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.

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

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 $\pi = (\mathbf{b}, \mathbf{c})$.

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

[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.

Solution This problem is adapted from Boneh & Shoup, exercise 20.22 part a.

Construction of $\mathcal{S}$:

1. $\mathcal{S}$ maintains a database for the random oracle that is initially empty.
2. $\mathcal{S}.\mathsf{RO}(\mathbf{y}, \mathbf{b})$: If $(\mathbf{y}, \mathbf{b}, m)$ appears in the database for some $m \in \mathbb{Z}_p$, then return m. Otherwise, sample $m \xleftarrow{\$} \mathbb{Z}_p$, add $(\mathbf{y}, \mathbf{b}, m)$ to the database, and return m.
3. $\mathcal{S}.\mathsf{Prove}(\mathbf{y})$:
   (a) Sample $m \xleftarrow{\$} \mathbb{Z}_p$ and $\mathbf{c} \xleftarrow{\$} \mathbb{Z}_p^{d_{in}}$.
   (b) Compute $\mathbf{b} = F(\mathbf{c}) \cdot \mathbf{y}^{-m}$.
   (c) If $(\mathbf{y}, \mathbf{b}, m')$ appears in the database, for some $m' \neq m$, then halt and output $\perp$. Else, add $(\mathbf{y}, \mathbf{b}, m)$ to the database, and output $\pi = (\mathbf{b}, \mathbf{c})$.

Lemma 1.5 *The NIZK constructed above satisfies zero-knowledge (def. 1.2) with the simulator $\mathcal{S}$ constructed above.*

Proof.

1. $\mathcal{S}$ correctly simulates the random oracle because on each input $(\mathbf{y}, \mathbf{b})$, the output of $\mathcal{S}.\mathsf{RO}$ is a uniformly random $m \in \mathbb{Z}_p$.
2. In $\mathcal{G}_{\mathsf{Ideal}}$, the probability that $\mathcal{S}$ outputs $\perp$ is negligible. The adversary makes a polynomial number of queries to $\mathcal{S}.\mathsf{RO}$ and $\mathcal{S}.\mathsf{Prove}$, so the size of the database is always polynomial. Next, during each call to $\mathcal{S}.\mathsf{Prove}$, $\mathbf{c}$ is sampled uniformly at random from $\mathbb{Z}_p^{d_{in}}$ and $\mathbf{b}$ is computed as $\mathbf{b} = F(\mathbf{c}) \cdot \mathbf{y}^{-m}$. So $F(\mathbf{c})$ has min entropy $\omega(\log^2(\lambda))$, and $\mathbf{c}$ also has min-entropy $\omega(\log^2(\lambda))$ due to the randomness of $F(\mathbf{c})$. Then the probability that $(\mathbf{y}, \mathbf{b}, *)$ appears in the database is $\leq \mathsf{poly}(\lambda) \cdot 2^{-\omega(\log^2(\lambda))} = \mathsf{negl}(\lambda)$.

   Likewise, in $\mathcal{G}_{\mathsf{Real}}$, the probability is negligible that $\mathsf{Prove}(\mathbf{x}, \mathbf{y})$ outputs a value $\mathbf{b}$, such that $(\mathbf{y}, \mathbf{b})$ has been previously been queried to $\mathcal{H}$. The Prove algorithm samples $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_p^{d_{in}}$ and computes $\mathbf{b} = F(\mathbf{a})$. Then the min-entropy of $\mathbf{b}$ is $\omega(\log^2(\lambda))$, so the probability that $(\mathbf{y}, \mathbf{b})$ was previously queried to $\mathcal{H}$ is $\leq \mathsf{poly}(\lambda) \cdot 2^{-\omega(\log^2(\lambda))} = \mathsf{negl}(\lambda)$.
3. On any proof query $(\mathbf{x}, \mathbf{y})$ (where $F(\mathbf{x}) = \mathbf{y}$), the output distributions of $\mathcal{S}.\mathsf{Prove}(\mathbf{y})$ and $\mathsf{Prove}(\mathbf{x}, \mathbf{y})$ are statistically close.

   In $\mathcal{G}_{\mathsf{Ideal}}$, let us condition on the event that $\mathcal{S}.\mathsf{Prove}(\mathbf{y})$ never outputs $\perp$. Since this occurs with overwhelming probability, this changes the output distribution of the proof queries negligibly. Next, the values $(\mathbf{b}, \mathcal{S}.\mathsf{RO}(\mathbf{y}, \mathbf{b}), \mathbf{c})$ have the following distribution: $\mathbf{c}$ and $\mathcal{S}.\mathsf{RO}(\mathbf{y}, \mathbf{b})$ are uniformly and independently random. Finally, $\mathbf{b}$ is the unique value for which:

$$F(\mathbf{c}) = \mathbf{y}^{\mathcal{S}.\mathsf{RO}(\mathbf{y}, \mathbf{b})} \cdot \mathbf{b}$$

