat modulin faktoroinnin, osapuolet voisivat murtaa toistensa avaimet. Näin ei ole kuitenkaan, koska myös "julkiset eksponentit" e pysyvät protokollassa salaisina tarpeeksi pitkään. Molemmat osapuolet generoivat itse avainparinsa (e, d), jossa nyt myös siis e:n tulee olla salainen eikä arvattavissa (standardieksponentteja 3, 17, jne. ei siis tule käyttää).

Tehtävä 2

a) A:n ja B:n on tunnettava modulin M tekijöihinjako $M = pq$, jotta molemmat osapuolet voivat generoida itselleen eksponenttiparin kuten edellisessä tehtävässä. Protokolla on (tässä K on A:n generoima istuntoavain, (e, d) A:n eksponenttipari ja (e', d') B:n eksponenttipari):

$$\begin{array}{ll} A \longrightarrow B & K^e \bmod M \\ B \longrightarrow A & (K^e)^{e'} \bmod M = K^{ee'} \bmod M \\ A \longrightarrow B & K^{e'} \bmod M \end{array}$$

Jos M:n faktorointi on *julkista* tietoa, protokollaa vastaan voi tehdä helposti man-in-the-middle -hyökkäyksen. Tämä johtuu siitä, ettei A kykene erottamaan B:tä vihollisesta mitenkään, koska vihollinen tietää täsmälleen kaiken minkä B:kin. Jos taas M:n faktoroinnin tuntevat vain A ja B, man-in-the-middle -hyökkäys ei onnistu yhtä helposti, koska vihollinen ei kykene generoimaan itselleen omaa avainparia, ja A voi autentikoida B:n sillä perusteella, että B kykenee tekemään avainparin modulille M.

Tällöinkin voi vihollinen hyökätä varomatonta toteutusta vastaan käyttämällä itse eksponenttia yksi:

$$\begin{array}{ll} A \longrightarrow I(B) & K^e \bmod M \\ I(B) \longrightarrow A & (K^e)^1 \bmod M = K^e \bmod M \\ A \longrightarrow I(B) & K^1 \bmod M = K \bmod M \end{array}$$

Tässä vihollinen I on onnistunut käymään protokollan lävitse, ja A voisi (virheellisesti) päätellä, että I on tuntenut modulin faktoroinnin ja olisi siis B. I käytti eksponenttiparinaan $(1, 1)$:tä, jolle pätee triviaalista $x^{ed} \equiv x^1 \equiv x \pmod{M}$.

b) Ei toimi. Protokolla on esitetty alla, missä K on A:n generoima istuntoavain ja K_A ja K_B ovat A:n ja B:n bittivirrat:

$$\begin{array}{ll} A \longrightarrow B & K \oplus K_A \\ B \longrightarrow A & K \oplus K_A \oplus K_B \\ A \longrightarrow B & K \oplus K_B \end{array}$$

Nämä viestit nähtyään voi vihollinen laskea helposti K:n XORaamalla kaikki kolme viestiä:

$$\begin{aligned} (K \oplus K_A) \oplus (K \oplus K_A \oplus K_B) \oplus (K \oplus K_B) &= (K \oplus K \oplus K) \oplus (K_A \oplus K_A) \oplus (K_B \oplus K_B) \\ &= K \oplus 0 \oplus 0 \oplus 0 \\ &= K \end{aligned}$$

Joten istuntoavain ei ole pysynyt salaisena.

c) Tämä vastaa täysin a)-kohtaa, jos a)-kohdassa modulin faktorointi on julkisesti tiedossa (koska tällöin molemmissa tapauksissa kaikki tuntevat modulille m $\varphi(m)$:n). Protokolla muistuttaa hyvin paljon Diffie-Hellmania, ja sanomat tuleekin allekirjoittaa kuten DH:ssa.

Tehtävä 3

Ratkaistaan tehtävä suoraan n henkilölle, joilla on nimet $A_1, \ldots, A_n$ ja jotka haluavat saada nopanheiton tulokseksi jonkin luvun väliltä $0 \ldots k-1$. ("Perinteinen" nopanheitto saadaan valitsemalla $k = 6$ ja lisäämällä lopputulokseen ykkönen.)

