CHALMERS — GÖTEBORGS UNIVERSITET

# EXAM IN CRYPTOGRAPHY

**TDA352 (Chalmers) - DIT250 (GU)**

**2nd May 2020, 08:30 – 12.30**

Answers must be given in *English* and should be clearly justified.

**Teacher/Examiner:** Katerina Mitrokotsa
**Questions during exam:** Carlo Brunetta, phone 031 772 1619

The exam is divided in four main topics and the total number of points is 50.
The grades are:

| | | | |
|---|---|---|---|
| CTH Grades: | 22-30 $\rightarrow$ 3 | 31-39 $\rightarrow$ 4 | 40-50$\rightarrow$5 |
| GU Grades: | 22-39 $\rightarrow$ G | 40-50 $\rightarrow$ VG | |

## Good luck!

# 1 Symmetric Ciphers (10 p)

(a) You are given the option to employ either one time pad or a stream cipher to guarantee confidentiality. Which of the two would you select? Provide the advantages and disadvantages of each. (2 p)

**Solution:** The one time pad is the only cipher that provides perfect secrecy when $|\mathcal{K}| > |\mathcal{M}|$, where $\mathcal{K}$ denotes the length of the key space and $\mathcal{M}$ denotes the length of the message space. It is very fast (both for encryption and decryption). However, since it requires very long keys at least as long as the messages it is difficult to use in practice.

Stream ciphers are a very good alternative since they are much more practical not requiring a key as long as the message and are very fast (both for encryption). Their main disadvantage is that they do not provide perfect secrecy contrary to OTP. Their security depends on the unpredictability of the employed PRG.

For realistic applications where you need to encrypt multiple and long messages stream ciphers is a better option.

(b) You are given access to a secure PRF $F : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$. Describe how you may build a PRG and a stream cipher based on this PRF. (3 p)

**Solution:** Let $F : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be a secure PRF.

We can define $\mathcal{G} : \mathcal{K} \to \{0,1\}^{nt}$ as:

$$\mathcal{G}(k) = F(k,0)||F(k,1)||F(k,2)||\ldots||F(k,t),$$

then $\mathcal{G}$ is a secure PRG (and it is parallelisable).

Using this PRG we can easily construct a stream cipher with Encryption and Decryption algorithms that work as following:

$\mathsf{Enc}(k,m) = m \oplus \mathcal{G}(k) = c$ and $\mathsf{Dec}(k,c) = c \oplus \mathcal{G}(k) = m$

(c) Is there any stream cipher that provides perfect secrecy? (1 p)

**Solution:** No stream cipher provides perfect secrecy since by definition for no stream cipher it holds: $|\mathcal{K}| > |\mathcal{M}|$, where $\mathcal{K}$ denotes the length of the key space and $\mathcal{M}$.

(d) Consider that you are cryptanalyst working in the Venona project. The language used is the following: $L = \{001000, 100111, 111001, 101000\}$. If you know that for two ciphertexts $c_1$ and $c_2$ generated via the one time pad it holds: $c_1 \oplus c_2 = 101111$, can you identify the original messages $m_1$ and $m_2$? (2 p)

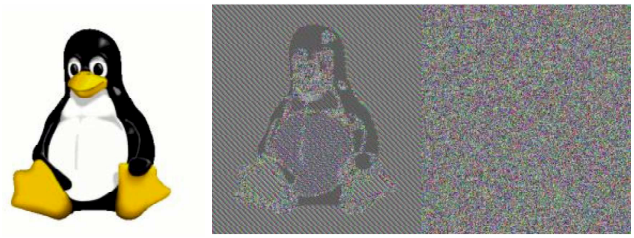
**Solution:** We know that it holds:

$$\begin{aligned} c_1 \oplus c_2 &= (m_1 \oplus k) \oplus (m_2 \oplus k) \\ &= (m_1 \oplus m_2) \oplus (k \oplus k) \\ &= m_1 \oplus m_2 \end{aligned}$$

Since $c_1 \oplus c_2 = 101111$ it holds: $m_1 \oplus m_2 = 101111$. Given the available language $L = \{001000, 100111, 111001, 101000\}$ and that $m_1 \oplus m_2 = 101111$ we can deduce that $m_1 = 001000$ and $m_2 = 100111$.

