

## Problems for week 5, Cryptography Course - TDA 352/DIT 250

**General remarks on problems for the weekly problem session:** Exercises will be classified in four different levels:

1. **Easy:** the exercise will require low numerical computations or it can be just a way to look back at the content of the lecture. Exercises of this level should easily be done with just *pen and paper* and are **important to pass** the exam.
2. **Medium:** the exercises will require some time to do (from 5 to 15 minutes each). Maybe a separate paper for some computation is needed! You need to study a bit to answer the questions. These exercises also **may appear** in the exam.
3. **Hard:** the exercises will require you to spend a lot of time doing numerical computations (and we highly recommend using a PC) **or** the questions are real challenge to see if you understood the course in depth. Some of these exercises **may appear** in the exam.
4. **Think:** problems that aim to using your imagination. You are invited to discussion with your colleagues/friends/family and find your best solutions. Generally, the exercises of this level do not take a lot of time in writing the solutions but they will let you think/discuss for (maybe) 30/40 minutes.

**In this weekly exercise sheet:** you will construct some secret sharing schemes, study hash functions and identification protocols.

**Completing the ex. sheet:** you will have a good understanding of hash function's theory, know how to build some secret sharing scheme and have some knowledge on identification protocols.

---

---

### Easy

---

1. Prove that  $g = 6$  is a generator of  $\mathbb{Z}_{41}^*$ .
2. Give the general definition of a Secret Sharing Scheme.
3. State the properties that a Secret Sharing Scheme needs to achieve.
4. Describe the properties of the Fiat-Shamir protocol.
5. Describe the general structure of a  $\Sigma$ -protocol.

---

---

### Medium

---

6. Let us consider the Mignotte's SSS with  $n = 4$  and  $t = 1$ . Let  $m_1 = 3, m_2 = 4, m_3 = 5, m_4 = 7$ . Let the secret be  $s = 9$ .
  - (a) Check that the given values  $m_i$  make a Mignotte's series.
  - (b) Compute the shares for the different parties.
  - (c) Consider the shares  $s_1$  and  $s_4$ . Reconstruct the original secret  $s$  using  $s_1$  and  $s_4$ .
  - (d) Consider the shares  $s_1, s_3$  and  $s_4$ . Reconstruct the original secret  $s$  using  $s_1, s_3$  and  $s_4$ .
7. *You may need a calculator to facilitate the computations.* Let us consider the Mignotte's SSS with  $n = 4$  and  $t = 2$ . Let  $m_1 = 6, m_2 = 11, m_3 = 13, m_4 = 19$ . Let the secret be  $s = 666$ .
  - (a) Compute the shares for the different parties.
  - (b) Consider the shares  $s_1, s_3$  and  $s_4$ . Reconstruct the original secret  $s$ .

8. Let us consider the Shamir SSS with  $n = 2$  and  $t = 1$ . The dealer choose to work in  $\mathbb{Z}_3$ . The secret is  $s = 1$  and the polynomial that he randomly generate is  $f(x) = 1 + 2x$ 
  - (a) Compute the shares for the two different parties.
  - (b) Consider the shares  $s_1$  and  $s_2$ . Reconstruct the original secret  $s$ .
9. Let us consider the Shamir SSS with  $n = 4$  and  $t = 2$ . The dealer chooses to work in  $\mathbb{Z}_7$ . The secret is  $s = 1$  and the polynomial that he randomly generates is  $f(x) = 1 + 3x + 6x^2$ 
  - (a) Compute the shares for the different parties.
  - (b) Consider the shares  $s_1, s_2$  and  $s_3$ . Reconstruct the original secret  $s$ .
10. Suppose that Netflix is using the Fiat-Shamir identification protocol (see slides for lecture 9) to log-in users (in this case the password is a number  $x \in \mathbb{Z}_N$ ). Victor shows you what he claims is a transcript of a protocol run where Peggy has proved her knowledge of the secret  $x$ . However, you do not trust Victor; why will the transcript not convince you?