1. Jokainen henkilö A_i valitsee luvun N_i väliltä $0 \ldots k-1$
2. Jokainen generoi (suuren) satunnaisluvun R_i.
3. Jokainen julkaisee viestin $HASH(A_i, N_i, R_i)$. Tämä hash-arvo sitoo yhteen henkilön nimen ja valitut kaksi numeroa, silti paljastamatta niitä.
4. Kun kaikki ovat julkaisseet hash-arvot, N_i:t ja R_i:t paljastetaan.
5. Osapuolet tarkastavat yhdessä, että kukaan ei huijannut hash-arvoja laskiessaan.
6. Nopanheiton tulos on $\sum_{i=1}^{n} R_i \bmod k$.

Tämä muistuttaa suuresti laskuharjoituksen 9 lantinheittoprotokollaa.

Tehtävät 4, 5 ja 6

a) Protokollassa Alice välittää Bobille itse generoimansa istuntoavaimen luotetun osapuolen, Trentin, kautta:

$$\begin{array}{ll} A \longrightarrow T & A, E_A(T_A, B, K_{SES}) \\ T \longrightarrow B & E_B(T_T, A, K_{SES}) \end{array}$$

Tämän protokollan vika on se, että sekä ykkös- että kakkosviestissä kulkevat salatut osat ovat täsmälleen samanrakenteiset. Tästä seuraa, että vihollinen voi soittaa uudelleen esimerkiksi kakkosviestin $E_B(\dots)$:n ykkösviestin salatun osan paikalla. Tätä sopivasti tekemällä voi vihollinen luoda jatkuvasti uusia yhteyksiä A:n ja B:n välillä ja pitää istuntoavainta elossa määrättömän pitkään.

Protokolla korjataan helposti laittamalla molempien viestien sisään eksplisiittinen tyyppikenttä, joka kertoo, onko viesti tarkoitettu Trentille vai Trentiltä poispäin:

$$\begin{array}{ll} A \longrightarrow T & A, E_A(init, T_A, B, K_{SES}) \\ T \longrightarrow B & E_B(resp, T_T, A, K_{SES}) \end{array}$$

Alice ei autentikoi Bobia eksplisiittisesti, koska Alice ei saa koko protokolla-ajon aikana yhtään viestiä eikä voi siis sanoa mitään Bobin olemassaolosta. Sen sijaan Bob autentikoi korjatussa versiossa Alicen eksplisiittisesti, koska voi päätellä näin:

- Sain Trentiltä viestin, jossa Trent sanoo, että Alice ottaa minuun yhteyttä.
- Trent on siis saanut äskeittäin Alicelta protokollan ensimmäisen viestin, jonka Alice on lähettänyt äskettäin (tämä perustuu aikaleimoihin).
- Tästä syystä juuri Alice on äskeittäin lähettänyt tämän istuntoavaimen ja on ollut äskettäin olemassa.

Tämä on halpa protokolla: yhteyksiä otetaan vain kaksi ja ne ovat melko lyhyitä. Offline -laskentaa ei voi suorittaa järkevästi. Luotetun osapuolen tehtävä on suhteellisen kevyt, koska se joutuu suorittamaan vain yhden dekryptauksen ja yhden kryptauksen symmetrisellä avaimella aikaleiman muodostamisen sekä tarkistuksien lisäksi. Protokolla perustuu kellojen käyttöön. Protokollan ainoa satunnaisluku on K_{SES}, jonka Alice generoi. On huomattava, että Trent joutuu ottamaan yhteyttä Bobiin, mikä voi olla hankalaa! (Muissa tehtävän protokollissa Trent keskustelee vain sellaisten osapuolten kanssa, jotka ottivat yhteyttä ensin Trentiin.)

b) Salaisen avaimen Needham-Schröderissä Alicen on tarkoitus luoda yhteys Bobiin luotetun osapuolen, Trentin, avulla. Protokolla menee näin:

$$\begin{array}{ll} A \longrightarrow T & A, B, N_A \\ T \longrightarrow A & E_A(N_A, B, K_{SES}, E_B(K_{SES}, A)) \\ A \longrightarrow B & E_B(K_{SES}, A) \\ B \longrightarrow A & E_{K_{SES}}(N_B) \\ A \longrightarrow B & E_{K_{SES}}(N_B + 1) \end{array}$$

