# Final Exam

- Time: 10:30-12:30
- Do not use "písané písmo" but "paličkové".
- Write your answers on the provided answer sheets. Additional sheets will be supplied upon request. Please ensure that your full name is clearly written on each page of the answer sheets.
- Include detailed computation steps for all solutions. Answers without supporting calculations will receive a score of zero.

**Question 1.** (10 marks) For the classical ciphers discussed in this course, we assume messages consist of English letters (`A` - `Z`), where each letter is mapped to an element in $\mathbb{Z}_{26}$. Table 1 outlines the specific mapping between the letters and their corresponding elements in $\mathbb{Z}_{26}$.

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |

| U | V | W | X | Y | Z |
|---|---|---|---|---|---|
| 20 | 21 | 22 | 23 | 24 | 25 |

Table 1: Converting English letters to elements in $\mathbb{Z}_{26}$.

Recall the definition of Affine cipher as follows:

**Definition 1** (Affine cipher)**.** Let $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$ and $\mathcal{K} = \{ (a, b) \mid a \in \mathbb{Z}_{26}^*, b \in \mathbb{Z}_{26} \}$. For each key $(a, b)$, define

$$E_{(a,b)} : \mathbb{Z}_{26} \to \mathbb{Z}_{26}, \quad p \mapsto ap + b \bmod 26; \quad D_{(a,b)} : \mathbb{Z}_{26} \to \mathbb{Z}_{26}, \quad c \mapsto a^{-1}(c - b) \bmod 26.$$

The cryptosystem $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where $\mathcal{E} = \{ E_{(a,b)} : (a, b) \in \mathcal{K} \}$, $\mathcal{D} = \{ D_{(a,b)} : (a, b) \in \mathcal{K} \}$, is called the *affine cipher*.

a) (2 mark) How many possible keys are there for an affine cipher?

b) (2 marks) Encrypt the word `STROM` using affine cipher with the key $k = (3, 1)$.

c) (6 marks) Frequency analysis is a cryptographic technique used to break classical ciphers by studying the frequency of letters or groups of letters in a ciphertext. It exploits the fact that in any given language, certain letters or letter combinations occur more frequently than others. By comparing these frequencies in a ciphertext to the known frequency distribution of a language, an attacker can infer the likely plaintext.

Assuming the language is English, we analyze the frequency distribution of English letters in typical English text. These frequencies are presented in Table 2.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| A | 0.082 | B | 0.015 | C | 0.028 | D | 0.043 | E | 0.127 | F | 0.022 |
| G | 0.020 | H | 0.061 | I | 0.070 | J | 0.002 | K | 0.008 | L | 0.040 |
| M | 0.024 | N | 0.067 | O | 0.075 | P | 0.019 | Q | 0.001 | R | 0.060 |
| S | 0.063 | T | 0.091 | U | 0.028 | V | 0.010 | W | 0.023 | X | 0.001 |
| Y | 0.020 | Z | 0.001 | | | | | | | | |

Table 2: Probabilities of each letter in a standard English text.

Consider the following ciphertext:

`VCVIRSKPOFPNZOTHOVMLVYSATISKVNVLIVSZVR`.

Apply frequency analysis to break the cipher and determine the original plaintext.

*Solution.*

a) (2 marks) We know that $26 = 2 \times 13$. By Theorem from lecture,

$$\varphi(26) = 26 \times \left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{13}\right) = 12.$$

So there are 12 possible values for $a \in \mathbb{Z}_{26}^*$. And there are 26 possible values for $b \in \mathbb{Z}_{26}$. Then the total number of possible keys $(a, b)$ is $12 \times 26 = 312$.

b) (2 marks) To encrypt the word `STROM`, we compute:

$$\begin{aligned} &3 \times 18 + 1 = 55 \equiv 3 \bmod 26, \quad 3 \times 19 + 1 = 58 \equiv 6 \bmod 26, \\ &3 \times 17 + 1 = 52 \equiv 0 \bmod 26, \quad 3 \times 14 + 1 = 43 \equiv 17 \bmod 26, \\ &3 \times 12 + 1 = 37 \equiv 11 \bmod 26. \end{aligned}$$

