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:*[2] 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.

Solution This problem comes from Boneh & Shoup, section 19.5.4.

Theorem 1.1 *The protocol satisfies completeness.*

Proof. If $\mathbf{y} = F(\mathbf{x})$, then the protocol will result in accept.
V checks whether:

$$F(\mathbf{c}) = \mathbf{y}^m \cdot \mathbf{b}$$

We know that $\mathbf{y} = F(\mathbf{x})$, $\mathbf{b} = F(\mathbf{a})$, $\mathbf{c} = m \cdot \mathbf{x} + \mathbf{a}$, and F is homomorphic. Then:

$$\begin{aligned} \mathbf{y}^m \cdot \mathbf{b} &= F(\mathbf{x})^m \cdot F(\mathbf{a}) \\ &= F(m \cdot \mathbf{x} + \mathbf{a}) \\ &= F(\mathbf{c}) \end{aligned}$$

Then the verifier's check is equivalent to checking that $F(\mathbf{c}) = F(\mathbf{c})$, which passes with probability 1.

Theorem 1.2 *The protocol satisfies knowledge soundness.*

Proof. The extractor will run P^* on two different challenges, m and m', by rewinding the prover. This gives the extractor two linear equations, which uniquely determine $\mathbf{x}$.
The extractor E^{P^*} is constructed as follows:

1. E runs P^* through one execution of the protocol and plays the role of the verifier. P^* outputs $\mathbf{b}$. Then E samples $m \xleftarrow{\$} \mathbb{Z}_p$ and sends m to P^*. Finally, P^* outputs $\mathbf{c}$.

2. E rewinds P^* to the end of step 1 of the protocol and then runs the rest of the protocol with a freshly random challenge. E samples $m' \xleftarrow{\$} \mathbb{Z}_p$ and sends m' to P^*. Finally, P^* outputs $\mathbf{c}'$.

[2] This definition comes almost verbatim from [Kog19].

3. E checks whether:

$$m \neq m'$$
$$F(\mathbf{c}) = \mathbf{y}^m \cdot \mathbf{b}$$
$$F(\mathbf{c}') = \mathbf{y}^{m'} \cdot \mathbf{b}$$

   If any check fails, then E outputs $\perp$ and aborts. Otherwise, E continues.

4. E computes and outputs:

$$\mathbf{x}' = \frac{\mathbf{c} - \mathbf{c}'}{m - m'}$$

Analysis: If E does not output $\perp$ (i.e. the checks pass), then E will correctly compute an $\mathbf{x}'$ such that $\mathbf{y} = F(\mathbf{x}')$.

$$\begin{aligned}
\mathbf{x}' &= \frac{\mathbf{c} - \mathbf{c}'}{m - m'} \\
\mathbf{x}' \cdot (m - m') + \mathbf{c}' &= \mathbf{c} \\
F\left[\mathbf{x}' \cdot (m - m') + \mathbf{c}'\right] &= F(\mathbf{c}) \\
F(\mathbf{x}')^{(m-m')} \cdot F(\mathbf{c}') &= F(\mathbf{c}) \\
F(\mathbf{x}') &= \left(F(\mathbf{c}) \cdot F(\mathbf{c}')^{-1}\right)^{1/(m-m')} \\
&= \left(\mathbf{y}^m \cdot \mathbf{b} \cdot (\mathbf{y}^{m'} \cdot \mathbf{b})^{-1}\right)^{1/(m-m')} \\
&= \left(\mathbf{y}^{m-m'}\right)^{1/(m-m')} \\
&= \mathbf{y}
\end{aligned}$$

It remains to show that if $\Pr[\langle P^*, V\rangle(F, \mathbf{y}) \to \mathsf{accept}]$ is non-negligible, then $\Pr[E^{P^*}(F, \mathbf{y}) \not\to \perp]$ is non-negligible as well. This is shown in lemma 1.3.

Lemma 1.3 *Let* $\varepsilon = \Pr[\langle P^*, V\rangle(F, \mathbf{y}) \to \mathsf{accept}]$*, and let* ε *be non-negligible. Then*

$$\Pr[E^{P^*}(F, \mathbf{y}) \not\to \perp] \geq \varepsilon^2 - \frac{\varepsilon}{p} = \mathsf{nonnegl}(\lambda)$$

Proof.