Protokollan vika on siinä, että kolmannen viestin, jossa Bob vastaanottaa istuntoavaimen, tuoreutta ei voi tarkistaa. Vihollinen voi nauhoittaa tällaisen viestin, ja sen jälkeen ajan kanssa murtaa istuntoavaimen, joka siinä siirrettiin. Myöhemmin vihollinen voi sitten esiintyä Bobille Alicena lähettämällä tämän nauhoitetun viestin, joka sisältää nyt istuntoavaimen, jonka vihollinen tuntee. Näin vihollinen voi esiintyä Bobille Alicena:

$$\begin{array}{ll} I(A) \longrightarrow B & E_B(K_{SES}, A) \\ B \longrightarrow I(A) & E_{K_{SES}}(N_B) \\ I(A) \longrightarrow B & E_{K_{SES}}(N_B + 1) \end{array}$$

Koska tämä on kaikki, minkä Bob näkee, uskoo Bob nyt puhuvansa Alicen kanssa, vaikka todellisuudessa toisessa päässä onkin vihollinen.

Protokollan voisi korjata monin eri tavoin. Suoraviivaisin, joskaan ei välttämättä paras, tapa on lisätä kolmanteen viestiin aikaleima:

$$\begin{array}{ll} A \longrightarrow T & A, B, N_A \\ T \longrightarrow A & E_A(N_A, B, K_{SES}, E_B(T, K_{SES}, A)) \\ A \longrightarrow B & E_B(T, K_{SES}, A) \\ B \longrightarrow A & E_{K_{SES}}(N_B) \\ A \longrightarrow B & E_{K_{SES}}(N_B + 1) \end{array}$$

Tämä sitoo viestin nykyisyyteen, eikä sitä voi soittaa jälkikäteen uudelleen.

Korjatussa protokollassa Bob autentikoi Alicen eksplisiittisesti, ja päättelee näin:

- Sain äskettäin aikaleimatun viestin, joka on salattu minun ja Trentin välisellä avaimella, ja joka sisältää istuntoavaimen ja Alicen nimen.

- Tämän viestin Trent on lähettänyt Alicen avaimella salattuna, joten, koska saan sen, on Alice lukenut Trentin lähettämän viestin mutta kukaan muu ei (Bob olettaa, ettei Alice paljasta tietoa tarkoituksellisesti ulkopuolisille).
- Siksi vain minä ja Alice tunnemme istuntoavaimen.
- Joku vastasi nonssihaasteeseen. Koska vastaajan on täytynyt tuntea istuntoavain, on se ollut Alice.

Alice *saattaa voida* autentikoida Bobin eksplisiittisesti, mutta tämä riippuu toteutuksesta. *Jos* neljännen viestin ($E_{K_{SES}}(N_B)$) mukana kulkee tarpeeksi redundanttia, arvattavissa olevaa tietoa, voi Alice tarkastaa, onko viesti kryptattu oikealla istuntoavaimella vai ei. Jos sen sijaan kaikki, mikä salataan, on satunnaista, ei Alice voi tarkistaa tästä viestistä, onko se kryptattu oikealla avaimella vai ei. Tässä tapauksessa Alice ei vakuutu Bobin henkilöllisyydestä ennenkuin todellista dataa on siirretty istuntoavaimella salattuna, joten autentikointi on tällöin implisiittinen.

Yhteyksiä protokollassa muodostuu kaksi: Alicen ja Trentin sekä Alicen ja Bobin välille. Sanomia vaihdetaan viisi, mikä melko paljon. Off-line -laskentaa ei voi tehdä järkevästi. Luotetun osapuolen tehtävä on tässä hiukan vaativampi kuin Wide-mouth-frogissa mutta ei paljon: Trent joutuu luomaan hyvän sessioavaimen, tekemään aikaleiman ja suorittamaan kaksi kryptausta. Kelloja tarvitaan korjatussa protokollassa. Satunnaislukuja generoivat Alice (N_A), Trent (K_{SES}) sekä Bob (N_B).

c) Denning-Sacco-protokollassa Alice pyytää Trentiltä sertifikaattia omalle ja Bobin julkiselle avaimelle ja ottaa sen jälkeen yhteyttä Bobiin:

$$\begin{array}{rl} A \longrightarrow T & A, B \\ T \longrightarrow A & B, K_B, S_T(B, K_B), A, K_A, S_T(A, K_A) \\ A \longrightarrow B & E_B(T_A, K_{SES}, S_A(T_A, K_{SES})), \\ & \; B, K_B, S_T(B, K_B), A, K_A, S_T(A, K_A) \end{array}$$