So the ciphertext is `DGARL`. We can list the correspondence between plaintext and ciphertext as follows:

| S | T | R | O | M |
|---|---|---|---|---|
| 18 | 19 | 17 | 14 | 12 |
| 3 | 6 | 0 | 17 | 11 |
| D | G | A | R | L |

c) (6 marks) We can calculate the frequencies of each letter that appear in the text:

| V | S | I | O | R | K | P | N | Z | T | L | C | F | H | M | Y | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 | 4 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |

The most frequent letter is `V`, and the second most frequent one is `S`. Thus, it makes sense to assume `V` is the ciphertext corresponding to `E` and `S` to `T`. (1 marks)

Let the key be $(a, b)$. By Definition, we have the following equations:

$$4a + b = 21 \bmod 26, \quad 19a + b = 18 \bmod 26,$$

which gives

$$15a = 23 \bmod 26.$$

By the extended Euclidean algorithm,

$$26 = 15 \times 1 + 11, \quad 15 = 11 \times 1 + 4, \quad 11 = 4 \times 2 + 3, \quad 4 = 3 + 1,$$

and

$$\begin{aligned} 1 &= 4 - 3 = 4 - (11 - 4 \times 2) = -11 + 4 \times 3 = -11 + (15 - 11) \times 3 \\ &= 15 \times 3 - 11 \times 4 = 15 \times 3 - (26 - 15) \times 4 = 15 \times 7 - 26 \times 4. \end{aligned}$$

Hence, we have $15^{-1} \bmod 26 = 7$ and

$$a = 23 \times 15^{-1} \bmod 26 = 23 \times 7 \bmod 26 = 5 \bmod 26.$$

(2 marks)

Furthermore, we get

$$b = 21 - 4a \bmod 26 = 21 - 4 \times 5 \bmod 26 = 1.$$

(1 mark)

To decrypt the message, we compute the decryption key by finding $a^{-1} \bmod 26 = 5^{-1} \bmod 26$:

$$26 = 5 \times 5 + 1 \Longrightarrow 1 = 26 - 5 \times 5 \Longrightarrow 5^{-1} \bmod 26 = -5 \bmod 26 = 21.$$

(1 mark)

Applying the decryption key $(21, 1)$ to the ciphertext, we get the following plaintext:

`EVERYTHING IS KNOWN EXCEPT FOR THE SECRET KEY.`

(1 mark)

**Question 2.** (10 marks) In this question, we consider RSA encryption and decryption. Suppose Bob would like to generate his private and public keys for RSA. Bob randomly generates $p = 29$ and $q = 41$.

a) (1 mark) Then he computes

$$n = ? \quad \varphi(n) = ?$$

b) (2 mark) From $\mathbb{Z}^*_{\varphi(n)}$, Bob chooses his private key $e = 3$. Compute using the extended Euclidean algorithm, the private key $d$ for Bob.

c) (1 mark) Alice would like to send plaintext $m = 2$ to Bob, using Bob's public key $n, e$. Alice computes ciphertext

$$c = ?$$

d) (3 marks) Bob receives a message $c = 142$ from Alice. Then to decrypt the message $c = 142$ with CRT-based RSA, Bob computes

$$m_p = ? \quad m_q = ? \quad y_p = ? \quad y_q = ?$$

Using Garner's algorithm, Bob gets

$$m = ?$$

e) (3 marks) Now we consider a malicious attacker who carries out a Bellcore attack during the RSA decryption. Suppose she injects fault during the computation of $m_p$ and the faulty $m'_p = 1$. Then the faulty message will be

$$m' = ?$$

With the knowledge of $m$, $m'$ and the public key $n, e$, how can the attacker recover the secret key $d$?

