Duration: 3 hours.

Structure: 3 questions, 2 items each.

Grading: Each item is worth 17 points. Grades above 100 will be rounded to 100.

Instructions:

• You can use any written materials.
• You can use statements shown in the lectures or home assignments as long as you state them clearly.
• If you don’t know the answer, you can write “I don’t know” and you will get 5/17 points.
• Write in any language you wish, but write clearly.
• Recommendation: each answer (including both items) shouldn’t take much more than a page.
• You don’t need to copy the question into your notebook.
• The questions aren’t ordered according to difficulty. If you get stuck, move on to the next question.

Good Luck!
1. Let \((E, D)\) be a 1-KPA-secure secret-key encryption for messages of length \(n + 1\) (for key length \(n\)).

   (a) Assume that the encryption algorithm \(E\) is deterministic. Prove that the following function \(f\) is one-way or give a counter example:
   \[
   \forall sk \in \{0, 1\}^n : f(sk) = E_{sk}(0^{n+1})
   \]

   (b) Assume that the encryption algorithm \(E\) also uses randomness \(r\) of length \(n\). Prove that the following function \(f\) is one-way or give a counter example:
   \[
   \forall sk, r \in \{0, 1\}^n : f(sk, r) = E_{sk}(0^{n+1}; r)
   \]

2. Let \((G, E, D)\) be a CPA-secure public-key encryption scheme that is (perfectly) correct. For each of the following suggestions, prove that it is a (perfectly) binding and computationally hiding commitment scheme, or give a counter example.

   (a)
   \[
   \text{Com}(m; (r_g, r_e)) = (pk, E_{pk}(m; r_e))
   \]
   where \(m\) is the committed message, \((r_g, r_e)\) are the randomness used by the commitment, each sampled at random and independently from \(\{0, 1\}^n\), \(pk\) is generated by \(G(1^n; r_g)\), with random coins \(r_g\), and \(r_e\) is the randomness used by the encryption algorithm.

   (b)
   \[
   \text{Com}(m; (r_g, r_e)) = E_{pk}(m; r_e)
   \]
   where all parameters are generated as in the previous item.

3. A triangle in a graph consists of three vertices that are all connected to each other by edges. Consider a variant of the GMW zero-knowledge proof system for 3COL where (after the prover commits to a coloring) instead of requesting that the prover opens a random edge, the verifier first flips a random coin \(b \leftarrow \{0, 1\}^n\): if \(b = 0\), or there are no triangles in the graph, the verifier asks that the prover opens a random edge as in the original protocol, whereas if \(b = 1\), and there are triangles, the verifier asks that the prover opens a random triangle. As in the original protocol, the verifier accepts if for every edge that the prover opened, the colors revealed are distinct.

   (a) Is the protocol still zero-knowledge. If your answer is no, give a counter example. If your answer is yes, describe a simulator (no need to prove validity).

   (b) Consider \(t = 20|E|\) sequential repetitions of the above protocol. Show that there exists an efficient extractor algorithm \(E\) such that given every graph \(G = (U, E)\) and the code of a deterministic prover \(P^*\) that with probability \(1/100\) convinces the verifier \(V\) of accepting \(G\), the extractor outputs a valid 3-coloring of \(G\) with probability \(0.99\). The extractor’s running time should be polynomial in \(|G|\) and the worst-case running time \(t\) of the prover \(P^*\).