Protokollassa oleva vika johtuu siitä, että kolmanteen viestiin sisältyvä Alicen allekirjoitus ei sisällä vastaanottajan, Bobin, nimeä. Saatuaan tämän allekirjoituksen käsiinsä voi Bob esiintyä Carolille Alicena:

$$\begin{array}{rl} B \longrightarrow T & B, C \\ T \longrightarrow B & C, K_C, S_T(C, K_C), B, K_B, S_T(B, K_B) \\ B(A) \longrightarrow C & E_C(T_A, K_{SES}, S_A(T_A, K_{SES})), \\ & \; C, K_C, S_T(C, K_C), A, K_A, S_T(A, K_A) \end{array}$$

Carol näkee vain viimeisen viestin, ja uskoo, että sen on lähettänyt Alice, koska se on täsmälleen samannäköinen kuin viesti, jonka oikea Alice olisi voinut lähettää. Huomaa, että tämä hyökkäys toimii vain niin kauan kuin aikaleima T_A on voimassa.

Protokollan voi korjata lisäämällä vastaanottajan nimen allekirjoituksen sisään:

$$\begin{array}{rl} A \longrightarrow T & A, B \\ T \longrightarrow A & B, K_B, S_T(B, K_B), A, K_A, S_T(A, K_A) \\ A \longrightarrow B & E_B(T_A, K_{SES}, S_A(B, T_A, K_{SES})), \\ & \; B, K_B, S_T(B, K_B), A, K_A, S_T(A, K_A) \end{array}$$

Tällöin tämä hyökkäys on mahdoton, koska Carol huomaa hyökkäyksessä, että Alicen allekirjoitus on tarkoitettu Bobille eikä Carolille.

Korjatussa protokollassa Alice epäilemättä ei voi autentikoida Bobia eksplisiittisesti, koska protokollaa päättyy, kun Bob saa ensimmäisen viestinsä. Sen sijaan Bob autentikoi Alicen eksplisiittisesti sen perusteella, että on saanut Alicelta allekirjoituksen, jonka tuoreuden aikaleima varmistaa.

Viestejä protokollassa on kolme, mikä on melko vähän. Off-line -laskentaa voidaan käyttää tehokkaasti hyväksi: Trent voi laskea allekirjoitukset valmiiksi ja jaella valmiita allekirjoituksia tarvittaessa. Tästä syystä on myös luotetun osapuolen tehtävä hyvin kevyt: sen ei tarvitse tarkastaa mitään eikä kryptata mitään.

Allekirjoitukset riittää laskea kerran, koska ne eivät vanhene koskaan, mutta tämä on myös ongelma: vaikka julkinen avain olisi paljastunut tai muuten vanhentunut, voi vihollinen esiintyä Trenttinä (jota tässä ei autentikoida), ja levittää edelleen Trentin vanhoja allekirjoituksia viallisille avaimille. Olisi hyvä, että allekirjoituksiin liitettäisiin aikaleimat (jolloin allekirjoituksia jouduttaisiin välillä laskemaan uudelleen).

Protokolla käyttää kelloja. Satunnaislukuja generoidaan yksi, nimittäin istuntoavain. Sen luo Alice.

d) Julkisen avaimen Needham-Schröder -protokollassa Alice ja Bob vaihtavat keskenään kaksi nonssia turvallisesti. Luotettua osapuolta käytetään vain julkisten avainten saamiseksi:

$$
\begin{array}{ll}
A \longrightarrow T & A, B \\
T \longrightarrow A & K_B^+, B, S_T(K_B^+, B) \\
A \longrightarrow B & E_B(N_A, A) \\
B \longrightarrow T & B, A \\
T \longrightarrow B & K_A^+, A, S_T(K_A^+, A) \\
B \longrightarrow A & E_A(N_A, N_B) \\
A \longrightarrow B & E_B(N_B)
\end{array}
$$

