Infosec

`Drangevåg, Strømsvåg, Braskereid, Mikalsen, Gundersen`

Oppgave1:

1. Lag et C-program som krypterer og dekrypterer med encrypt decrypt-funksjonen og demonstrer at funksjonen fungerer.

```c
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

void encrypt_decrypt(unsigned int key,
                     unsigned char plain[],
                     unsigned char cipher[]) {

    /* Use the key as seed for the random function */
    srand(key);

    /* Get length of plaintext */
    int len = strlen(plain);

    /* For each byte in the plaintext */
    int i;
    for (i = 0; i < len; i++) {

        /* Obtain a random byte to use as pad */
        unsigned char pad = (unsigned char) rand();

        /* Encrypt a byte by XORing it with the pad */
        cipher[i] = plain[i] ^ pad;
    }
}


 void main() {
   uint k1 = 13;
   uint k2 = 13;
   char m1[] = "test123";
   char M[strlen(m1)];
   char m2[strlen(m1)];
   printf("Orginal melding:%s\n", m1);

   encrypt_decrypt(k1,m1,M);
   printf("kryptert:%s\n", M);

   encrypt_decrypt(k2,M,m2);
   printf("dekryptert: %s\n", m2);
 }
```

2. Hvorfor er bruken av rand-funksjonen problematisk?

`Fordi rand() kan ha forskjellig implementering på forskjellige maskiner. Samme nøkkel kan da gi forskjellig svar.`

3. I tillegg er krypteringsmetoden sårbar mot både brute force-angrep og nøkkelkollisjonsangrep. Forklar hvorfor. Hva måtte vi gjøre om vi ville "redde" krypteringsmetoden?

`seeden i srand() er begrenset av størrelsen på variabler datamaskinen kan håndtere, for en 64bit maskin blir dette 64bit. I tillegg har ikke srand() nødvendigvis noe beskyttelse mot at forskjellige nøkler kan gi samme resultat, og det kan fort oppstå nøkkelkollisjoner. For å "redde" krypteringsmetoden kan den brukes flere ganger etter hverandre med forskjellige nøkler. Maks nøkkel lengde blir da n*64bit, og man er bedre rustet mot bruteforce. Den vil fortsatt være utsatt for nøkkelkolisjon.`

Oppgåve2:

1. Bruk framgangsmåten under til å knekke WEP-krypteringa til Wi-Fi-et som er sett opp i klasserommet.

`Read 803986 packets (got 140129 ARP requests and 338780 ACKs), sent 342204 packets`

```
                          Aircrack-ng 1.2 rc4

               [00: 09: 48] Tested 343926 keys (got 58785 IVs)

        KB    depth   byte(vote)
         0    0/  1   4A(88320) 45(68608) F2(68608) 11(67840) 5E(67840)
         1    1/  2   57(72448) 73(69120) 68(66816) 4E(66304) 23(65792)
         2    0/  1   67(79872) 67(70400) B1(70400) 57(68864) D1(67840)
         3    0/  1   58(84992) 54(67584) 17(67072) 0A(66560) 98(66560)
         4    0/  1   53(72448) 54(68864) 81(68608) 89(68096) 35(67840)
         5    0/  1   37(82688) 65(67840) CC(67840) 34(67328) 4C(67328)
         6    0/  1   3F(82944) 7C(70656) 35(69376) C0(67840) 99(67584)
         7    0/  1   62(69888) 5B(68864) 78(68608) 3F(67840) F9(67840)
         8    0/  1   67(80896) F4(69888) A2(68352) 2C(67840) C7(67584)
         9    0/  1   62(77568) F9(67072) B2(66816) 80(66304) 82(66304)
        10    1/  1   46(69376) EB(68864) BC(68608) 45(67328) 67(67328)
        11    4/  1   6B(67072) 02(66816) B4(66816) 0B(66048) D9(65792)
        12    0/  3   DE(69376) 7F(68864) 59(68608) FE(68096) 5E(67840)


    KEY FOUND! [ 4A: 57: 67: 58: 53: 37: 3F: 62: 67: 62: 3F: 74: 52 ]
```

```
   (ASCII: JWgXS7?bgb?tR )
   Decrypted correctly: 100%
```

`wep-key:4A57675853373F6267623F7452`

2. Bruk Wireshark til å samanlikne pakke-capture før og etter dekryptering.

`Vi ser nå til og fra iper og innholdet i pakkene`

3. På side 389 i læreboka (Introduction to computer security) er det skildra fire kategoriar åtak mot kryptering. Kva kategori fell åtaket frå oppgåve a) under?
   `Vi benytter oss av IV-collision. Vi samler inn en rekke IV'er og så venter på at aksesspunktet skal gjenta en.`

4. I dei fyrste implementasjonane av WEP var ikkje IV -ane tilfeldige men sekvensnummer. Ein konsekvens av dette var høgt gjenbruk av IV -ar. Forklar korleis dette opna for nøkkelkollisjonsåtak mot WEP.

`Økt antall kollisjoner gjør det lettere å revesrere nøkkelstrengen.`

Oppgåve3:

1. Følg framgangsmåten for å kryptere eit bilete med AES-128 og ECB.

2. Krypter det same biletet med den same nøkkelen men med CBC.

3. Samanlikne dei to bileta. Kvifor har dei blitt som dei har blitt?

`ECB håndterer en block på 128-bit(lengden av nøkkelen) for så å fortsette til den neste. Dette fører til at dersom en klartekst blokk er lik som en annen vil den krypterte også være lik. Dette gjør igjen at vi fortsatt kan se mønsteret i bildet men ikke fargene.`

`CBC derimot "lenker" sammen blokkene slik at når den krypeter en blokk blir den krypterte avhengig av forige blokk. Dette fører til at mønsteret i bildet ikke lenger er synelig.`