(e) You are required to encrypt images to guarantee confidentiality. You are given the option to select one of the following two modes of encryption ECB or CBC. Which of the two will you select and for what reason? (2 p)

**Solution:** By the definition of the ECB mode of operation, the encryption is deterministic *i.e.,* equal plaintext blocks will have equal cipher text blocks. Thus, by encrypting an image blocks that have the same pattern coloraturas will produce the same ciphertext. This as can be seen in the following example leaks a lot of information and compromises the confidentiality of the encryption.




However, CBC by definition is not deterministic since the encryption of each block depends on the encryption of the previous block. Thus, CBC is a better and more secure choice.

## 2 Public Key Encryption (13 p)

(a) What is the difference between a probabilistic and a deterministic encryption scheme (2 p).

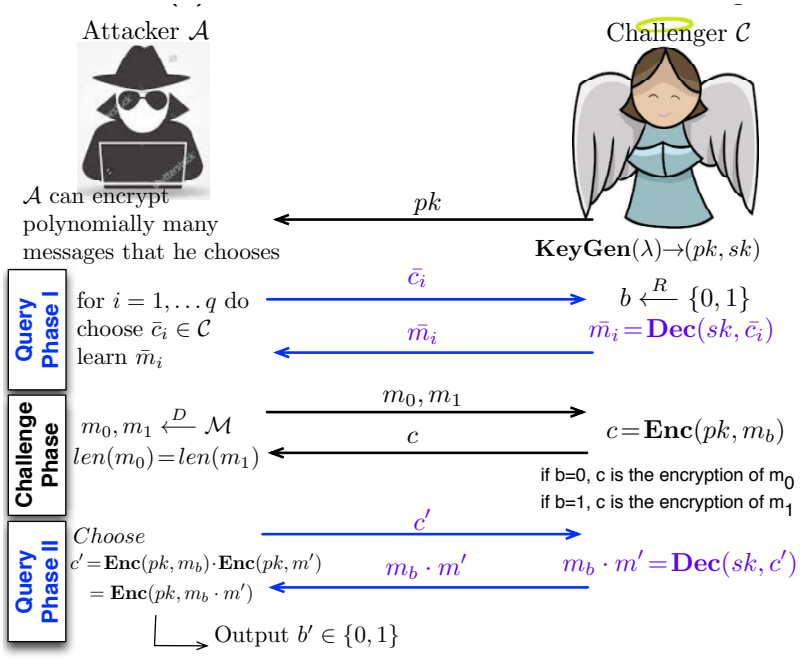
**Solution:** Deterministic is an encryption scheme that for a fixed key and message always generates the same ciphertext. Contrary a probabilistic encryption scheme may produce different ciphertexts when encrypting the same message with the same key.

(b) Prove that textbook RSA is insecure against IND-CCA attacks (4 p).

**Solution:** Textbook RSA has the homomorphic property which implies that it holds:

$$Enc_{RSA}(pk, m_1) \cdot Enc_{RSA}(pk, m_2) = Enc_{RSA}(pk, m_1 \cdot m_2).$$

Thus, we can consider the following game to prove that textbook RSA is insecure against IND-CCA attacks:




What should the attacker do next? He knows $m'$ so he can get $\frac{m_b \cdot m'}{m'} = m_b$!

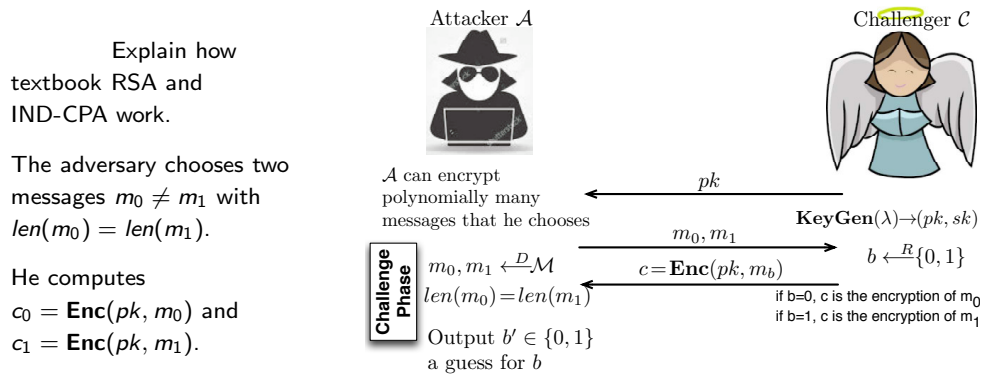
Let $W_0$ be the event that $\mathcal{C}$ chooses $b = 0$ and $\mathcal{A}$ outputs $b' = 0$.
Let $W_1$ be the event that $\mathcal{C}$ chooses $b = 1$ and $\mathcal{A}$ outputs $b' = 0$.