Protokollassa on vika, jonka avulla vihollinen I voi esiintyä Bobille Alicena. Vian kannalta Trentin kanssa käydyt keskustelut ovat irrelevantteja, joten keskitytään vain kolmeen Alicen ja Bobin vaihtamaan viestiin:

$$
\begin{array}{ll}
A \longrightarrow B & E_B(N_A, A) \\
B \longrightarrow A & E_A(N_A, N_B) \\
A \longrightarrow B & E_B(N_B)
\end{array}
$$

Hyökkäys perustuu siihen, että kakkosviesti ei sisällä tietoa siitä, kuka N_B:n on luonut, ja siihen, että Alice ottaa vapaaehtoisesti yhteyttä vihollliseen:

$$
\begin{array}{rl}
A \longrightarrow I & E_I(N_A, A) \\
I(A) \longrightarrow B & E_B(N_A, A) \\
B \longrightarrow I(A) & E_A(N_A, N_B) \\
I(A) \longrightarrow A & E_A(N_A, N_B) \\
A \longrightarrow I & E_I(N_B) \\
I(A) \longrightarrow B & E_B(N_B)
\end{array}
$$

Bob tietenkin näkee vain itsensä ja I:n välisen liikenteen, joka on:

$$
\begin{array}{ll}
I(A) \longrightarrow B & E_B(N_A, A) \\
B \longrightarrow I(A) & E_A(N_A, N_B) \\
I(A) \longrightarrow B & E_B(N_B)
\end{array}
$$

ja joka on siis kaikinpuolin oikealta näyttävä protokolla-ajo. Nyt Bob uskoo, että hän on yhteydessä Aliceen, vaikka onkin yhteydessä viholliseen.

Vika voidaan korjata lisäämällä Bobin nimi eksplisiittisesti kakkosviestin sisään:

$$
\begin{array}{ll}
A \longrightarrow B & E_B(N_A, A) \\
B \longrightarrow A & E_A(N_A, N_B, B) \\
A \longrightarrow B & E_B(N_B)
\end{array}
$$

Tällöin mainittu hyökkäys on mahdoton.

Kun korjaus on tehty, autentikaatio on molempiin suuntiin eksplisiittinen. A tunnistaa B:n, koska B kykenee palauttamaan N_A:n joka oli B:n avaimella salattu. Samoin B tunnistaa A:n, koska A kykenee palauttamaan N_B:n joka oli A:n avaimella salattu.

Protokolla on luotetun osapuolen kannalta melko kevyt. Off-line laskentaa voidaan hyödyntää, mutta tässä on sama vika kuin edellisessäkin protokollassa: allekirjoitukset ovat voimassa ikuisesti, joten vanhoja avaimia voidaan pakottaa käyttöön esiintymällä Trenttinä ja soittamalla sen vanhoja allekirjoituksia uudelleen.

Viestejä on yhteensä seitsemän, ja yhteyksiä muodostuu kolme. Lisäksi protokolla sisältää yhteensä kuusi julkisen avaimen operaatiota (kolme kryptausta ja kolme dekryptausta), joten se on tässäkin suhteessa melko raskas. Kelloja ei käytetä. Satunnaislukuja generoivat Alice (N_A) ja Bob (N_B).

$\boxed{\text{Laskuharjoitus 11}}$

Tehtävä 1

a) Yksisuuntainen X.509-autentikointi sisältää vain yhden viestin:

$$A \longrightarrow B \quad T_A, N_A, B, E_B(K_{AB}), S_A(T_A, N_A, B, E_B(K_{AB})$$

Tässä tietenkään A:n ei tarvitse vakuuttua mistään, koska ainoastaan lähettää viestin. B vakuuttuu siitä, että hänen saamansa viesti on äsken A:n lähettämä, koska se sisältää aikaleiman, joka on allekirjoitettu A:n avaimella. Aikaleiman voimassaoloaikana B voi muistaa nonssit N_A, jolloin hän ei hyväksy samaa viestiä uudelleen. Täten toistohyökkäykset pystytään torjumaan myös aikaleiman voimassaoloaikana.

b) Kaksisuuntaisessa X.509-autentikoinnissa lisätään toinen viesti:

$$\begin{aligned} A \longrightarrow B \quad & T_A, N_A, B, E_B(K_{AB}), S_A(T_A, N_A, B, E_B(K_{AB})) \\ B \longrightarrow A \quad & T_B, N_B, A, N_A, E_A(K_{BA}), S_B(T_B, N_B, A, N_A, E_A(K_{BA})) \end{aligned}$$

