Problems for week 6, 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 you to correctly remember the different definitions in a formal way.
2. **Medium**: the exercises will require you to express some objective thoughts on definitions and/or proofs.
3. **Hard**: the exercises will require you to do some computations (and it can take some times) with the main goal of reviewing and use some more complex cryptographic constructions.
4. **Think**: this exercises will not be present in the exam. The goal of these questions is just to challenge yourself in thinking about the entire course and construct a personal and formal opinion about it.

In this weekly exercise sheet: you will go through for all the different topics that we have seen during the course. Almost every question can be present in the exam.
Proposed way to solve the sheet: we encourage you to **first** study the other sheets, the notes that you have taken during lecture, the slides and the reading material.
After that, you should try to solve this exercise sheet with the exam rules (just a calculator, no books or other materials, etc...) but with more time!
Finally, compare your answers with the solutions and the course material.
Completing correctly the sheet in the proposed way: you will have all the knowledge/skills to pass the exam *:)* .

Easy

---

1. **Define** semantic security.
   Define the OTP encryption scheme and prove if it is semantic secure (or not).
2. **Define** what is a PseudoRandom Function (PRF).
   Define Blockcipher.
   Define CBC and ECB mode of operations.
3. **Define** the RSA encryption scheme.
   Define the computational security assumption that we use to prove that RSA is secure.
   Why is it an assumption that the Discrete Logarithm is computationally hard?
   Can you think of other computational assumption?
4. **Define** what is a secure Hash function.
   Where and why should you use such functions in Cryptography?

Medium

---

5. **Define** Unconditional, Computational and Provable security. Give also a description with your own words of the intuitions behind these three notions.
6. **Prove** that ECB is not semantically secure. **Prove** that CBC is semantic secure under the assumption that the blockcipher is semantically secure.
7. **Describe** the (textbook) Diffie Hellman key exchange protocol.
   What are the similarities with the ElGamal public key encryption scheme?

8. **Define** the birthday paradox.
   Explain with your own words, what is the intuition behind the birthday paradox and how it relates to hash functions.

9. Another signature scheme based on discrete logarithms is Schnorr's scheme. The setting is an Abelian group G generated by an element g of prime order q, i.e., G has the q elements $\{g^i | i = 0, 1, \ldots q-1\}$. We also assume that we have agreed on a a hash function $h : \{0,1\}^* \times G \to \mathbb{Z}_q$.

   A user has a secret key $x \in \mathbb{Z}_q^*$ and a corresponding public key $X = g^x$. To sign a message, the signer proceeds as follows:

   (a) Choose a random $y \in \mathbb{Z}_q^*$.
   (b) Compute $Y = g^y$.
   (c) Compute $e = h(m||Y)$.
   (d) Compute $s = y + x \cdot e$ in $\mathbb{Z}_q$.
   (e) The signature is (e, s).

   To verify a signature, the verifier proceeds as follows:

   (a) Compute $r = g^s \cdot X^{-e}$.
   (b) Accept the signature if and only if $e = h(m||r)$.

   Justify this signature scheme, i.e., show that a correctly signed message will be accepted (completeness/correctness property).

Hard

10. In ElGamal encryption scheme, the computation of K^{-1} is often not done using the Extended Euclidean Algorithm but rather using Fermat's little theorem. How? Describe the corresponding method for determining the private key in RSA?

11. Assume that we have five parties $P_1, \cdots, P_5$ and that we tolerate $t = 2$ corrupted parties in a Shamir threshold secret sharing scheme.
    Assume that we work in $\mathbb{Z}_{11}$ and want to share the secret value $s = 6$.

    - Show how we can distribute s among five parties, i.e., compute the shares $s_1, \cdots, s_5$. Each of the shares s_i is sent to the party P_i ($i \in \{1, \cdots, 5\}$)
    - Assume that someone is given the shares s_3, s_4, s_5. Is it possible for her to compute the secret s? Show how.

12. Assume we are using a Mignotte's secret sharing scheme. We have $m_1 = 5, m_2 = 6, m_3 = 7, m_4 = 11, m_5 = 13$.

    - For what threshold t, the given m_i can be used in the Mignotte's secret sharing scheme?
    - Suppose that the threshold is set to $t = 2$. What is the range of possible secrets s?
    - In the case $t = 2$, compute the secret s knowing that $s_1 = 0, s_2 = 1, s_3 = 5$.

13. (a) A web-based auction web-site uses textbook RSA encryption to maintain the secrecy of bids. The site has public RSA key (N, e). For the sake of this problem we make the completely unrealistic assumption that a bid is sent in a message containing only a single integer, representing the bid value. Now, Alice has just made a bid and the adversary Mallory has eavesdropped and heard the ciphertext c. Mallory's main aim is to prevent Alice's bid from winning. Of course, he cannot recover Alice's bid, but makes the guess that her bid is an integer which is a multiple of 10. Show that, if Mallory's guess is right, he can himself make a bid which is 10% higher than Alice's.

    (b) To prevent the attack described in point (a) and other attacks against textbook RSA, messages should be padded using some padding scheme, such as OAEP[1]. Describe some guiding principles in constructing such padding schemes. You do *not* need to describe the actual padding scheme in detail.

[1] You can find a full description of OAEP on the Wikipedia page.

Think

14. **What was** the most interesting concept you encountered during the course?

15. **Starting from** your personal stream cipher and block cipher (built in the first and second exercise sheet), use them as building blocks (*"bricks"*) to construct *"cryptographic walls"*. What walls can you build? Can you prove the security of your *"walls"*?

16. **Do you think** that the law should regulate the role and the usage of cryptography?
You can watch these videos to get inspiration:

    - DEFCON 24 - Nate Cardozo - Crypto State of the Law
    - Keynote NSA - General Alexander - Blackhat USA 2013
    - Could We Ban Encryption? - Computerphile
    - Why would a ban on encryption be a bad thing? - N O D E YouTube Channel

17. Can we hide secrets in software? Can we obfuscate programs, that is, make programs unintelligible while preserving their functionality? What exactly do we mean by *unintelligible*? Why would we even want to do this?