Homework 7

May 5, 2022

You will need 50% of all homework points to qualify for the exam.
You may hand in your solutions in person or by email to kristiinesaarmann@gmail.com. If you submit by email, either scan a handwritten solution or typeset your solution readable. I do not consider ASCII formulas readable.
When submitting, indicate your name and your matriculation number.
The total number of points for each homework is 20 (not including points for bonus problems, if available).
For submitting your solution in a nicely typeset way (e.g., using LaTeX), you get up to 3 bonus points, but not more than 30% of the points you reached for content.

Problem 1: Random oracle model
Write down the definition of IND-CPA security in the random oracle model (for symmetric encryption schemes).

Problem 2: MPC: Insecure E-voting
Your small organization of 21 people has decided to elect a new leader. There are two candidates for the leader, you and Alice. Because your organization is a forward-looking one, you have decided to do digital voting using secure multiparty computation instead of paper-voting. You are using an additively homomorphic IND-CPA public-key encryption scheme, namely, an encryption scheme with the property that there is an easy-to-compute operation $\oplus$ such that $\text{Enc}(pk, m_1) \oplus \text{Enc}(pk, m_2) = \text{Enc}(pk, m_1 + m_2)$ for all plaintexts $m_1, m_2$. (several such encryption schemes exist, for example Paillier cryptosystem or lattice-based fully homomorphic encryption systems)
The voting scheme is the following. There are two somewhat trusted parties, Bob and Charlie. It is assumed that Bob and Charlie will not collude because they hate each other. Also we will assume that they will follow the protocol and cannot be bribed or otherwise influenced.
Charlie will generate a pair of keys $(pk, sk)$ and give everyone the public key $pk$. Now, if a voting party $P_i$ wishes to vote for you, they will compute an encryption of 1, if they want to vote for Alice, they will compute an encryption of 0. That is, $c_i = \text{Enc}(pk, 1)$ if
\( P_i \) wants to vote for you and \( c_i = Enc(pk, 0) \) if \( P_i \) wants to vote for Alice. All the voting parties will send their ciphertexts \( c_i \) to Bob, who will compute \( c := c_1 \oplus c_2 \oplus \cdots \oplus c_{21} \). Bob sends \( c \) to Charlie who will decrypt \( c \) with his secret key and produce \( m := Dec(sk, c) \). Charlie will tell \( m \) to everybody.

1. Assuming everyone is honest, how should we decide the winner? I.e for which values of \( m \) should you be declared the winner and for which values of \( m \) should Alice be declared the winner? Assume that all 21 people vote and that all of them vote for either Alice or you.

2. Now let’s assume you want to cheat to guarantee that you will be the leader. From chat around the office you know that at least 3 other people (i.e not including yourself) will vote for you but no more than 6 other people will. What should you do to guarantee that you win and that people don’t notice that somebody cheated? You can assume that all the other participants (there’s 20 of them) behave honestly, all of them vote and that all of them vote for either Alice or you. (For the sake of clarity - you also have a vote.) You will not be able to post a vote on someone else’s behalf or change other people’s votes. Otherwise you are allowed to not follow the protocol. What should you do?

3. What type of cryptographical primitive could be used to avoid an attack like that? You don’t have to describe the exact details, just mention the type of cryptographic primitive that seems like it could be useful.