B vakuuttuu A:n viestin tuoreudesta kuten yllä. A vakuuttuu B:n lähettämän viestin tuoreudesta kahdesta syystä: ensinnäkin se sisältää aikaleiman (T_B), ja toiseksi se sisältää A:n äsken tuottaman nonssin N_A, molemmat B:n allekirjoittamana. Välittömästi huomataan, että T_B on toisessa viestissä tarpeeton, koska N_A takaa jo viestin tuoreuden.

c) Kolmisuuntaisessa X.509-autentikoinnissa lisätään vielä yksi viesti:

$$\begin{aligned} A \longrightarrow B \quad & T_A, N_A, B, E_B(K_{AB}), S_A(T_A, N_A, B, E_B(K_{AB})) \\ B \longrightarrow A \quad & T_B, N_B, A, N_A, E_A(K_{BA}), S_B(T_B, N_B, A, N_A, E_A(K_{BA})) \\ A \longrightarrow B \quad & N_B, B, S_A(N_B, B) \end{aligned}$$

Tässä B vakuuttuu kolmannen viestin tuoreudesta siitä, että se sisältää N_B:n, joka on B:n itsensä äsken generoima nonssi. T_B on tarpeeton, kuten yllä. Tässä ei ole nyt edes T_A välttämätön, koska B voi vakuuttua ensimmäisen viestin tuoreudesta jälkikäteen: saadessaan kolmannen viestin, B päättelee:

- A:n allekirjoitus kolmannessa viestissä on tuore, koska se sisältää äsken luomani nonssin.
- A näki äsken lähettämäni viestin, joka sisälsi N_A:n.

- A hyväksyi viestissä olleen N_A:n arvon, joten se oli sama, jonka A itse lähetti vähän aikaa sitten. Koska A pyrkii generoimaan tuoreita nonsseja, ja nonssi on sama, joka oli ensimmäisessä viestissä, oli ensimmäinen viesti tuore.

Tehtävä 2

Avainpalvelija tarvitsee yhden tietokannan, joka on kuvaus osapuolten nimiltä näiden julkisille avaimille. Toiminta-algoritmi olisi jotakin tämän kaltaista:

Vastaanota X:ltä kaksi nimeä, n_1 ja n_2
JOS n_1 ja n_2 ovat tietokannassa
NIIN
    Hae tietokannasta vastaavat avaimet K_1 ja K_2
    Luo satunnainen avain K_{SES}
    Tarkista kellonaika ja luo aikaleima T_T
    Lähetä X:lle $E_{K_1}(T_T, n_2, K_{SES})$ sekä $E_{K_2}(T_T, n_1, K_{SES})$
    Tuhoa K_{SES} muistista
MUUTOIN
    Katkaise yhteys

Tehtävä 3

a) Luotetun osapuolen lähettämistä paketeista on poistettu toisen osapuolen nimi. Vihollinen voi kohdistaa esimerkiksi seuraavan hyökkäyksen protokollaa vastaan:

$$\begin{aligned} A \longrightarrow I(T) \quad & A, B \\ I \longrightarrow T \quad & I, A \\ T \longrightarrow I \quad & A, E_I(T_T, K_{SES}), E_A(T_T, K_{SES}) \\ I \longrightarrow T \quad & I, B \\ T \longrightarrow I \quad & B, E_I(T_{T'}, K_{SES'}), E_A(T_{T'}, K_{SES'}) \\ I(T) \longrightarrow A \quad & B, E_A(T_T, K_{SES}), E_B(T_{T'}, K_{SES'}) \\ A \longrightarrow B \quad & A, E_B(T_{T'}, K_{SES'}) \end{aligned}$$

Tämä johtuu siitä, että B ei voi selvittää, kenen kanssa todellisuudessa K_{SES} on jaettu. Nyt A uskoo A:n ja B:n välisen instuntoavaimen olevan K_{SES} ja B uskoo, että se on $K_{SES'}$. Vihollinen tuntee molemmat, ja voi nyt asettua A:n ja B:n väliin, lukien kaikki viestit ja kryptaten ne uudelleen eri avaimella. Jos esimerkiksi A on lähettämässä viestiä X B:lle, toimii vihollinen näin:

$$
\begin{aligned}
A &\longrightarrow I(B) \quad E_{K_{SES}}(X) \\
I(A) &\longrightarrow B \quad E_{K_{SES'}}(X)
\end{aligned}
$$

b) Nyt viesteistä on poistettu aikaleimat, joten kerran luotetun osapuolen lähettämät paketit pysyvät oikean näköisinä ikuisesti. Vihollinen voi nyt esiintyä T:nä, ja soittaa vanhoja T:n lähettämiä istuntoavainviestejä yhä uudelleen. Jos jokin tällainen viesti sisältää istuntoavaimen, joka on murrettu, saa vihollinen pakotettua Alicen ja Bobin välille epäkelvon avaimen.