Then we have: $|\mathbf{P}(W_0) - \mathbf{P}(W_1)| = |1 - 0| = 1$

(c) Is textbook RSA secure against IND-CPA attacks? Justify your answer (2 p).

**Solution:** We know that if an encryption scheme that is IND-CCA is also IND- CPA (since IND-CCA contains the definition of IND-CPA). Since we have proven in the previous question that textbook RSA is IND-CCA, we conclude it is also IND-CPA.

Alternatively, we can prove it separately as follows:




Let $W_0$ be the event that $\mathcal{C}$ chooses $b = 0$ and $\mathcal{A}$ outputs $b' = 0$.
Let $W_1$ be the event that $\mathcal{C}$ chooses $b = 1$ and $\mathcal{A}$ outputs $b' = 0$.

$\mathcal{A}$ has the **public key** and the encryption is **deterministic**,
so he knows that for $\mathbf{m_0}$ it holds $\mathbf{c} = \mathbf{c_0}$ and for $\mathbf{m_1}$ it holds $\mathbf{c} = \mathbf{c_1}$.

$\mathcal{A}$ can output $\mathbf{b' = 0}$, when $\mathbf{c} = \mathbf{c_0}$ and $\mathbf{b' = 1}$ when $\mathbf{c} = \mathbf{c_1}$.

When $\mathcal{C}$ chooses $b = 0$, $\mathcal{A}$ outputs $b' = 0$.
When $\mathcal{C}$ chooses $b = 1$, $\mathcal{A}$ outputs $b' = 1$.

Then, we have: $|\mathbf{P(W_0)} - \mathbf{P(W_1)}| = |\mathbf{1} - \mathbf{0}| = \mathbf{1}$

(d) We consider double textbook RSA encryption using a common modulus $N$ and two public keys $e_1$ and $e_2$ with corresponding private keys. Thus, a message $m$ is encrypted first using RSA encryption with the key $e_1$; the result is encrypted again using key $e_2$. Explain why this approach does not increase security. (3 p)

**Solution:** The general argument against double encryption is that it is subject to the *meet-in-the-middle* attack, which has time complexity similar to that of a single brute force attack. In the particular case of RSA encryption, double encryption is also meaningless, since the double encryption is equivalent to the single RSA encryption with public key $e_1e_2$ and private key $d_1d_2$. It is easy to verify this since it holds:

$$(m^{e_1}(modN))^{e_2}(modN) = m^{e_1e_2}(modN)$$

(e) Consider a textbook RSA system with modulus $N = pq$, public key $e$ and private key $d$. Show that if an adversary finds out $\Phi(N)$, she can easily factorise $N$. (2 p)

**Solution:** Recall that $\Phi(N) = (p-1)(q-1) = pq - p - q + 1 = N - p - q + 1$

So if the Adversary knows $\Phi(N)$, he also knows $p + q = N + 1 - \Phi(N)$. But if you know both $p \cdot q = N$ and $p + q = a$ it is easy to compute $p$ and $q$: using $q = N/p$, you know $p + N/p = a$, which gives the ordinary second degree equation $p^2 - ap + N = 0$ to solve for $p$.

# 3 Data Integrity (9 p)

(a) You are given access to an $n$-bit hash function and you attempt to perform a birthday attack. Explain briefly what steps you would follow for a successful attack. What is the level of security provided by the hash function? (2 p)

**Solution:** A birthday attacks attempts to find two different messages $m_1$ and $m_2$, such that $h(m_1) = h(m_2)$ (*i.e.,* a collision), where $h$ is the hash function under consideration. Any hash function is vulnerable to the birthday attack. To perform this attack we may find a collision by generating random messages, hashing them and storing the results in a dictionary, checking each time if the new hash value is already in the dictionary. If the hash values are uniformly distributed $n$-bit strings, one can show that a collision can be expected in $O(\sqrt{2^n}) = O(n^{1/2})$ messages. A $n$-bit hash function since it is vulnerable to a birthday attack that takes $2^{n/2}$ steps is said to offer $n/2$ bits of security.

