CS 276: Homework 8

Due Date: Saturday November 16th, 2024 at 8:59pm via Gradescope

1 A Proof System for Knowledge of a Preimage

We will study a general proof system to prove knowledge of a secret preimage $\mathbf{x}$ of some public output $\mathbf{y}$, for any homomorphic function over a cryptographic group. This protocol generalizes many proof systems, including the Schnorr protocol (that proves knowledge of the discrete log of a group element) and the Chaum-Pedersen protocol (that proves that a given triple of group elements is a DDH triple).

Definitions: Let $\mathbb{G}$ be a cryptographic group of prime order p, where $\frac{1}{p} = \mathsf{negl}(\lambda)$. Let $d_{in}, d_{out} \in \mathbb{N}$ be the dimensions of the input and output spaces, respectively. A function F mapping $\mathbb{Z}_p^{d_{in}} \to \mathbb{G}^{d_{out}}$ is *homomorphic* if for any $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}_p^{d_{in}}$, $F(\mathbf{x} + \mathbf{x}') = F(\mathbf{x}) \cdot F(\mathbf{x}')$. [1]

The proof system will prove knowledge of a secret preimage $\mathbf{x}$ of a public output $\mathbf{y}$. An *instance* of the language L is any tuple $(F, \mathbf{y})$ such that F is a homomorphic function mapping $\mathbb{Z}_p^{d_{in}} \to \mathbb{G}^{d_{out}}$, and $\mathbf{y} \in \mathsf{Im}(F)$. The corresponding *witness* is an input $\mathbf{x} \in \mathbb{Z}_p^{d_{in}}$ such that $F(\mathbf{x}) = \mathbf{y}$.

For example, if we set $F(\mathbf{x}) = g^{\mathbf{x}}$, then we obtain a protocol to prove knowledge of the discrete log $\mathbf{x}$ of a given group element $\mathbf{y}$. This is essentially the Schnorr protocol. If we set $F(\mathbf{x}) = (g^{\mathbf{x}}, h^{\mathbf{x}})$, then we obtain a protocol to prove that $(g^{\mathbf{x}}, h, h^{\mathbf{x}})$ are a DDH triple, which is essentially the Chaum-Pedersen protocol.

The Protocol:

1. P samples $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_p^{d_{in}}$, computes $\mathbf{b} = F(\mathbf{a})$, and sends $\mathbf{b}$ to V.
2. V samples $m \xleftarrow{\$} \mathbb{Z}_p$ and sends m to P.
3. P computes $\mathbf{c} = m \cdot \mathbf{x} + \mathbf{a}$ and sends $\mathbf{c}$ to V.
4. V checks whether
$$F(\mathbf{c}) = \mathbf{y}^m \cdot \mathbf{b}$$
If so, V outputs accept. If not, V outputs reject.

Properties of the Protocol: Let P and V be the honest prover and honest verifier, who must follow the protocol. Let P^* and V^* be a dishonest prover and verifier, who may deviate from the protocol arbitrarily. Next, the *transcript* of the protocol is $(\mathbf{b}, m, \mathbf{c})$, the list of messages sent between the prover and verifier during the protocol.

The protocol should satisfy the following properties:

- *Completeness:* If $\mathbf{y} = F(\mathbf{x})$, then the protocol between $P(F, \mathbf{y}, \mathbf{x})$ and $V(F, \mathbf{y})$ will result in accept with probability 1.

[1] Note that the typical group operation for $\mathbb{Z}_p$ is addition, and the group operation for $\mathbb{G}$ is multiplication, so the homomorphic property simply states that applying the group operation to the inputs before applying F is equivalent to applying the group operation to the outputs after applying F.

- *Knowledge Soundness:* There exists an extractor E that runs in expected polynomial time such that for every F and every $\mathbf{y} \in \mathbb{G}^{d_{out}}$, if $\Pr[\langle P^*, V\rangle(F, \mathbf{y}) \to \mathsf{accept}]$ is non-negligible, then $\Pr[F(\mathbf{x}') = \mathbf{y} : \mathbf{x}' \leftarrow E^{P^*}(F, \mathbf{y})]$ is non-negligible as well.

  The notation $\langle P^*, V\rangle(F, \mathbf{y}) \to \mathsf{accept}$ is the event that the interaction between a dishonest prover P^* and the honest verifier V on inputs $(F, \mathbf{y})$ results in accept. The notation E^{P^*} means that E gets black-box access to P^*, which includes the ability to rewind P^*.

- *Honest-Verifier Zero-Knowledge:* For any valid witness-instance tuple $(\mathbf{x}, \mathbf{y}, F)$, which satisfies $\mathbf{y} = F(\mathbf{x})$, the transcript of the protocol between $P(F, \mathbf{y}, \mathbf{x})$ and $V(F, \mathbf{y})$ can be efficiently simulated given only $(F, \mathbf{y})$.

Question: Prove that the protocol given above satisfies completeness, knowledge soundness, and honest-verifier zero-knowledge. Your proof should not require any computational assumptions.