FRANCESCA GRIMALDI

IDATT2503 - Cryptography - Autumn 2023

Cryptography Handout 3

Exercise 1

a) Given the recursive sequence/LFSR defined by

$$z_{i+4} = z_i + z_{i+1} + z_{i+2} + z_{i+3} (\mathrm{mod}\, 2)$$

What are the periods using the keys

1 $K = 1000$?

2 $K = 0011$?

3 $K = 1111$?

b) What are the periods with the same keys using the following LFSR?

$$z_{i+4} = z_i + z_{i+3} (\mathrm{mod}\, 2)$$

a1) key $K = 1000$
$K_1\ K_2\ K_3\ K_4$

$z_1 = K_1 = 1$
$z_2 = K_2 = 0$
$z_3 = K_3 = 0$
$z_4 = K_4 = 0$
$z_5 = z_{1+4} = z_1 + z_{1+1} + z_{1+2} + z_{1+3} \mod 2 = 1 \mod 2 = 1$
$z_6 = z_{2+4} = z_2 + z_3 + z_4 + z_5 \mod 2 = 1 \mod 2 = 1$
$z_7 = z_{3+4} = z_3 + z_4 + z_5 + z_6 \mod 2 = 2 \mod 2 = 0$
$z_8 = z_{4+4} = z_4 + z_5 + z_6 + z_7 \mod 2 = 2 \mod 2 = 0$
$z_9 = z_{5+4} = z_5 + z_6 + z_7 + z_8 \mod 2 = 2 \mod 2 = 0$
$z_{10} = z_{6+4} = z_6 + z_7 + z_8 + z_9 \mod 2 = 1 \mod 2 = 1$
$z_{11} = z_{7+4} = z_7 + z_8 + z_9 + z_{10} \mod 2 = 1 \mod 2 = 1$
...

The period is 5.

a2) key $K = 0011$ ($K_1 K_2 K_3 K_4$)

i	z_i	z_{i+1}	z_{i+2}	z_{i+3}
1	0	0	1	1
2	0	1	1	0
3	1	1	0	0
4	1	0	0	0
5	0	0	0	1
6	0	0	1	1

period = 5

a3) key $K = 1111$ ($K_1 K_2 K_3 K_4$)

i	z_i	z_{i+1}	z_{i+2}	z_{i+3}
1	1	1	1	1
2	1	1	1	0
3	1	1	0	1
4	1	0	1	1
5	0	1	1	1
6	1	1	1	1

period = 5

b1) key $K = 1000$ ($K_1 K_2 K_3 K_4$)

$z_1 = 1$
$z_2 = 0$
$z_3 = 0$
$z_4 = 0$
$z_5 = z_{1+4} = z_1 + z_{1+3} \mod 2 = 1 \mod 2 = 1$
$z_6 = z_{2+4} = z_2 + z_5 \mod 2 = 1 \mod 2 = 1$
$z_7 = z_{3+4} = z_3 + z_6 \mod 2 = 1 \mod 2 = 1$
$z_8 = z_{4+4} = z_4 + z_7 \mod 2 = 1 \mod 2 = 1$
$z_9 = z_{5+4} = z_5 + z_8 \mod 2 = 2 \mod 2 = 0$
$z_{10} = z_{6+4} = z_6 + z_9 \mod 2 = 1 \mod 2 = 1$
$z_{11} = z_{7+4} = z_7 + z_{10} \mod 2 = 2 \mod 2 = 0$
$z_{12} = z_{8+4} = z_8 + z_{11} \mod 2 = 1 \mod 2 = 1$
$z_{13} = z_{9+4} = z_9 + z_{12} \mod 2 = 1 \mod 2 = 1$
$z_{14} = z_{10+4} = z_{10} + z_{13} \mod 2 = 2 \mod 2 = 0$
$z_{15} = z_{11+4} = z_{11} + z_{14} \mod 2 = 0 \mod 2 = 0$
$z_{16} = z_{12+4} = z_{12} + z_{15} \mod 2 = 1 \mod 2 = 1$
$z_{17} = z_{13+4} = z_{13} + z_{16} \mod 2 = 2 \mod 2 = 0$
$z_{18} = z_{14+4} = z_{14} + z_{17} \mod 2 = 0 \mod 2 = 0$
$z_{19} = z_{15+4} = z_{15} + z_{18} \mod 2 = 0 \mod 2 = 0$

$z_{20} = z_{16+4} = 1$
$z_{21} = z_{17+4} = 1$
$z_{22} = z_{18+4} = 1$
$z_{23} = z_{19+4} = 1$
$z_{24} = z_{20+4} = 0$
$z_{25} = z_{21+4} = 1$
$z_{26} = z_{22+4} = 0$
$z_{27} = z_{23+4} = 1$
$z_{28} = z_{24+4} = 1$
$z_{29} = z_{25+4} = 0$
$z_{30} = z_{26+4} = 0$
$z_{31} = z_{27+4} = 1$
...