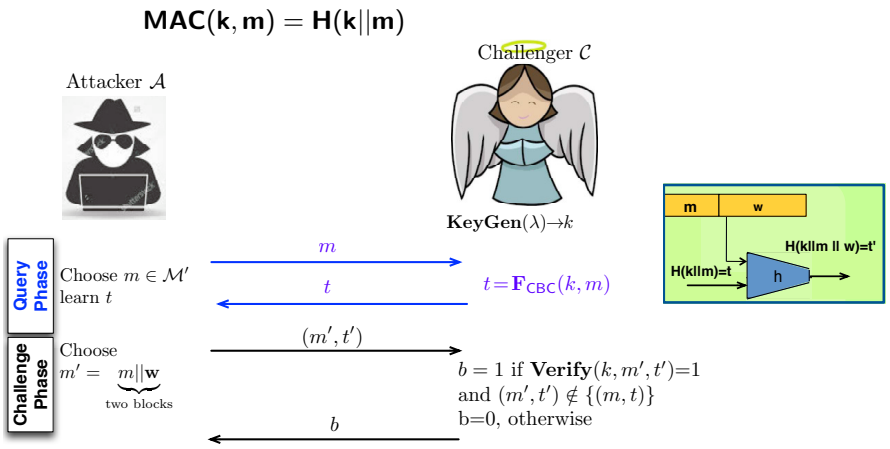
(b) You are asked to build a MAC based on a hash function that relies on the Merkle-Damgård construction. Two possible suggestions are provided for building the MAC

- $MAC(k,m) = H(k||m)$
- $MAC(k,m) = H\big(k \oplus opad || H(k \oplus ipad||m)\big)$, where $opad$ and $ipad$ are fixed constants for H.

Which of the two would you select for your designed MAC considering security aspects (*i.e.,* unforgettability of the resulting MAC)? Justify your answer and prove that one of them is forgeable. (3 p)

**Solution:** The first one is vulnerable to forgeability attacks as can be seen below. The second is the HMAC and has been proven to be secure.

We can easily show that the first MAC construction is forgeable using the following game:




The attacker is able to produce a valid pair $(\mathbf{m}', \mathbf{t}')$ thus $b = 1$. So it holds:

$\mathbf{P}$(Challenger outputs 1) $= \mathbf{1}$

(c) Alice and Bob use digital signatures to sign the transactions they are performing (*e.g.,* Alice paid Bob 100 SEK). However, Alice denies that she has performed a transaction. Can Bob prove that Alice indeed performed the transaction. How do we call this property of digital signatures? (1 p)

**Solution:** Since Alice is signing her transactions, she uses her private key to sign a transaction. She is the only one having access to her private key and thus no one else could have signed it. The property is called *non-repudiation*.

(d) Consider that Alice is using textbook RSA signatures and she wants to sign the hash of her messages/transactions, rather than the messages itself. Why is that? Give two reasons. (3 p)

**Solution:** The main advantages of using the hash and sign paradigm for textbook RSA signatures are: *(i)* The messages become much shorter after being hashed so the signing process is faster and *(ii)* We avoid the forgeability attacks due to the homomorphic property of textbook RSA signatures.

