COMP2700 Lab 8 -- Solutions

Exercise 1.

In this case we have $S_1 = 7667, S_2 = 50997, S_3 = 6447$. Since this PRNG is built using the linear congruential generator, we know that:

$S_1 \equiv AS_0 + B \; mod \; m$, $S_2 \equiv AS_1 + B \; mod \; m$ and $S_3 \equiv AS_2 + B \; mod \; m$.

Using the last two equations, we get:

$$(S_2 - S_3) \equiv A(S_1 - S_2) \; mod \; m$$

Hence $A = (S_2 - S_3)(S_1 - S_2)^{-1} \; mod \; m$.

To calculate A, we calculate the inverse of $(S_1 - S_2)$ modulo m, i.e.,

$$(7667 - 50997)^{-1} mod \; 64283.$$

Since $(7667 - 50997) = 20953 \; mod \; 64283$, using the provided script, we compute

$$20953^{-1} mod \; m = 63967.$$

So $A = (S_2 - S_3) \cdot 63967 \; mod \; m = 177$. Once we know A, we can calculate $B = S_2 - AS_1 \; mod \; m = 43881$.

S_0 can then be calculated as $S_0 = (S_1 - B) \cdot A^{-1} \; mod \; m = 45193$.

Note that in this case, m = 64283 is a prime number, so multiplicative inverse of any non-zero number modulo m always exists.

Exercise 2.

a) $A(x) + B(x) = (x^2 + 1) + (x^3 + x^2 + 1) = x^3 + 2x^2 + 2 = x^3$.

b) Since addition and subtraction are the same in $GF(2^4)$, we have
$$A(x) - B(x) = A(x) + B(x) = x^3.$$

c) $A(x) \cdot B(x)$
$= (x^2 + 1)(x^3 + x^2 + 1)$
$= (x^5 + x^4 + x^2 + x^3 + x^2 + 1)$
$= x^5 + x^4 + x^3 + 1$
We need to compute the remainder of this modulo $P(x)$. It is easier to simplify first the leading terms x^5 and x^4.
$$x^4 = 1 \cdot P(x) - (x + 1) = 1 \cdot P(x) + (x + 1) \equiv (x + 1) \; mod \; P(x).$$
$$x^5 = x \cdot x^4 = x^2 + x \; mod \; P(x).$$
Substituting these back to the equation for $A(x) \cdot B(x)$ we get:
$$A(x) \cdot B(x) = (x^2 + x) + (x + 1) + x^3 + 1 \equiv x^3 + x^2 \; mod \; P(x).$$

Exercise 3.

We first need to find the inverse of $B(x)\ mod\ P(x)$ using the provided inverse table. Expanding $B(x)$ to include all the 0 coefficients, we have:

$$B(x) = 0 \cdot x^7 + 1 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 0 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x + 0$$

So $B(x)$ can be represented as the bitstring 01010010, which is 52 in HEX. So the inverse of $B(x)$ corresponds to the entry in the inverse tablet at row 5 and column 2, which is 05. This in binary is 00000101, which gives us the polynomial:

$$B^{-1}(x) = x^2 + 1.$$

The polynomial $A(x) \cdot B^{-1}(x)\ mod\ P(x)$ can then be calculated as follows:

$$\begin{aligned} A(x) \cdot B^{-1}(x) &= (x^3 + 1) \cdot (x^2 + 1) \\ &= x^5 + x^3 + x^2 + 1\ mod\ P(x) \end{aligned}$$

Exercise 4.