*Solution.*

a) (1 mark)

$$n = 29 \times 41 = 1189, \quad \varphi(n) = 28 \times 40 = 1120.$$

b) (2 marks) By the extended Euclidean algorithm

$$1120 = 3 \times 373 + 1 \Longrightarrow 1 = 1120 - 3 \times 373.$$

Hence Bob's private key $d = -373 \bmod 1120 = 747$.

c) (1 mark) Alice computes

$$c = m^e \bmod n = 2^3 \bmod 1189 = 8.$$

d) (3 marks) After receiving the ciphertext $c = 142$, with CRT-based RSA implementation, Bob computes

$$\begin{aligned} m_p &= c^{d \bmod (p-1)} \bmod p = 142^{747 \bmod 28} \bmod 29 = 26^{19} \bmod 29 = 11, \text{(1 mark)} \\ m_q &= c^{d \bmod (q-1)} \bmod q = 142^{747 \bmod 40} \bmod 41 = 19^{27} \bmod 41 = 11. \text{(1 mark)} \end{aligned}$$

To compute $26^{19} \bmod 29$, we note that

$$\begin{aligned} 26^2 \bmod 29 &= 9, \\ 26^4 \bmod 29 &= 9^2 \bmod 29 = 23, \\ 26^8 \bmod 29 &= 23^2 \bmod 29 = 7, \\ 26^{16} \bmod 29 &= 7^2 \bmod 29 = 20. \end{aligned}$$

Thus

$$26^{19} \bmod 29 = 26^{16} \times 26^2 \times 26 \bmod 29 = 20 \times 9 \times 26 \bmod 29 = 11.$$

Similarly,

$$\begin{aligned} 19^2 \bmod 41 &= 33 \\ 19^3 \bmod 41 &= 33 \times 19 \bmod 41 = 12 \\ 19^9 \bmod 41 &= 12^3 \bmod 41 = 6 \\ 19^{27} \bmod 41 &= 6^3 \bmod 41 = 11. \end{aligned}$$

By the extended Euclidean algorithm

$$41 = 29 + 12,\ 29 = 12 \times 2 + 5,\ 12 = 5 \times 2 + 2,\ 5 = 2 \times 2 + 1,$$

which gives

$$\begin{aligned} 1 &= 5 - 2 \times (12 - 5 \times 2) = -2 \times 12 + (29 - 12 \times 2) \times 5 \\ &= 29 \times 5 - 12 \times (41 - 29) = -41 \times 12 + 29 \times 17. \end{aligned}$$

We have (0.5 marks)

$$\begin{aligned} y_p &= p^{-1} \bmod q = 29^{-1} \bmod 41 = 17 \bmod 41, \\ y_q &= q^{-1} \bmod p = 41^{-1} \bmod 29 = -12 \bmod 29 = 17 \bmod 29. \end{aligned}$$

By Garner's algorithm, (0.5 marks)

$$m = m_p + ((m_q - m_p)y_p \bmod q)p = 11 + 0 = 11$$

e) (3 marks) The faulty message is given by

$$\begin{aligned} m' &= m'_p + ((m_q - m'_p)y_p \bmod q)p = 1 + ((11 - 1) \times 17 \bmod 41) \times 29 \\ &= 1 + 6 \times 29 = 175. \end{aligned}$$

According to the Bellcore attack,

$$q = \gcd(m' - m, n) = \gcd(175 - 11, 1189) = \gcd(164, 1189).$$

By the Euclidean algorithm

$$\begin{aligned} 1189 &= 164 \times 7 + 41, & \gcd(164, 1189) &= \gcd(164, 41), \\ 164 &= 41 \times 4, & \gcd(164, 41) &= 41. \end{aligned}$$

Hence $q = 41$ and

$$p = \frac{n}{q} = 29.$$

Then the attacker can compute

$$\varphi(n) = (p-1)(q-1) = 1120.$$

The private key

$$d = e^{-1} \bmod \varphi(n) = 3^{-1} \bmod 1120.$$

can be found using the extended Euclidean algorithm.

