CS 276: Homework 9

Due Date: Friday November 22nd, 2024 at 8:59pm via Gradescope

1 Simulation-Sound NIZKs

We will use the Fiat-Shamir transform to convert the interactive sigma protocol from homework 8 into a non-interactive zero-knowledge proof (NIZK).

We will also define the notion of simulation soundness for NIZKs, which combines soundness and zero-knowledge into one security definition. Simulation soundness essentially states that an adversary who sees simulated proofs of true and false statements of their choosing, cannot produce an accepting proof on a different false statement.

Simulation-sound NIZKs can be used to construct CCA2-secure encryption and signatures, among other applications.

The Fiat-Shamir Transform: Let us start with the sigma protocol from homework 8 and make it non-interactive by computing the verifier's message m with a random oracle $\mathcal{H}$ applied to the partial transcript of the protocol. This is known as the *Fiat-Shamir transform*.

As in homework 8, 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 $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}_p^{d_{in}}$, $F(\mathbf{x} + \mathbf{x}') = F(\mathbf{x}) \cdot F(\mathbf{x}')$. 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 $F(\mathbf{x}) = \mathbf{y}$.

Additionally, let us assume that if we sample $\mathbf{x} \xleftarrow{\$} \mathbb{Z}_p^{d_{in}}$, then $F(\mathbf{x})$ has min-entropy $\omega(\log^2(\lambda))$. In other words, for any $\mathbf{y} \in \mathbb{G}^{d_{out}}$,

$$\Pr_{\mathbf{x} \xleftarrow{\$} \mathbb{Z}_p^{d_{in}}} [F(\mathbf{x}) = \mathbf{y}] \leq 2^{-\omega(\log^2(\lambda))} = \mathsf{negl}(\lambda)$$

Let us also assume that the sigma protocol from homework 8 has **unique responses**. This means that for any $(\mathbf{y}, \mathbf{b}, m)$, there is at most one value of $\mathbf{c}$ for which $F(\mathbf{c}) = \mathbf{y}^m \cdot \mathbf{b}$.[1]

Also, let $\mathcal{H}$ be a random oracle mapping $\mathbb{G}^{d_{out}} \times \mathbb{G}^{d_{out}} \to \mathbb{Z}_p$.

Finally, the NIZK is a pair of algorithms $(\mathsf{Prove}, \mathsf{Verify})$, which are constructed as follows.

$\mathsf{Prove}(\mathbf{x}, \mathbf{y})$:

1. Sample $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_p^{d_{in}}$, and compute $\mathbf{b} = F(\mathbf{a})$.
2. Compute $m = \mathcal{H}(\mathbf{y}, \mathbf{b})$.
3. Compute $\mathbf{c} = m \cdot \mathbf{x} + \mathbf{a}$ and output $\pi = (\mathbf{b}, \mathbf{c})$.

$\mathsf{Verify}(\mathbf{y}, \pi)$:

[1] The unique responses property holds, for instance, when F is injective, and it holds for the Schnorr and Chaum-Pedersen protocols.

1. Compute $m = \mathcal{H}(\mathbf{y}, \mathbf{b})$.
2. If $F(\mathbf{c}) = \mathbf{y}^m \cdot \mathbf{b}$, then output accept. Else output reject.

Zero-Knowledge: Let us define the notion of zero-knowledge for NIZKs.

Definition 1.1 (Zero-Knowledge Adversary and Simulator) *The zero-knowledge adversary $\mathcal{A}$ is run in one of the following games, $\mathcal{G}_{\mathsf{Real}}$ or $\mathcal{G}_{\mathsf{Ideal}}$, and they are not told which one. $\mathcal{A}$ makes proof queries of the form $(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_p^{d_{in}} \times \mathbb{G}^{d_{out}}$, where $F(\mathbf{x}) = \mathbf{y}$, and random oracle queries of the form $(\mathbf{y}, \mathbf{b}) \in \mathbb{G}^{d_{out}} \times \mathbb{G}^{d_{out}}$, and finally they output a bit b in order to guess which game they are in.*

In the real world, $\mathcal{G}_{\mathsf{Real}}$, the challenger samples a random oracle $\mathcal{H}$ and responds to each random oracle query with $\mathcal{H}(\mathbf{y}, \mathbf{b})$. For each proof query $(\mathbf{x}, \mathbf{y})$ such that $F(\mathbf{x}) = \mathbf{y}$, the challenger responds with $\pi = \mathsf{Prove}(\mathbf{x}, \mathbf{y})$.

