Homework 4

November 2018

1 Commitment Schemes [50 points]

1) (Definition of a Commitment Scheme, 10 points). What is a commitment scheme? What are the two properties that a commitment scheme must satisfy? Write the formal definition.

2) (Impossibility of Commitment Scheme, 10 points). Provide an informal argument for the fact that a commitment scheme **cannot** be both *statistically* hiding and *statistically* binding.

3) (ElGamal Commitment Scheme, 30 pts). Let $\mathbb{G}$ be a group of order q, with generator g, and assume that the DDH assumption holds in $\mathbb{G}$. Let $h \leftarrow \mathbb{G}$ be an element of $\mathbb{G}$ sampled uniformly at random. $\mathbb{G}, q, g, h$ are publicly known to all parties. Consider the following procedures.

- Commitment Procedure. To commit to a message $m \in \mathbb{Z}_q$, the committer picks a random $u \leftarrow \mathbb{Z}_q$, and compute $(g^u, g^m h^u)$. Let us define $\mathsf{Com}(m, u) = (g^u, g^m h^u)$.
- Opening. To open a commitment, simply reveal (m, u).

This scheme is perfectly binding since there cannot exists $(m, u), (m', u') \leftarrow \mathbb{Z}_q \times \mathbb{Z}_q$ such that $\mathsf{Com}(m, u) = \mathsf{Com}(m', u')$. On the other hand this scheme is computationally hiding. To prove hiding of this scheme we need to use the assumption that DDH assumption is true in $\mathbb{G}$.

1. (Hiding Proof by Reduction, 20 points). Prove hiding of the commitment by showing a reduction to the DDH assumption. Namely, show that: if there exists a PPT adversarial receiver $\mathcal{A}_{\mathsf{hiding}}$ that is able to distinguish commitments of m_0 from commitments of m_1, then this adversary can be used to distinguish a DDH tuple from a random tuple. Recall that the DDH assumption says that given the tuple (g, g^a, g^b, g^c) any polynomial time adversary $\mathcal{A}_{\mathsf{ddh}}$ cannot tell whether $c = ab$ or c is an exponent chosen uniformly at random.

   Note. Your reduction $\mathcal{A}_{\mathsf{ddh}}$ takes in input a tuple (g, g_1, g_2, g_3), nothing else. The goal of the reduction is to use that tuple to generate the commitment for the receiver $\mathcal{A}_{\mathsf{hiding}}$.

   Hint. In the proof, the reduction is allowed to choose all parameters used in the commitment scheme.

   Reduction $\mathcal{A}_{\mathsf{ddh}}(g, g_1, g_2, g_3)$

   (a) ...
   (b) ...
   (c) ...
   (d) ...
   (e) Output

2. (10 points) What happens if the receiver knows $\log_g h$?

2 Zero Knowledge Proofs [50 points]

4) The Guillou-Quisquater identification scheme is directly based on the RSA problem. The identification scheme is a honest verifier zero-knowledge proof that the prover knows x such that $x^e = y \mod n$ where n is an RSA modulus. The **public information** is $\mathsf{pk} = (n, e, y)$ and the **corresponding secret** is x. The protocol is as follows :

1. P chooses $r \leftarrow_\$ \mathbb{Z}_n^*$ and sends $\alpha \leftarrow r^e$ to V
2. V chooses $\beta \leftarrow_\$ \{0, 1\}$ and sends it to P
3. P computes $\gamma \leftarrow rx^\beta$ and sends it to V
4. V accepts the proof if $\gamma^e = \alpha y^\beta$

Completeness To prove that the zero knowledge proof is indeed functional, we need to show that the equation that the verifier checks is indeed correct. Show that the above mentioned zero knowledge proof is indeed complete. **[5 points]**

Soundness To prove soundness of the scheme we want to show that if we have 2 accepting transcripts that have the same first message, then we can extract the secret of the prover. Therefore this is a proof that the prover can convince the verifier only if she knows the secret.

1. How can we obtain 2 accepting transcripts from a prover that have the same first message? Recall, the proof is a mental experiment, so we can execute the prover as many times as we want. **[5 points]**
2. Assume that we obtained 2 accepting transcripts: (α, β, γ) and $(\alpha, \beta', \gamma')$. Show how you can extract the secret x. **[10 points]**

Zero knowledge Show a simulator that can compute an accepting transcript without knowing the secret. Your simulator must run in polynomial time. The input of the simulator is only the theorem $\mathsf{pk} = (n, e, y)$.

3. What is the transcript in this protocol? **[5 points]**
4. Write the simulator : **[20 points]**
   $\underline{\mathsf{Sim}(n, e, y)}$
   (a)
   (b)
   (c)
   (d) Output
5. Argue (informally) that the transcript given in output by the simulator is distributed identically to the real transcript computed via the interaction between prover and verifier. **[5 points]**