**Question 3.** (5 marks) The stochastic leakage model assumes each bit of the target intermediate value $\boldsymbol{v} = v_{m_v-1} v_{m_v-2} \dots v_1 v_0$ has a different leakage.

$$\mathcal{L}(\boldsymbol{v}) = \sum_{s=0}^{m_v-1} \alpha_s v_s + \text{noise},$$

where noise $\sim \mathcal{N}(0, \sigma^2)$ denotes the noise with variance $\sigma^2$ and $\alpha_s$ $(s = 0, 1, \dots, m_v - 1)$ are real numbers. We have discussed how to profile the DUT to find estimations for $\alpha_s$. Let

$$\boldsymbol{\ell}_{pf} := (l_{\text{POI}}^{1,pf}, l_{\text{POI}}^{2,pf}, \dots, l_{\text{POI}}^{M_{pf},pf})$$

be the vector of leakages at $t =$POI from all $M_{pf}$ profiling traces, where $l_{\text{POI}}^{j,pf}$ is the leakage at POI from the $j$th profiling trace.

Furthermore, for the $j$th profiling trace, let

$$\boldsymbol{v}_j^{pf} = v_{j(m_v-1)}^{pf} \dots v_{j1}^{pf} v_{j0}^{pf}, \quad j = 1, 2, \dots, M_{pf}$$

be the corresponding target intermediate value. Then we compute matrix $M_{\boldsymbol{v}}$

$$M_{\boldsymbol{v}} := \begin{pmatrix} v_{10}^{pf} & v_{11}^{pf} & \dots & v_{1(m_v-1)}^{pf} \\ v_{20}^{pf} & v_{21}^{pf} & \dots & v_{2(m_v-1)}^{pf} \\ \vdots & \vdots & \ddots & \vdots \\ v_{M_{pf}0}^{pf} & v_{M_{pf}1}^{pf} & \dots & v_{M_{pf}(m_v-1)}^{pf} \end{pmatrix}$$

The estimated values $\hat{\alpha}_s$ for $\alpha_s$ are given by

$$(\hat{\alpha}_0, \hat{\alpha}_1, \dots, \hat{\alpha}_{m_v-1}) = \left(M_{\boldsymbol{v}}^T M_{\boldsymbol{v}}\right)^{-1} M_{\boldsymbol{v}}^T \boldsymbol{\ell}_{pf}.$$

Suppose we are interested in the computation of the first round of PRESENT. Our target intermediate value $\boldsymbol{v}$ is the 0th Sbox output in the first round. We have collected $M_{pf} = 5000$ traces.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C | 5 | 6 | B | 9 | 0 | A | D | 3 | E | F | 8 | 4 | 7 | 1 | 2 |

Table 3: PRESENT Sbox.

a) (2 marks) The first trace in our dataset corresponds to the 0th nibble of the plaintext= 4 and the 0th nibble of the first round key= 7. Thus the intermediate value for the first trace is given by:

$$\boldsymbol{v}_1^{pf} = ?$$

b) (1 mark) The first row of our matrix $M_{\boldsymbol{v}}$ is given by?

c) (1 mark) Suppose we got the following estimated values for $\alpha_s$s:

$$\hat{\alpha}_0 \approx -0.02019, \quad \hat{\alpha}_1 \approx -0.02027, \quad \hat{\alpha}_2 \approx -0.01920, \quad \hat{\alpha}_3 \approx -0.02039.$$

Then according to the stochastic leakage model, the leakage of $\boldsymbol{v} = v_3v_2v_1v_0$ is given by

$$\mathcal{L}(\boldsymbol{v}) = ?.$$

d) (1 mark) In particular,

$$\mathcal{L}(\mathtt{E}) = ?$$

*Solution.*

a) (2 marks) The first trace in our dataset corresponds to the 0th nibble of the plaintext= 4 and the 0th nibble of the first round key= 7. Thus the intermediate value for the first trace is given by:

$$\boldsymbol{v}_1^{pf} = \mathrm{SB}_{\mathrm{PRESENT}}(\mathtt{4} \oplus \mathtt{7}) = \mathrm{SB}_{\mathrm{PRESENT}}(\mathtt{3}) = \mathtt{B} = 1011_2.$$

b) (1 mark) And the first row of our matrix $M_{\boldsymbol{v}}$ is given by

$$\begin{pmatrix} 1 & 1 & 0 & 1 \end{pmatrix}.$$

c) (1 mark) The leakage of a $\boldsymbol{v} = (v_0, v_1, v_2, v_3)$ is given by

$$\mathcal{L}(\boldsymbol{v}) = \hat{\alpha_0}v_0 + \hat{\alpha_1}v_1 + \hat{\alpha_2}v_2 + \hat{\alpha_3}v_3 + \text{noise}.$$

d) (1 mark) $\mathcal{L}(\mathtt{E}) = \hat{\alpha_1} + \hat{\alpha_2} + \hat{\alpha_3} + \text{noise} = -0.05986 + \text{noise}$.

**Question 4.** (5 marks) We have introduced a method for implementing masked PRESENT by using precomputed lookup tables, denoted as T1 and T2. For any $\boldsymbol{v} \in \mathbb{F}_2^4$, any input mask $\mathfrak{m}_{\text{in}} \in \mathbb{F}_2^4$ and the corresponding output mask $\mathfrak{m}_{\text{out}} \in \mathbb{F}_2^4$ for PRESENT Sbox, the table T1 is defined as:

$$\mathrm{T1}[\boldsymbol{v} \oplus \mathfrak{m}_{\text{in}}, \mathfrak{m}_{\text{in}}] = \mathrm{SB}(\boldsymbol{v}) \oplus \mathfrak{m}_{\text{out}}. \tag{1}$$

Additionally, T2 is used to keep track of the masks:

$$\mathrm{T2}[\mathfrak{m}_{\text{in}}] = \mathfrak{m}_{\text{out}}, \quad \mathfrak{m}_{\text{in}} = \mathtt{0}, \mathtt{1}, \ldots, \mathtt{F}. \tag{2}$$

This approach eliminates the need to regenerate a masked S-box table each time the input mask changes.

Please complete the masked PRESENT algorithm by filling in each missing line based on the specific instructions provided.

a) For Line 7, implement the computation of the masked Sbox.

b) For line 8, store the resulting output masks from the Sbox operation.

c) For line 9, apply the pLayer operation to the current cipher state.

d) For line 10, apply the pLayer operation to the masks.

e) For line 12, remove masks from the ciphertext.

**Algorithm 1:** Masked implementation of PRESENT.

**Input:** $p$, T1, T2, $K_i$ $(i = 1, 2, \ldots, 32)$// $p$ is the plaintext for encryption; T1 is the table for masked Sbox as given in Equation 1; T2 specifies the output mask given the input mask for PRESENT Sbox as defined in Equation 2; $K_i$ are round keys for PRESENT encryption

**Output:** ciphertext

1 randomly generate 16 masks $\mathfrak{m}_0, \mathfrak{m}_1, \ldots, \mathfrak{m}_{15}$

2 **array** of size 16 state $= p \oplus \mathfrak{m}_{15}, \mathfrak{m}_{14}, \ldots, \mathfrak{m}_1, \mathfrak{m}_0$// mask the $j$th nibble of the plaintext with $\mathfrak{m}_j$, each entry of the array is one masked nibble

3 **array** of size 16 masks $= \mathfrak{m}_{15}, \mathfrak{m}_{14}, \ldots, \mathfrak{m}_1, \mathfrak{m}_0$

4 **for** $i = 0$, $i < 31$, $i++$ **do**

5 state = addRoundKey(state, $K_i$)

6 **for** $j = 0$, $j < 16$, $j++$ **do**

// for each nibble

7 ?// masked Sbox computation

8 ?// record the output masks of Sbox computation

9 ?// apply pLayer to the cipher state

10 ?// apply pLayer to the masks

11 state = addRoundKey(state, $K_i$)