## Hard

11. Let us consider a Secure Multi Party Computation (SMPC) protocol for addition between 2 parties. Every party will use a Shamir SSS with  $n = 2$  and  $t = 1$ . The parties decide to work in  $\mathbb{Z}_5$ . The secrets are  $s_1 = 1$  and  $s_2 = 2$  and they want to compute the sum of the two values. The polynomials that they randomly generate are  $f_1(x) = 1 + 3x$  for  $P_1$  and  $f_2(x) = 2 + x$  for  $P_2$ .
  - (a) Compute all the different shares, i.e.,  $s_{1,1}, s_{1,2}, s_{2,1}, s_{2,2} \in \mathbb{Z}_5$
  - (b) Compute the partial results  $a_1$  and  $a_2$
  - (c) Compute the final result  $y$  of the multi-party computation, i.e., the sum  $s_1 + s_2 = 3 \pmod{5}$ .
12. Alice and Bob have invented the following protocol for sending a message securely from A to B. The protocol is based on the ideas of the one-time pad, but without a common, shared secret. Instead, for each message, both A and B invent a random nonce and execute the following protocol to send message M from A to B
  1.  $A \rightarrow B$  :  $M_1 = M \oplus N_A$
  2.  $B \rightarrow A$  :  $M_2 = M_1 \oplus N_B$
  3.  $A \rightarrow B$  :  $M_2 \oplus N_A$

Here, we extend our protocol notation to give names to an entire message  $M_1 = \dots$ . Only the message in the right hand side is sent, but the left hand side can be used to refer to the message content in subsequent messages.

  - (a) Show that B can recover M.
  - (b) Is the system secure?
13. Alice and Bob have a shared secret  $k$  and have decided to use it in the following protocol, which enables Alice to identify Bob as the party at the other end.
  1. Alice picks a random bit string  $r$  and sends it as challenge to Bob.
  2. Bob responds with  $r \oplus k$ .

Alice's and Bob's analysis of the protocol is this: The protocol does indeed provide identification, since Alice can check that the sender of message 2 knows  $k$ . It is also secure, since only random numbers are ever sent on the communication channel.

  - (a) How does Alice check that the sender of message 2 knows  $k$ ?
  - (b) Do you agree with Alice and Bob about the security of their protocol? Motivate your answer!
14. State and prove the birthday paradox.

---

---

## Think

---

15. Try to give a formal definition of a cryptographic hash function.  
Two fundamental properties for cryptographic hash functions are the **collision resistance** (find two message with the same hash) and the *first-image resistance* (from a hash, find the original message). We consider a hash function **secure** if the complexity of *breaking* one of the properties is computationally unfeasible.

Is it possible to have an “*ideal perfect*” hash function that does not have any collision, i.e. all the messages have a unique hash value?

*Hint: try to think on the dimension of the message space and the digest space.*

16. An attack that can be done against hash function is to pre-compute a **rainbow table** that consist of a table with the pair: message and the hash value for that message. Generally these tables are optimized for research and retrieve the *inverse* for the hash function. You can easily find online rainbow tables for passwords! We will consider SHA256 as the hash function and we want to compute the dimension of a rainbow table for a fixed length password:

suppose that we have a password of length  $n = 5$ . SHA256 generate a 256-bit digest (32 byte) and a single  $n$ -character long password can be represented with  $n \cdot 8 = 40$ -bit<sup>1</sup>. The number of characters that can be used for the password is 64 (26 lowercase letters, 26 uppercase letters, 10 digits and 2 symbols like ‘.’ and ‘.’)<sup>2</sup>.

- a. How much is the size of the rainbow table in this case where the password length is  $n = 5$ ?

In the last year, NIST **opened to the public** their new draft for security procedures and in this draft there is a clear definition on *good-practices* for generate a secure password.

In this draft, NIST suggested minimal length for password password should be  $n = 8$  character long.

- b. How much is the size of the rainbow table in this case with  $n = 8$  with 64 possible different characters?

Another suggestion is to avoid rules like “*your password must contains at least a symbol, two digit, 3 lowercase letter, a uppercase letter but not in a even position and the length must be equal to 2 modulo 3*”<sup>3</sup> in order to just use the alpha-numeric character-set (so 62 different characters).

- c. How much is the size of the rainbow table in this case with  $n = 8$  with 62 possible different characters?  
d. What is your opinion/idea on password rules in order to achieve a secure password?

---

<sup>1</sup>We suppose you use ASCII 8-bit representation for a single char.

<sup>2</sup>Usually the number of symbols is 10 but it really depends on the implementation.

<sup>3</sup>Impossible password rules **exists**.