In $\mathcal{G}_{\mathsf{Real}}$, let us condition on the event that $\mathsf{Prove}(\mathbf{x}, \mathbf{y})$ outputs a value $\mathbf{b}$, such that $(\mathbf{y}, \mathbf{b})$ has not previously been queried to $\mathcal{H}$ by the adversary. Then the values $(\mathbf{b}, \mathcal{H}(\mathbf{y}, \mathbf{b}), \mathbf{c})$ have the following distribution: $\mathbf{c}$ is uniformly random due to the randomness of $\mathbf{a}$. $\mathcal{H}(\mathbf{y}, \mathbf{b})$ is uniformly and independently random. Finally, $\mathbf{b}$ is the unique value for which

$$F(\mathbf{c}) = \mathbf{y}^{\mathcal{H}(\mathbf{y},\mathbf{b})} \cdot \mathbf{b}$$

The distribution of $(\mathbf{b}, \mathcal{H}(\mathbf{y}, \mathbf{b}), \mathbf{c})$ in $\mathcal{G}_{\mathsf{Real}}$ is the same as the distribution of $(\mathbf{b}, \mathcal{S}.\mathsf{RO}(\mathbf{y}, \mathbf{b}), \mathbf{c})$ in $\mathcal{G}_{\mathsf{Ideal}}$.

4. In summary, the adversary's view in $\mathcal{G}_{\mathsf{Real}}$ and $\mathcal{G}_{\mathsf{Ideal}}$ are statistically close, so

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

Then the NIZK satisfies zero-knowledge with the simulator $\mathcal{S}$.

Lemma 1.6 *For all PPT simulation soundness adversaries $\mathcal{B}$,* $\Pr[\mathcal{B}$ *wins* $\mathcal{G}_{\mathsf{SS}}] = \mathsf{negl}(\lambda)$.

Proof. We will reduce to the soundness of the interactive sigma protocol from homework 8.

1. We showed in HW 8 that the sigma protocol satisfies **knowledge soundness**, which says that for any $\mathbf{y}$, if there exists a cheating prover P^* that can convince the verifier to accept $\mathbf{y}$ with non-negligible probability, then we can extract from this prover – with non-negligible probability – a witness $\mathbf{x}$ that satisfies $F(\mathbf{x}) = \mathbf{y}$.

   Knowledge soundness implies **(regular) soundness**, which is defined as follows. The sigma protocol satisfies regular soundness if for any given value $\mathbf{y} \notin \mathsf{Im}(F)$, no cheating prover can convince the honest verifier to accept $\mathbf{y}$, except with negligible probability.

   If the protocol did not satisfy regular soundness, then there would exist a cheating prover that could convince the verifier to accept $\mathbf{y} \notin \mathsf{Im}(F)$ with non-negligible probability. Then we could use our knowledge soundness extractor to extract a value $\mathbf{x}$ such that $F(\mathbf{x}) = \mathbf{y}$. However, this is impossible because $\mathbf{y} \notin \mathsf{Im}(F)$, so no such $\mathbf{x}$ exists. Therefore, the protocol must satisfy regular soundness in addition to knowledge soundness.

2. Next, if there exists an adversary $\mathcal{A}_{\mathsf{SS}}$ that wins the simulation soundness game $\mathcal{G}_{\mathsf{SS}}$ with non-negligible probability, then we can construct an adversary $\mathcal{A}_{\Sigma}$ that breaks the soundness of the sigma protocol from homework 8.

   Construction of $\mathcal{A}_{\Sigma}$:

   (a) Let $q = \mathsf{poly}(\lambda)$ be an upper bound on the number of queries that $\mathcal{A}_{\mathsf{SS}}$ makes to $\mathcal{S}.\mathsf{RO}$ during $\mathcal{G}_{\mathsf{SS}}$. Sample a query $i \xleftarrow{\$} [q+1]$.

   (b) Run $\mathcal{S}$ and $\mathcal{A}_{\mathsf{SS}}$ in a simulation of $\mathcal{G}_{\mathsf{SS}}$, and allow $\mathcal{A}_{\mathsf{SS}}$ to query $\mathcal{S}.\mathsf{Prove}$ and $\mathcal{S}.\mathsf{RO}$. On the i-th query to $\mathcal{S}.\mathsf{RO}$, which has input $(\mathbf{y}_i, \mathbf{b}_i)$, $\mathcal{A}_{\Sigma}$ sends $(\mathbf{y}_i, \mathbf{b}_i)$ to the sigma protocol's verifier and receives a uniformly random $m_i \in \mathbb{Z}_p$. Then $\mathcal{A}_{\Sigma}$ adds $(\mathbf{y}_i, \mathbf{b}_i, m_i)$ to the database and responds to the $\mathcal{S}.\mathsf{RO}$ query with m_i.