12 ?// remove masks from the ciphetext

13 **return** state

*Solution.*

**Algorithm 2:** Masked implementation of PRESENT.

**Input:** $p$, T1, T2, $K_i$ $(i = 1, 2, \ldots, 32)$// $p$ is the plaintext for encryption; T1 is the table for masked Sbox as given in Equation 1; T2 specifies the output mask given the input mask for PRESENT Sbox as defined in Equation 2; $K_i$ are round keys for PRESENT encryption

**Output:** ciphertext

randomly generate 16 masks $\mathfrak{m}_0, \mathfrak{m}_1, \ldots, \mathfrak{m}_{15}$
**array** of size 16 state $= p \oplus \mathfrak{m}_{15}, \mathfrak{m}_{14}, \ldots, \mathfrak{m}_1, \mathfrak{m}_0$// mask the $j$th nibble of the plaintext with $\mathfrak{m}_j$, each entry of the array is one masked nibble
**array** of size 16 masks $= \mathfrak{m}_{15}, \mathfrak{m}_{14}, \ldots, \mathfrak{m}_1, \mathfrak{m}_0$
**for** $i = 0$, $i < 31$, $i++$ **do**
    state = addRoundKey(state, $K_i$)
    **for** $j = 0$, $j < 16$, $j++$ **do**
        // for each nibble
        state[$j$] = T1[state[$j$], masks[$j$]]// masked Sbox computation
        masks[$j$] = T2[masks[$j$]]// record the output masks of Sbox computation
    state = pLayer(state)// apply pLayer to the cipher state
    masks = pLayer(masks)// apply pLayer to the masks
state = addRoundKey(state, $K_i$)
state = state $\oplus$ masks
**return** state

**Question 5.** (10 marks) For an Sbox SB: $\mathbb{F}_2^{\omega_1} \to \mathbb{F}_2^{\omega_2}$, the *(extended) difference distribution table (DDT)* of SB is a 2−dimensional table $T$ of size $(2^{\omega_1}-1)\times 2^{\omega_2}$ such that for any $0 < \delta < 2^{\omega_1}$ and $0 \le \Delta < 2^{\omega_2}$, the entry of $T$ at the $\Delta$th row and $\delta$th column is given by

$$T[\Delta, \delta] = \{\, \boldsymbol{a} \mid \boldsymbol{a} \in \mathbb{F}_2^{\omega_1}, \mathrm{SB}(\boldsymbol{a} \oplus \delta) \oplus \mathrm{SB}(\boldsymbol{a}) = \Delta \,\}.$$

Let SB be the following Sbox

| $x$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| SB($x$) | 4 | 7 | 0 | 5 | 2 | 6 | 3 | 1 |

Define

$$\begin{aligned} f : \mathbb{F}_2^3 &\to \mathbb{F}_2^3 \\ \boldsymbol{x} &\mapsto \mathrm{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \boldsymbol{b}. \end{aligned}$$

We consider a differential fault analysis attack on $f$. Our attack assumption is as follows:

- Fault location: input of $f$
- Fault model: bit flip
- Fault mask: $\varepsilon \in \mathbb{F}_2^4$ s.t. $\boldsymbol{x}' = \boldsymbol{x} \oplus \varepsilon$
- Attacker knowledge: Sbox design, inputs and outputs of $f$, fault mask
- Attacker goal: recover values of $\boldsymbol{a}$ and $\boldsymbol{b}$
- The Attacker can repeat the computation with the same input (not chosen by the attacker)

We know that with input $\boldsymbol{x} = \mathtt{3}$, the correct output is $\mathtt{1}$.

a) (4 marks) Complete the DDT for SB

| $\Delta$ \ $\delta$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | | 46 | 03 | 15 | 27 | | |
| 2 | 67 | 13 | | | 05 | 24 | |
| 3 | 01 | | 47 | 26 | | 35 | |
| 4 | 45 | 02 | | 37 | | | 16 |
| 5 | 23 | | 56 | | 14 | | 07 |
| 6 | | | | 04 | 36 | ? | ? |
| 7 | | 57 | 12 | | | ? | ? |

b) (2 marks) Suppose the attacker injects one fault with fault mask $\varepsilon_1 = \mathtt{2}$ and the resulting output is $\mathtt{5}$. What do we know about $\boldsymbol{a}$?

c) (4 marks) Suppose the attacker injects another fault with fault mask $\varepsilon_2 = \mathtt{3}$ and the resulting output is $\mathtt{0}$. Find the values of $\boldsymbol{a}$ and $\boldsymbol{b}$.