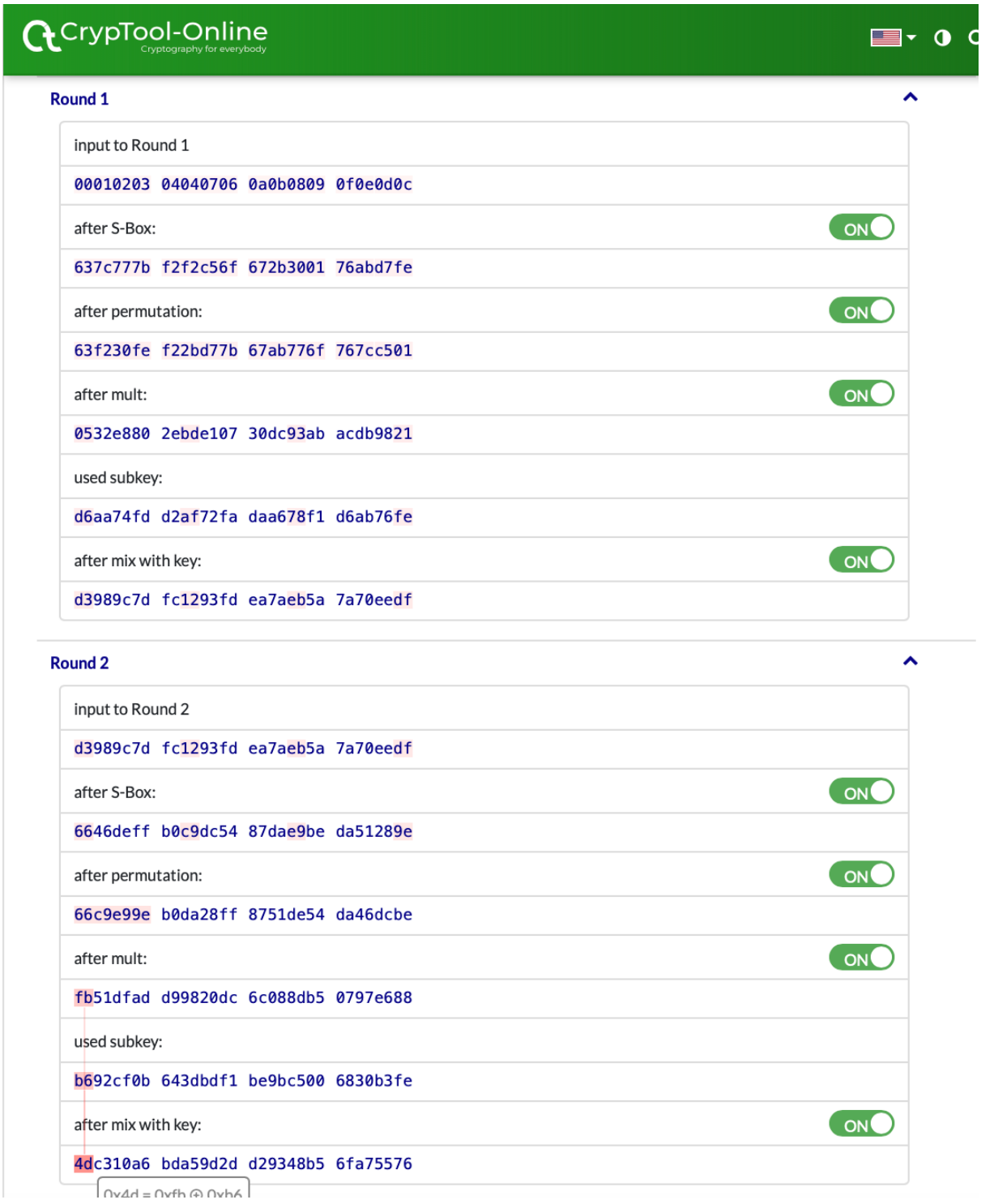
a. The first subkey is d6aa74fd d2af72fa daa678f1 d6ab76fe, and the second subkey is b692cf0b 643dbdf1 be9bc500 6830b3fe.
b. There is a key addition layer just before Round 1. So the input to Round 1 is obtained from performing an XOR between the original input and the key.
c. With the original input 00000000 01010101 02020202 03030303, the resulting output of Round 1 is d3989c7d d58b0a4d ea7aeb5a 7a70eedf.
   With the modified input 01000000 01010101 02020202 03030303, the resulting Round 1 output is ed87835c d58b0a4d ea7aeb5a 7a70eedf. It's quite obvious that only the first four bytes change. The Hamming distance between the two ciphertexts is 17, so there are exactly 17 bits that change in the output of Round 1 for a 1 bit flip in the input.
d. Two rounds are sufficient to achieve an average of 50% bit flips in the output in response to flipping 1 bit in the input. For the specific input and key combination for this exercise, here are some samples of bit flips combinations:

Position of bit flip in input	Hamming distance at Round 2
8	61
16	60
24	57
32	63
40	82
48	64
56	61
64	62
72	76
80	67
88	78
96	71
104	66

112	65
120	66
128	66
Average	66.6

The fact that 2 rounds are sufficient to provide 50% bit flips was already anticipated in the design of AES. The Cryptool website provides a nice visualisation on the dependency of each byte in the output of a round on inputs from previous rounds. You can check that each byte in the output of Round 2 is indeed dependent on every byte in the input of Round 1 (hence also every byte in the original input), so a change in one byte (by a bit flip) is likely to influence every byte in the output of the second round.

The figure below shows, for example, that the first byte in the output of Round 2 is potentially influenced by the value of every byte in the input to Round 1 (the highlighted bytes indicate they were used in the computation that ultimately derives the first byte of the output of Round 2).