1. Let L be a random variable that refers to all of the prover's random coins and the value of $\mathbf{b}$ that they send as the first message in the protocol. We can assume that the value of L is fixed by the time the prover has finished sending their first message. Next, let M and M' be random variables that refer to the m and m' values that the extractor E samples during the two executions of the protocol. Note that M and M' are uniformly random over $\mathbb{Z}_p$, and they are independent of each other and of L. Finally, let $A(L, M) = 1$ if the sampled values of L, M lead the prover to generate an accepting transcript. Then:

$$\Pr[E^{P^*}(F, \mathbf{y}) \not\to \perp] = \Pr_{L,M,M'}[A(L, M) = 1 \wedge A(L, M') = 1 \wedge M \neq M']$$
$$\text{and} \quad \varepsilon = \Pr[\langle P^*, V\rangle(F, \mathbf{y}) \to \mathsf{accept}] = \Pr_{L,M}[A(L, M) = 1]$$

2. Next, for a given value ℓ that L takes, let G_ℓ be the set of all m-values for which $A(\ell, m) = 1$. Then:

$$\begin{aligned}\varepsilon &= \Pr_{L,M}[A(L,M)=1] \\ &= \sum_{\ell} \Pr_{L}[L=\ell] \cdot \Pr_{M}[A(\ell,M)=1] \\ &= \mathbb{E}_L\left[\frac{|G_\ell|}{p}\right]\end{aligned}$$

Furthermore, for a given ℓ,

$$\Pr_{M,M'}[A(\ell,M)=1 \wedge A(\ell,M')=1 \wedge M \neq M'] = \frac{|G_\ell| \cdot (|G_\ell|-1)}{p^2}$$

3. Next,

$$\begin{aligned}\Pr[E^{P^*}(F,\mathbf{y}) \not\to \bot] &= \Pr_{L,M,M'}[A(L,M)=1 \wedge A(L,M')=1 \wedge M \neq M'] \\ &= \sum_{\ell} \Pr_{L}[L=\ell] \cdot \Pr_{M,M'}[A(\ell,M)=1 \wedge A(\ell,M')=1 \wedge M \neq M'] \\ &= \mathbb{E}_L\left[\frac{|G_\ell| \cdot (|G_\ell|-1)}{p^2}\right] \\ &= \mathbb{E}_L\left[\left(\frac{|G_\ell|}{p}\right)^2\right] - \mathbb{E}_L\left[\frac{|G_\ell|}{p^2}\right] \\ &\geq \left(\mathbb{E}_L\left[\frac{|G_\ell|}{p}\right]\right)^2 - \frac{\varepsilon}{p} \\ &= \varepsilon^2 - \frac{\varepsilon}{p}\end{aligned}$$

We used Jensen's inequality to say that $\mathbb{E}_L\left[\left(\frac{|G_\ell|}{p}\right)^2\right] \geq \left(\mathbb{E}_L\left[\frac{|G_\ell|}{p}\right]\right)^2$.

Finally, observe that $\varepsilon^2 - \frac{\varepsilon}{p}$ is non-negligible because ε^2 is non-negligible, and $\frac{\varepsilon}{p}$ is negligible.

Theorem 1.4 *The protocol satisfies honest-verifier zero-knowledge.*

Proof. The simulator $\mathcal{S}$ will sample the transcript variables $(\mathbf{b}, m, \mathbf{c})$ in a different order from the regular protocol. The construction of $\mathcal{S}$ is as follows:

1. $\mathcal{S}$ receives $(F, \mathbf{y})$.
2. $\mathcal{S}$ samples $m \xleftarrow{\$} \mathbb{Z}_p$ and $\mathbf{c} \xleftarrow{\$} \mathbb{Z}_p^{d_{in}}$, and then computes $F(\mathbf{c})$.
3. $\mathcal{S}$ computes

$$\mathbf{b} = F(\mathbf{c}) \cdot \mathbf{y}^{-m}$$

and outputs $(\mathbf{b}, m, \mathbf{c})$.

Analysis: $\mathcal{S}$ samples $(\mathbf{b}, m, \mathbf{c})$ from the same distribution as in the real protocol.

In the real protocol, m is sampled uniformly at random by the honest verifier, and $\mathbf{c}$ is uniformly and independently random due to the randomness of $\mathbf{a}$. Finally, for a given $(\mathbf{y}, m, \mathbf{c})$, $\mathbf{b}$ is the unique value for which $F(\mathbf{c}) = \mathbf{y}^m \cdot \mathbf{b}$.

Similarly, $\mathcal{S}$ chooses m and $\mathbf{c}$ uniformly and independently, and chooses the unique $\mathbf{b}$-value for which $F(\mathbf{c}) = \mathbf{y}^m \cdot \mathbf{b}$. Therefore, $\mathcal{S}$'s output is identically distributed to the transcript in the real protocol.

■