*Solution.*

a) (4 marks)

| $\Delta$ \ $\delta$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | | 46 | 03 | 15 | 27 | | |
| 2 | 67 | 13 | | | 05 | 24 | |
| 3 | 01 | | 47 | 26 | | 35 | |
| 4 | 45 | 02 | | 37 | | | 16 |
| 5 | 23 | | 56 | | 14 | | 07 |
| 6 | | | | 04 | 36 | 17 | 25 |
| 7 | | 57 | 12 | | | 06 | 34 |

b) (2 marks) For a fault mask $\varepsilon$, let $\Delta$ denote the difference between the correct and faulty output, then

$$\begin{aligned}\Delta &= (\mathrm{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \boldsymbol{b}) \oplus (\mathrm{SB}(\boldsymbol{x}' \oplus \boldsymbol{a}) \oplus \boldsymbol{b}) = \mathrm{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \mathrm{SB}(\boldsymbol{x}' \oplus \boldsymbol{a}) \\ &= \mathrm{SB}(\boldsymbol{x} \oplus \boldsymbol{a}) \oplus \mathrm{SB}(\boldsymbol{x} \oplus \boldsymbol{a} \oplus \varepsilon).\end{aligned}$$

We can conclude that he value $\boldsymbol{x} \oplus \boldsymbol{a}$ is in the entry of DDT corresponding to input difference $\delta = \varepsilon$ and output difference $\Delta$.
With fault mask $\mathtt{2}$ and

$$\Delta = \mathtt{1} \oplus \mathtt{5} = \mathtt{4},$$

we know that the value $\boldsymbol{x} \oplus \boldsymbol{a}$ is in the entry of DDT corresponding to input difference $\mathtt{2}$ and output difference $\mathtt{4}$. Thus the possible values of $\boldsymbol{x} \oplus \boldsymbol{a}$ are given by $\mathtt{0}$ and $\mathtt{2}$. Knowing that $\boldsymbol{x} = \mathtt{3}$, the possible values of $\boldsymbol{a}$ are $\mathtt{3}$ and $\mathtt{1}$.

c) (4 marks) Similarly, with fault mask $\mathtt{3}$ and

$$\Delta = \mathtt{1} \oplus \mathtt{0} = \mathtt{1},$$

we know that the value $\boldsymbol{x} \oplus \boldsymbol{a}$ is in the entry of DDT corresponding to input difference $\mathtt{3}$ and output difference $\mathtt{1}$. Thus the possible values of $\boldsymbol{x} \oplus \boldsymbol{a}$ are given by $\mathtt{0}$ and $\mathtt{3}$. Knowing

that $\boldsymbol{x} = \mathtt{3}$, the possible values of $\boldsymbol{a}$ are $\mathtt{3}$ and $\mathtt{0}$. Together with the previous answer, we know that the value of $\boldsymbol{a} = \mathtt{3}$.

We also know that when $\boldsymbol{x} = \mathtt{3}$,

$$\mathrm{SB}(\mathtt{3} \oplus \boldsymbol{a}) \oplus \boldsymbol{b} = \mathtt{1},$$

which gives

$$\boldsymbol{b} = \mathrm{SB}(\mathtt{3} \oplus \mathtt{3}) \oplus \mathtt{1} = \mathrm{SB}(\mathtt{0}) \oplus \mathtt{1} = \mathtt{4} \oplus \mathtt{1} = \mathtt{5}.$$