# 4 Cryptographic Protocols (17 p)

1. Let $G$ be a multiplicative group. Show that $G = \mathbb{Z}_{13}^*$ has as generator $g = 6$. (2 p)

   **Solution:** It holds: $6^1 = 6 \pmod{13}$

   $6^2 = 36 = 10 \pmod{13} = -3 \pmod{13}$

   $6^3 = 6 \cdot 6^2 = 6 \cdot (-3) = -18 = 26 - 18 = 8 \pmod{13}$

   $6^4 = 9 \pmod{13}$

   $6^5 = 2 \pmod{13}$

   $6^6 = 12 \pmod{13}$

   $6^7 = 7 \pmod{13}$

   $6^8 = 3 \pmod{13}$

   $6^9 = 5 \pmod{13}$

   $6^{10} = 4 \pmod{13}$

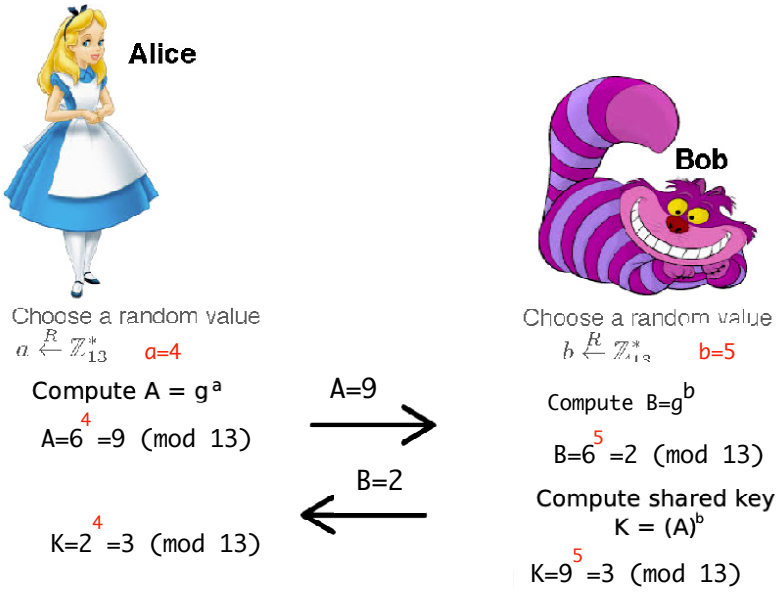
   $6^{11} = 11 \pmod{13}$

   $6^{12} = 1 \pmod{13}$

   $6^{13} = 6 \pmod{13}$. Thus, indeed 6 is the generator of $G = \mathbb{Z}_{13}^* = \{1,2,3,4,5,6,7,8,9,10,11,12\}$.

2. Consider that Alice and Bob are running the Diffie Hellman protocol for the multiplicative group $G = \mathbb{Z}_{13}^*$, with generator $g = 6$. Assume that the secret value of Alice is $a = 4$ and the secret value of Bob is $b = 5$. Show how the DH protocol would be run between Alice and Bob and how would each of them compute the shared key $K$ and the value of the key. (2 p)

   **Solution:**




3. Consider that Eve is playing the role of a "man-in-the-middle" while Alice and Bob are running the DH protocol. Describe concretely how the attack would be performed by Eve and which keys (use concrete values) she would share with Alice and Bob after the completion of the attack. (3 p)

   **Solution:**

4. We consider a protocol where Peggy proves her identity to Victor by giving evidence that she knows a secret $x$.

   The system involves a trusted third party T. Initially, T chooses primes $p$ and $q$ as in RSA and computes $N = p \cdot q$ and a RSA key pair $(e,d)$. $N$ and $e$ are made public and can be used by a whole community of Peggies and Victors. T keeps the private key $d$ for himself. All computations below are in $\mathbb{Z}_N^*$.

   Whenever (a new) Peggy wants to use the system, she chooses a public key $X \in \mathbb{Z}_N^*$ (which could be based on her name, email address etc, using some public way of transforming this to a number in $\mathbb{Z}_N^*$). She sends $X$ to T, who computes Peggy's secret key $x = X^{-d}$ and sends it to her in some secure way. Peggy then announces her public key $X$.

   When Peggy wants to identify herself to Victor, the following protocol is used:

   1. Commitment: Peggy chooses a random $r \in \mathbb{Z}_N^*$, computes $R = r^e$ and sends $R$ to Victor.
   2. Challenge: Victor chooses a random $c$ with $1 \leq c \leq e$ and sends $c$ to Peggy.
   3. Response: Peggy computes $y = r \cdot x^c$ and sends $y$ to Victor.

   Victor now checks that $y \neq 0$ and $R = y^e \cdot X^c$; if this holds he believes that the other party is Peggy.

   (a) Show that a true Peggy, following the protocol, will be identified correctly by Victor. (2 p)

   **Solution:** First we note that, since $x = X^{-d}$, we have $x^{-e} = X^{ed} = X$. Victor will get $y^e \cdot X^c = (r \cdot x^c)^e \cdot (x^{-e})^c = r^e = R$, so he will accept Peggy.

   (b) Is the condition $y \neq 0$ necessary or can we omit it? (1 p)

   **Solution:** Yes this is a necessary condition. If it did not exist a false Peggy could chooses $R = 0$ and $y = 0$.

   (c) Can Victor transfer his knowledge, that indeed Peggy has the secret $x$, to someone else? Explain why. (2 p).

