Notes for Lecture 9

1 Zero Knowledge

Consider an NP statement x like "a graph has a Hamiltonian cycle". Suppose Alice has a witness for x and she wants to prove to Bob that the graph does indeed have a Hamiltonian cycle, but she doesn't want Bob to learn any of Hamiltonian cycles.

We'll have two requirements for this setting.

- **Completeness:** If A is honest, then B will accept
- **Soundness:** If x is false, B will reject.

The ideal security notion is zero knowledge, which roughly means that B learns nothing.

We'll consider a non-interactive version of this. Let's suppose there's some common reference string (CRS) in the sky. For now, we won't say this reference string is random. In this protocol, we have an algorithm $\mathsf{Prove}(x, w, \mathsf{CRS})$ (playing the role of Alice) that sends a proof π to the verifier (Bob). The verifier $\mathsf{Ver}(x, \pi, \mathsf{CRS}) \to 1/0$ decides whether or not to accept the proof. This format of non-interactive proof is called non-interactive zero knowledge (NIZK).

- **Completeness:** If w is a valid witness, then

$$\mathsf{Ver}(x, \mathsf{Prove}(x, w, \mathsf{CRS}), \mathsf{CRS}) = 1.$$

- **Soundness:** If x is false, then for all cheating provers Prove', we have

$$Pr[\mathsf{Ver}(x, \mathsf{Prove}'(x, \mathsf{CRS}), \mathsf{CRS}) = 1] < negl.$$

The statistical definition of soundness interprets "for all cheating provers" to mean this holds for any Prove'. The computational definition interprets this to hold for all PPT Prove'.

With these definitions, zero knowledge means that there exists S such that

$$(\mathsf{CRS} \leftarrow \mathsf{Setup}(), \mathsf{Prove}(x, w, \mathsf{CRS})) \approx (\mathsf{CRS}, \pi) \leftarrow S(x)$$

At first glance, this definition appears contradictory. There shouldn't be a way for $S(x)$ to generate a π on the righthand side. However, the catch is that the CRS on the right is not necessarily generated properly. Soundness only holds for CRS generated honestly.

We give the following construction.

- $\mathsf{Setup}()$. First, generate a random PRF key $k \leftarrow \{0,1\}^\lambda$. Define $P_k(x, w)$ to work as follows.

  check if w is a valid witness for x
  if not, abort.
  otherwise, output $PRF_k(x)$.

  Define $V_k(x, \pi)$ to work as follows.

  check if $OWF(\pi) = OWF(PRF_k(x))$
  if so, output 1
  if not, output 0

  Let $\hat{P} = iO(P_k), \hat{V} = iO(V_k)$, and output $\mathsf{CRS} = (\hat{P}, \hat{V})$.

- $\mathsf{Prove}(x, w, \mathsf{CRS})$. Output $\pi = \hat{P}(x, w)$.

- $\mathsf{Ver}(x, \pi, \mathsf{CRS})$. Output $\hat{V}(x, \pi)$

Theorem 1. *This construction satisfies completeness.*

$\mathsf{Ver}(x, \mathsf{Prove}(x, w, \mathsf{CRS}), \mathsf{CRS})$ is $\mathsf{Ver}(x, PRF_k(x), (\hat{P}, \hat{V}))$ which will clearly evaluate to 1.

Theorem 2. *This construction satisfies soundness.*

Proof. As usual, we prove this with a sequence of hybrids.

Hybrid 0. This hybrid corresponds to the honestly generated CRS.

Hybrid 1. let x^* be a false NP statement. Define $P'_{k\{x^*\}}(x, w)$ as follows.

if $x = x^*$, abort.
else if w is invalid, abort.
else output $PRF(k\{x^*\}, x)$

This hybrid is the same as Hybrid 0, except we switch to $P'_{k\{x^*\}}$. In Hybrid 0, the program aborts on false inputs x, so the input output behavior is not changed by the introduction of the first line. By iO security, Hybrid 1 is indistinguishable from Hybrid 0.

Hybrid 2. Let $y^* \leftarrow OWF(PRF(k, x^*))$, and define $V'_{k\{x^*\},y^*}(x, \pi)$ as follows.

```
if x = x*
    if OWF(π) = y*
        output 1
    else output 0
else if OWF(π) = OWF(PRF(k{x*}, x))
    output 1
else output 0
```

This hybrid is the same as Hybrid 1, except we switch to this $V'_{k\{x^*\},y^*}(x, \pi)$. The programs again have the same input output behavior, so by iO they are indistinguishable.

Hybrid 3. Pick a random r^*, and make $y^* \leftarrow OWF(r^*)$. This is indistinguishable by punctured pseudorandom function security.

If the prover can win here, we have a cheating prover that finds π such that $OWF(\pi) = y^*$. But since y^* is the output of a one way function, this means the cheating prover is inverting a one way function. □

Theorem 3. *This construction is zero knowledge.*

Proof. We construct a simulator S that outputs a simulated (CRS, π). S picks a random k and generates $\hat{P}, \hat{V}$ the same way $\mathsf{Setup}()$ does, which gives the simulated CRS. The simulated π is just $PRF_k(x)$. This is indistinguishable from the honestly generated $(\mathsf{CRS}, \mathsf{Prove}(x, w, \mathsf{CRS}))$ via pseudorandom function security. □

[$BP'14$] show that iO = OWF gives NIZKs in the common *random* string model. Note that the common reference string here was clearly not random.

Noninteractive Witness Indistinguishability (NIWI) can be done in the standard model without any CRS. In this problem, completeness and soundness are defined the same, but for security, we want that for any two valid witnesses, w_0, w_1, we have $\mathsf{Prove}(x, w_0) \approx_c \mathsf{Prove}(x, w_1)$. Zero knowledge implies witness indistinguishability.