In the ideal world, $\mathcal{G}_{\mathsf{Ideal}}$, there is a PPT simulator $\mathcal{S}$ that handles the queries. $\mathcal{S}$ receives each random oracle query $(\mathbf{y}, \mathbf{b})$ and computes the response $\mathcal{S}.\mathsf{RO}(\mathbf{y}, \mathbf{b})$. For each proof query, $(\mathbf{x}, \mathbf{y})$ such that $F(\mathbf{x}) = \mathbf{y}$, $\mathcal{S}$ only receives $\mathbf{y}$ and must compute the response $\mathcal{S}.\mathsf{Prove}(\mathbf{y})$.

Definition 1.2 (Zero-Knowledge for NIZKs) *The NIZK satisfies **zero-knowledge** if there exists a PPT simulator $\mathcal{S}$ such that for all PPT adversaries $\mathcal{A}$,*

$$|\Pr[\mathcal{A} \to 1 \text{ in } \mathcal{G}_{\mathsf{Real}}] - \Pr[\mathcal{A} \to 1 \text{ in } \mathcal{G}_{\mathsf{Ideal}}]| = \mathsf{negl}(\lambda)$$

Simulation Soundness: In the definition of zero-knowledge, $\mathcal{S}$ is only required to output an accepting proof for a statement in the language (i.e. an $(\mathbf{x}, \mathbf{y})$ for which $F(\mathbf{x}) = \mathbf{y}$). Simulation soundness allows the adversary to run $\mathcal{S}$ on false statements as well (where $\mathbf{y} \notin \mathsf{Im}(F)$) and guarantees that the adversary cannot forge an accepting proof on a new false statement.

Definition 1.3 (Simulation Soundness Game $\mathcal{G}_{\mathsf{SS}}$) *The simulation soundness adversary $\mathcal{B}$ interacts with $\mathcal{S}$ directly. $\mathcal{B}$ can make proof queries of the form $\mathbf{y} \in \mathbb{G}^{d_{out}}$ and receives the response $\mathcal{S}.\mathsf{Prove}(\mathbf{y})$. $\mathcal{B}$ can also make random oracle queries of the form $(\mathbf{y}, \mathbf{b}) \in \mathbb{G}^{d_{out}} \times \mathbb{G}^{d_{out}}$ and receives the response $\mathcal{S}.\mathsf{RO}(\mathbf{y}, \mathbf{b})$.*

Finally $\mathcal{B}$ outputs a statement-proof tuple $(\mathbf{y}^*, \pi^*)$, which the challenger verifies by computing $\mathsf{Verify}(\mathbf{y}^*, \pi^*)$. If Verify needs to query the random oracle, then the challenger queries $\mathcal{S}.\mathsf{RO}$.

$\mathcal{B}$ wins $\mathcal{G}_{\mathsf{SS}}$ if $(\mathbf{y}^*, \pi^*)$ was not a previous query-response pair for $\mathcal{S}.\mathsf{Prove}$, and $\mathsf{Verify}(\mathbf{y}^*, \pi^*)$ outputs accept, and $\mathbf{y} \notin \mathsf{Im}(F)$ ($\mathbf{y}$ is a false statement).

Definition 1.4 (Simulation Soundness) *A NIZK is simulation-sound if there exists a PPT simulator $\mathcal{S}$ such that the following hold:*

- *Zero Knowledge: For all PPT zero-knowledge adversaries $\mathcal{A}$,*

$$|\Pr[\mathcal{A} \to 1 \text{ in } \mathcal{G}_{\mathsf{Real}}] - \Pr[\mathcal{A} \to 1 \text{ in } \mathcal{G}_{\mathsf{Ideal}}]| = \mathsf{negl}(\lambda)$$

- *Unforgeability: For all PPT simulation soundness adversaries $\mathcal{B}$,*

$$\Pr[\mathcal{B} \text{ wins } \mathcal{G}_{\mathsf{SS}}] = \mathsf{negl}(\lambda)$$

Question: Prove that the NIZK $(\mathsf{Prove}, \mathsf{Verify})$ constructed above satisfies simulation soundness.