The period is 15.

b2) key $K = \underset{K_1}{0}\,\underset{K_2}{0}\,\underset{K_3}{1}\,\underset{K_4}{1}$

i	z_i	z_{i+1}	z_{i+2}	z_{i+3}
1	0	0	1	1
2	0	1	1	1
3	1	1	1	1
4	1	1	1	0
5	1	1	0	1
6	1	0	1	0
7	0	1	0	1
8	1	0	1	1
9	0	1	1	0
10	1	1	0	0
11	1	0	0	1
12	0	0	1	0
13	0	1	0	0
14	1	0	0	0
15	0	0	0	1
16	0	0	1	1

period = 15

b3) Key $K = 1\ 1\ 1\ 1$
$k_1\ k_2\ k_3\ k_4$

i	z_i	z_{i+1}	z_{i+2}	z_{i+3}
1	1	1	1	1
2	1	1	1	0
3	1	1	0	1
4	1	0	1	0
5	0	1	0	1
6	1	0	1	1
7	0	1	1	0
8	1	1	0	0
9	1	0	0	1
10	0	0	1	0
11	0	1	0	0
12	1	0	0	0
13	0	0	0	1
14	0	0	1	1
15	0	1	1	1
16	1	1	1	1

period = 15

Exercise 2

Vi define a HMAC as follows:

- Key $K = 1001$
- ipad = 0011
- opad = 0101
- h is the midsquare-hasing, calculationg $x^2 (\mathrm{mod}\, 2^8)$ and retrieving the middle four binary digits. Eg. $1011^2 = 01111001$ (with leading 0), giving us 1110 as hash value.

a) Find the HMAC for the message 0110

b) You receive the message 0111, with HMAC 0100. Is it reason to believe that the message is authenic?

$HMAC(K, x) = h((K \oplus opad) \| h((K \oplus ipad) \| x))$

a) $K \oplus opad = 1001 \oplus 0101 = 1100$

$K \oplus ipad = 1001 \oplus 0011 = 1010$

$h((K \oplus ipad) \| x) = h(10100110) = 1001$

$\downarrow dec^2 \bmod 128$

$166^2 \bmod 128 = 27556 \bmod 128 = 36$
1

bin
00100100

$h(1100\,1001) = \boxed{0100}$ ← is the HMAC for message 0110

↓ dec² mod 128

$201^2 \bmod 128 = 40401 \bmod 128 = 81$

↓ bin

01010001

b) $h((K \oplus ipad) \| x) = h(10100111) = 1100$

↓ dec² mod 128

$167^2 \bmod 128 = 27889 \bmod 128 = 113$

↓ bin

01110001

$h(11001100) = \boxed{0100}$ ← is the HMAC for message 0111

↓ dec² mod 128

$204^2 \bmod 128 = 41616 \bmod 128 = 16$

↓ bin

00010000

So there is reason to believe that the message is authentic.

Oppgave 3

Use the Cæsar cipher, with encryption $e_3(x) = x + 3 \pmod{2^8}$ and find the CBC-MAC to the following two messages:

$$x = 1101\,1111\,1010\,0001$$
$$x' = 0010\,1100\,0001\,1111$$

Considering the encryption function, we divide each message in 2 blocks of 8 bits each.

For message x, we have

- 1st iteration

$y_0 = 00000000$
$x_1 = 11011111$

$y_0 \oplus X_1 = X_1$

$e_3(y_0 \oplus X_1) = dec(X_1) + 3 \pmod{2^8} =$
$223 + 3 \pmod{2^8} = 226 = y_1$

- 2nd iteration

$y_1 = 11100010$
$X_2 = 10100001$

$y_1 \oplus X_2 = 01000011$

$e_3(y_1 \oplus X_2) = dec(y_1 \oplus X_2) + 3 \pmod{2^8} =$
$67 + 3 \pmod{2^8} = 70 = y_2$