   **Solution:** No! Victor could have produced $(R;c;y)$ by generating $c$ and $y$ at random and computing $R = y^e \cdot X^c$.

5. Consider that we have three parties $P_1, P_2, P_3$ and each of them has a secret value $a = 3$, $b = 4$ and $c = 1$ correspondingly. We are using the secure multi party computation (SMPC) protocol for addition (that we have seen in the lectures) based on Shamir's Secret Sharing Scheme with $t = 2$. Consider that we are in $\mathbb{Z}_{11}$.

   (a) Show how $P_1$, $P_2$ and $P_3$ can compute the sum $s = a + b + c$, without disclosing the values $a$, $b$ and $c$. (3 p)
   *Hint:* Describe how $P_1$, $P_2$ and $P_3$ create their shares and distribute them and how finally the sum is computed.

   **Solution:**

   Since $t = 2$ each of the $P_1$;$P_2$;$P_3$ selects a polynomial of degree 2, the only restrictions is that each if $p_1(x)$ is the polynomial selected by the party $P_1$ then it should hold $p_1(0) = a = 3$. Similarly for the other two polynomials it should hold: $p_2(0) = b = 4$ and $p_3(0) = c = 1$. More precisely, lets assume that $P_1$ selects the polynomial: $p_1(x) = 3 + x + x^2$. Then, we we have:

$$\begin{aligned} p_1(1) &= 3 + 1 + 1 = 5 = a_1 \\ p_1(2) &= 3 + 2 + 2^2 = 9 = a_2 \\ p_1(3) &= 3 + 3 + 3^2 = 15 = a_3 \end{aligned}$$

Lets assume that $P_2$ selects the polynomial $p_2(x) = 4 + x + x^2$. Then, we have:

$$\begin{aligned} p_2(1) &= 4 + 1 + 1 = 6 = b_1 \\ p_2(2) &= 4 + 2 + 2^2 = 10 = b_2 \\ p_2(3) &= 4 + 3 + 3^2 = 16 = b_3 \end{aligned}$$

Lets assume that $P_3$ selects the polynomial $p_3(x) = 1 + x + x^2$. Then, we have:

$$\begin{aligned} p_3(1) &= 1 + 1 + 1 = 3 = c_1 \\ p_3(2) &= 1 + 2 + 2^2 = 7 = c_2 \\ p_3(3) &= 1 + 3 + 3^2 = 13 = c_3 \end{aligned}$$

Then, the shares of the sum $\sigma_1$, $\sigma_2$ and $\sigma_3$ can be calculated as follows:

$$\begin{aligned} \sigma_1 &= a_1 + b_1 + c_1 = 5 + 6 + 3 = 14 \\ \sigma_2 &= a_2 + b_2 + c_2 = 9 + 10 + 7 = 26 \\ \sigma_3 &= a_3 + b_3 + c_3 = 15 + 16 + 13 = 44 \end{aligned}$$

We also have

$$\delta_1(0) = \prod_{j=\{2,3\}, i=1} \frac{j}{j-i} = \frac{2}{1} \cdot \frac{3}{2} = 3$$

$$\delta_2(0) = \prod_{j=\{1,3\}, i=2} \frac{j}{j-i} = \frac{1}{1-2} \cdot \frac{3}{1} = -3$$

$$\delta_3(0) = \prod_{j=\{1,2\}, i=3} \frac{j}{j-i} = \frac{1}{1-3} \cdot \frac{2}{2-3} = \frac{1}{-2} \cdot \frac{2}{-1} = 1$$

Thus, we have:

$$\sigma = \sigma_1 \cdot \delta_1(0) + \sigma_2 \cdot \delta_2(0) + \sigma_3 \cdot \delta_3(0) = 14 \cdot 3 + 26 \cdot (-3) + 44 \cdot 1 = 42 - 78 + 43 = 8$$

(b) Consider that $P_2$ decides not to collaborate with $P_1$ and $P_3$. Can $P_1$ and $P_3$ still compute the sum $s$? If yes, justify why and show how. (2 p)

**Solution:** If $P_2$ will not collaborate with $P_1$ and $P_3$, then $P_1$ and $P_3$ cannot compute the sum $s$. This is because $t = 2$ and thus at least 3 parties are needed to compute $s$.