(c) At the end of the simulation of $\mathcal{G}_{\mathsf{SS}}$, $\mathcal{A}_{\mathsf{SS}}$ outputs $\mathbf{y}^*$ and $\pi^* = (\mathbf{b}^*, \mathbf{c}^*)$. $\mathcal{A}_\Sigma$ queries $\mathcal{S}.\mathsf{RO}(\mathbf{y}^*, \mathbf{b}^*)$ to obtain m^*.

(d) If $\mathcal{A}_{\mathsf{SS}}$'s output satisfies $\mathbf{y}^* = \mathbf{y}_i$ and $\mathbf{b}^* = \mathbf{b}_i$, then $\mathcal{A}_\Sigma$ sends $\mathbf{c}^*$ to the verifier.

3. Let us consider the case where $\mathcal{A}_{\mathsf{SS}}$ wins $\mathcal{G}_{\mathsf{SS}}$, and its output satisfies $\mathbf{y}^* = \mathbf{y}_i$ and $\mathbf{b}^* = \mathbf{b}_i$. Then $\mathcal{A}_\Sigma$ will convince the sigma protocol's verifier to accept because $\mathcal{S}.\mathsf{RO}(\mathbf{y}_i, \mathbf{b}_i)$ equals the m_i chosen by the verifier, and

$$F(\mathbf{c}^*) = (\mathbf{y}^*)^{\mathcal{S}.\mathsf{RO}(\mathbf{y}^*, \mathbf{b}^*)} \cdot \mathbf{b}^* = \mathbf{y}_i^{m_i} \cdot \mathbf{b}_i$$

4. Furthermore, by the end of $\mathcal{A}_\Sigma$'s execution, the database contains the entry $(\mathbf{y}^*, \mathbf{b}^*, m^*)$ because $\mathcal{A}_\Sigma$ queries $\mathcal{S}.\mathsf{RO}(\mathbf{y}^*, \mathbf{b}^*)$ and obtains m^*.

5. Next, $(\mathbf{y}^*, \mathbf{b}^*, m^*)$ was first added to the database on a call to $\mathcal{S}.\mathsf{RO}$, not a call to $\mathcal{S}.\mathsf{Prove}$ because otherwise $\mathcal{A}_{\mathsf{SS}}$ would not have won $\mathcal{G}_{\mathsf{SS}}$. If $(\mathbf{y}^*, \mathbf{b}^*, m^*)$ were first added to the database on a query to $\mathcal{S}.\mathsf{Prove}$, then that query returned $\pi^* = (\mathbf{b}^*, \mathbf{c}')$ where $\mathbf{c}'$ is the unique value for which $F(\mathbf{c}') = (\mathbf{y}^*)^{m^*} \cdot \mathbf{b}^*$. If $\mathcal{A}_{\mathsf{SS}}$'s final output – $\mathbf{y}^*, \pi^* = (\mathbf{b}^*, \mathbf{c}^*)$ – is accepted by the verifier, then $\mathbf{c}^* = \mathbf{c}'$, so $(\mathbf{y}^*, \pi^*)$ was previously generated on a query to $\mathcal{S}.\mathsf{Prove}$. Then $\mathcal{A}_{\mathsf{SS}}$ will not win $\mathcal{G}_{\mathsf{SS}}$.

   Therefore, if $\mathcal{A}_{\mathsf{SS}}$ wins $\mathcal{G}_{\mathsf{SS}}$, then $(\mathbf{y}^*, \mathbf{b}^*, m^*)$ was first added to the database on a call to $\mathcal{S}.\mathsf{RO}$.

6. Next, given that $\mathcal{A}_{\mathsf{SS}}$ wins $\mathcal{G}_{\mathsf{SS}}$, the probability that $(\mathbf{y}^*, \mathbf{b}^*, m^*)$ was first added to the database on the i-th query to $\mathcal{S}.\mathsf{RO}$ is $\frac{1}{q+1} = \mathsf{nonnegl}(\lambda)$. This is because i is uniformly random and independent of $\mathcal{A}_{\mathsf{SS}}$'s view. The verifier samples $m_i \xleftarrow{\$} \mathbb{Z}_p$, and for every $j \neq i$, $\mathcal{S}.\mathsf{RO}$ also samples m_j uniformly from $\mathbb{Z}_p$, so $\mathcal{A}_{\mathsf{SS}}$'s view is the same for any value of i that $\mathcal{A}_\Sigma$ chose.

7. In summary, with non-negligible probability, $\mathcal{A}_{\mathsf{SS}}$ wins $\mathcal{G}_{\mathsf{SS}}$, and $(\mathbf{y}^*, \mathbf{b}^*, m^*)$ was first added to the database on the i-th query to $\mathcal{S}.\mathsf{RO}$. In this case, $\mathcal{A}_\Sigma$ convinces the verifier to accept a false statement, which violates soundness.

■