→ y_2 is the CBC-MAC for message X
01000110

For message X', we have
- 1st iteration

$y_0 = 00000000$
$X_1' = 00101100$

$y_0 \oplus X_1' = X_1'$

$e_3(y_0 \oplus X_1') = dec(X_1') + 3 \pmod{2^8} =$
$44 + 3 \pmod{2^8} = 47 = y_1$

- 2nd iteration

$y_1 = 00101111$
$X_2' = 00011111$

$y_1 \oplus X_2' = 00110000$

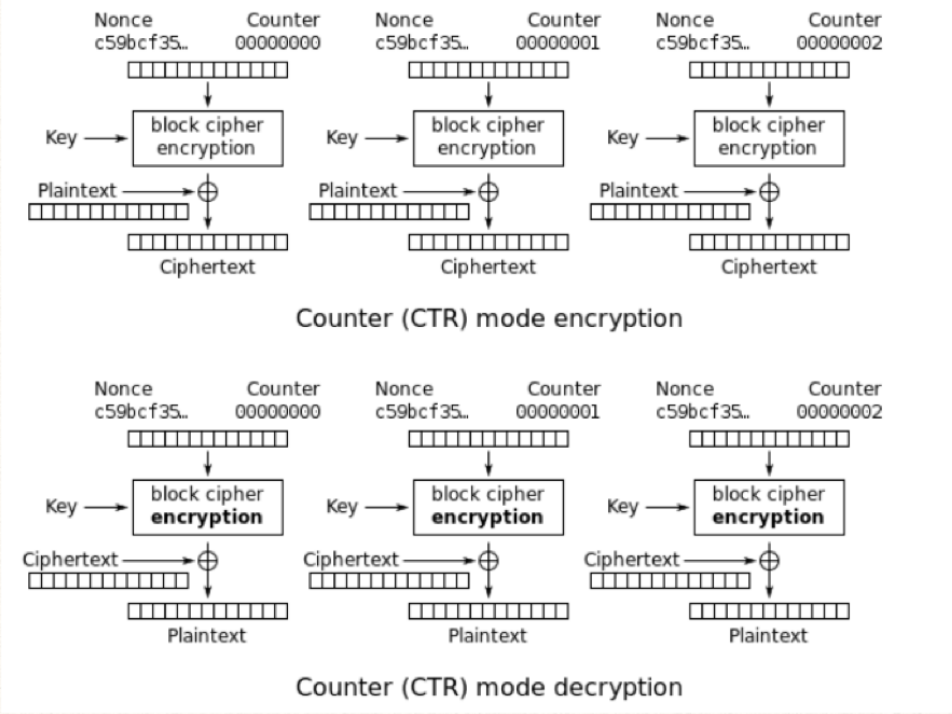
$e_3(y_1 \oplus X_2') = dec(y_1 \oplus X_2') + 3 \pmod{2^8} =$
$48 + 3 \pmod{2^8} = 51 = y_2$

→ y_2 is the CBC-MAC for message X'
00110011

(The same results are obtained with the code provided in the caesar-cbcmac.py file)

Exercise 4

We can construct a key-stream by using a block cipher in CTR-mode, by simply encrypting a sequence of values with a block cipher. It will use a *nonce* (initial value) combined with a counter. Here we will use addition modulo $2^8 = 256$ as the operation combining the nonce and the counter. The counter starts at 0, and add 1 to it for each round.
For the block cipher in parts a)-c) below, we use the SUBBYTES of the AES cipher, which substitutes a byte (8 bits) for another byte using a table lookup (actually a calculation in a certain Galois field, but its not in the curriculum).
Write programs to do the calculations.



a) Using the nonce = 01100101, write down the first 4 bytes produced with counter values 0,1,2,3.

b) What is the period of the key-stream?

c) Can the computation of the keystream be easily parallelized?

d) You intercept a message which is ecrypted by XOR-ing a key-stream generated by CTR, as described above, but with an unknown block cipher (not the one above). How could information about known plaintexts-ciphertext pairs be used to infer information about the key used?

(see ctr-keystream.py file)

a) The first 4 bytes produced are: 4d338545

b) Its period is 256.

c) Yes, it can be easily parallelized because at each iteration it only depends on the nonce and the counter, but not on previous calculations.

Therefore different blocks of the keystream can be computed in parallel at the same time.

d) Since in the encryption we XOR the plaintext with the keystream to produce the ciphertext, one could XOR the plaintext and the ciphertext to obtain the keystream used for those blocks.