c) Aikaleimat on korvattu nonssilla N_A, jonka A on generoinut. Valitettavasti N_A ei kuitenkaan sano B:lle mitään, joten vihollinen voi nauhottaa Alicen Bobille lähettämän viestin, ja lähettää sen Bobille myöhemmin uudelleen esiintyen Alicena. Varsinaisesti tästä on hyötyä vain, jos vihollinen on onnistunut murtamaan aiemmin istuntoavaimen K_{SES}, joka on viestin sisällä.

Tehtävä 4

Normaali DH-avaintenvaihto menee näin: (g on julkinen generaattori ja p julkinen moduli, x ja y Alicen ja Bobin salaiset, äsken generoidut avaimet)

$$
\begin{aligned}
A &\longrightarrow B \quad g^x \bmod p \\
B &\longrightarrow A \quad g^y \bmod p
\end{aligned}
$$

Jonka jälkeen molemmat laskevat salaiseksi istuntoavaimeksi $K_{SES} = HASH(g^{xy})$. Jos g^x ja g^y ovat valmiiksi julkisessa hakemistossa, ei A:n tarvitse odottaa B:n lähettämää viestiä, vaan voi laskea istuntoavaimen valmiiksi ja siirtää ensimmäisen viestin mukana jo kryptattua dataa X:

$$A \longrightarrow B \quad A, E_{HASH(g^{xy})}(X)$$

Saadessaan viestin B hakee hakemistosta A:n julkisen avaimen g^x, minkä jälkeen pystyy itse laskemaan istuntoavaimen ja lukemaan X:n. A:n nimen on oltava viestissä mukana, jotta B voi hakea oikean avaimen hakemistosta.

a) Kahden osapuolen välille syntyy joka istunnossa sama istuntoavain, koska g^x ja g^y ovat vakioita (ne ovat hakemistossa, eikä niitä voi vaihtaa koko ajan).

b) Eräs ratkaisu on, että istuntoavaimena ei käytetä suoraan $HASH(g^{xy})$:tä, vaan g^{xy}:n ja jonkin satunnaisluvun hash-arvoa. Muutetaan Alicen lähettämää viestiä sisältämään tämä satunnaisluku, jonka nyt Alice generoi (R:n ei tarvitse olla salainen):

$$A \longrightarrow B \quad A, R, E_{HASH(R,g^{xy})}(X)$$

Saadessaan viestin, B voi laskea istuntoavaimen kuten A:kin, koska tuntee g^{xy}:n ja R:n. Kukaan muu ei voi laskea istuntoavainta, koska g^{xy} on salainen. Istuntoavain on joka istunnossa tuore, jos R on aina tuore. Protokolla sallii edelleen lähettäjän aloittaa salauksen välittömästi, kuten esimerkki osoittaa.

Tehtävä 5

Otaway-Rees protokolla menee näin (tässä E_A ja E_B) ovat avaimia Trentin ja Alicen tai Bobin välillä):

$$
\begin{aligned}
A &\longrightarrow B \quad I, A, B, E_A(N_A, I, A, B) \\
B &\longrightarrow T \quad I, A, B, E_A(N_A, I, A, B), E_B(N_B, I, A, B) \\
T &\longrightarrow B \quad I, E_A(N_A, K_{SES}), E_B(N_B, K_{SES}) \\
B &\longrightarrow A \quad I, E_A(N_A, K_{SES})
\end{aligned}
$$

Oletetaan ensin, että I on täysin satunnainen, eikä se ole arvattavissa etukäteen. Tällöin A varmistuu B:n identiteetistä eksplisiittisesti päätellen näin:

- Sain viestin, joka on salattu minun ja Trentin välisellä avaimella ($E_A(N_A, K_{SES})$), joten Trent on lähettänyt sen, koska en itse lähetä tällaisia viestejä.
- Se sisältää N_A:n, joten Trent on lukenut viestin $E_A(N_A, I, A, B)$.
- Trent suostuu lähettämään N_A:n eteenpäin vain, jos se saa viestin $E_A(N_A, I, A, B)$ kanssa B:n (jonka nimi on viestissä) salaisella avaimella salatun viestin $E_B(N_B, I, A, B)$.
- Trent on saanut siis tällaisen viestin. Koska I on tuore eikä sitä ole voitu etukäteen arvata, on B tehnyt tällaisen viestin äskettäin.
- Juuri B on siis vastannut lähettämääni yhteydenottopyyntöön äskettäin.

B ei voi kuitenkaan varmistua A:n identiteetistä eksplisiittisesti, koska ei voi tarkastaa indeksinumeron I tuoreutta: ensimmäinen viesti voisi olla uudelleensoitettu viesti, vaikka A olisi kauan sitten kadonnut maailmankaikkeudesta.

Jos indeksinumerot voidaan arvata etukäteen (esim. Alice kasvattaa indeksinumeroaan joka kerta yhdellä), on seuraava vihollisen hyökkäys mahdollinen. Tässä sulkeissa oleva numerot (n) viittavat protokolla-ajon numeroon (protokollaa ajetaan kolmeen kertaan):

$$
\begin{array}{rrll}
(1) & A \longrightarrow B & I, A, B, E_A(N_A, I, A, B) \\
(2) & I(A) \longrightarrow B & I+1, A, B, X \\
(2) & B \longrightarrow T & I+1, A, B, X, E_B(N_B, I+1, A, B) \\
 & (1) & \ldots \\
 & & \\
(3) & A \longrightarrow I(B) & I+1, A, B, E_A(N_{A'}, I+1, A, B) \\
(3) & I(B) \longrightarrow T & I+1, A, B, E_A(N_{A'}, I+1, A, B), E_B(N_B, I+1, A, B) \\
(3) & T \longrightarrow I(B) & I, E_A(N_{A'}, K_{SES}), E_B(N_B, K_{SES}) \\
(3) & I(B) \longrightarrow A & I, E_A(N_A, K_{SES})
\end{array}
$$

Nyt on I onnistunut esiintymään A:lle B:nä, joten aiemmin esitetty perustelu sille, että A autentikoisi B:n eksplisiittisesti on virheellinen mikäli indeksinumerot kyetään arvamaan. Huomaa, että A kuitenkin kykenee autentikoimaan B:n implisiittisesti, koska vihollinen ei tiedä, mikä avain K_{SES} on A:n ja B:nä esiintyvän I:n välillä, joten huijaus paljastuu, kun istunnon dataa aletaan siirtämään.

Tehtävä 6

a) PGP:ssä jokainen käyttäjä luottaa itseensä, joten relaatio R on refleksiivinen (kaikille x pätee: xRx). Muuten relaatio voi olla minkälainen tahansa, koska jokainen käyttäjä voi valita vapaasti, kehen luottaa ja kehen ei.

b) X.509-autentikointihierarkiassa luottavat hierarkian palvelimet kaikki toisiinsa, joten palvelimien osalta hierarkia on ekvivalenssirelaatio. Käyttäjät luottavat hierarkian johonkin palvelimeen, joten he luottavat koko hierarkiaan (koska palvelin luottaa koko hierarkiaan).

   Jos oletetaan, että käyttäjät eivät luota toisiinsa (tai että tätä ei ainakaan huomioida relaatiossa), ja että käyttäjät luottavat vain yhteen X.509-hierarkiaan (tai että muita hierarkioita ei tässä huomioida), on koko luottamusrelaatio transitiivinen ja euklidinen, mutta ei symmetrinen (hierarkia ei luota käyttäjiin) eikä välttämättä refleksiivinen (luottavatko käyttäjät itseensä?).

termi	relaatio R toteuttaa
refleksiivinen	$\forall x : xRx$
symmetrinen	$\forall x, y : xRy \rightarrow yRx$
transitiivinen	$\forall x, y, z : xRy \wedge yRx \rightarrow xRz$
euklidinen	$\forall x, y, z : xRy \wedge xRz \rightarrow yRz$