CrypTool-Online
Cryptography for everybody
Round 1
input to Round 1
00010203 04040706 0a0b0809 0f0e0d0c
after S-Box:
ON
637c777b f2f2c56f 672b3001 76abd7fe
after permutation:
ON
63f230fe f22bd77b 67ab776f 767cc501
after mult:
ON
0532e880 2ebde107 30dc93ab acdb9821
used subkey:
d6aa74fd d2af72fa daa678f1 d6ab76fe
after mix with key:
ON
d3989c7d fc1293fd ea7aeb5a 7a70eedf
Round 2
input to Round 2
d3989c7d fc1293fd ea7aeb5a 7a70eedf
after S-Box:
ON
6646deff b0c9dc54 87dae9be da51289e
after permutation:
ON
66c9e99e b0da28ff 8751de54 da46dcbe
after mult:
ON
fb51dfad d99820dc 6c088db5 0797e688
used subkey:
b692cf0b 643dbdf1 be9bc500 6830b3fe
after mix with key:
ON
4dc310a6 bda59d2d d29348b5 6fa75576
0x4d = 0xfb ⊕ 0xb6

Extension Exercises (Optional)

Exercise 5.

From Exercise 1, we know that as soon as we figure out S_1, S_2, S_3 we can calculate the values of A, B, S_0 which give us the key. Since every random number is encoded as two bytes in the key stream, the first six bytes of the key stream encode those three numbers S_1, S_2, S_3. From the information that the plaintext was an HTML file, we guess that the first 6 bytes of the plaintext could be the tag "<html>". Let K[0:6] be the first six bytes of the key stream, and let X[0:6] be the first six bytes of the plaintext, and Y[0:6] be the first six bytes of the ciphertext, then we have the following relation:

$$Y[0:6] = X[0:6] \oplus K[0:6]$$

and therefore:

$$K[0:6] = X[0:6] \oplus Y[0:6].$$

This gets us the first six bytes of the key stream, which contains the first three random numbers S_1, S_2, S_3. Speficially, S_1 corresponds to K[0:2], S_2 corresponds to K[2:4] and S_3 corresponds to K[4:6]. In python we can use the function bytes_to_long to convert byte sequences back to integer. The following is a complete python script to extract the key stream and compute the key.

```
from Crypto.Util.number import *
from Crypto.Util.strxor import *

with open('doc.enc', 'rb') as f:
    ct = f.read()

known_pt = b'<html>'

# get the first six bytes of keystream
S = strxor(known_pt[0:6], ct[0:6])

# get the values of the first three random numbers
S1 = bytes_to_long(S[0:2])
S2 = bytes_to_long(S[2:4])
S3 = bytes_to_long(S[4:6])

print("S1 = %d, S2 = %d, S3 = %d" % (S1,S2,S3))

A = ((S2 - S3) * inverse(S1 - S2, 64283)) % 64283
B = (S2 - A*S1) % 64283
S0 = ((S1 - B) * inverse(A, 64283)) % 64283

print("The key is (S0 = %d, A = %d, B = %d)" % (S0, A, B))
```

Running this script gives us $S_0 = 1234, A = 678, B = 2532$. We can now use this key to decrypt the file doc.enc, using the provided lcgcipher.py:

```
$ python3 lcgcipher.py 1234 678 2532 doc.enc plaintext.html

$ cat plaintext.html

<html>

<body>Linear congruential generator is a very bad choice for a stream cipher. DO NOT use this
stream cipher in practice.</body>



</html>
```

Exercise 6.

a) $d = 1 \cdot (b_7x^7 + \cdots + b_0) = b.$

b) $d = x(b_7x^7 + \cdots + b_0) = b_7x^8 + b_6x^7 + \cdots + b_0x$
Since $x^8 \equiv x^4 + x^3 + x + 1 \; mod \; P(x)$ we have
$$d = b_7(x^4 + x^3 + x + 1) + b_6x^7 + \cdots + b_0x$$
$$= b_6x^7 + b_5x^6 + b_4x^5 + [b_7 + b_3]x^4 + [b_7 + b_2]x^3 + b_1x^2 + [b_7 + b_0]x + b_7$$
So $d_7 = b_6, d_6 = b_5, d_5 = b_4, d_4 = b_7 + b_3, d_3 = b_7 + b_2, d_2 = b_1, d_1 = b_7 + b_0, d_0 = b_7.$

c) $d = (x + 1)(b_7x^7 + \cdots + b_0)$. From a) and b) above, we have that:
$$d = (b_6 + b_7)x^7 + (b_5 + b_6)x^6 + (b_4 + b_5)x^5 + (b_3 + b_4 + b_7)x^4 + (b_2 + b_3 + b_7)x^3 + (b_1 + b_2)x^3 + (b_0 + b_1 + b_7)x + (b_0 + b_7).$$

Exercise 7.

- $RC[8] = x^7 = (1000\ 0000)_2$
- $RC[9] = x^8 = x^4 + x^3 + x + 1 = (0001\ 1011)_2$ (Recall that $x^8 \equiv x^4 + x^3 + x + 1 \; mod \; P(x)$ where $P(x) = x^8 + x^4 + x^3 + x + 1$.)
- $RC[10] = x^9 = x \cdot x^8 = x^5 + x^4 + x^2 + x = (0011\ 0110)_2$