Notes for Lecture 20

Today we'll talk about the limits of obfuscation. We'll see how we formalize arguments about what obfuscation cannot do. We'll survey results and techniques but we won't go into great detail about proofs. We'll look at collision resistance, $\mathsf{NP} \cap \mathsf{coNP}$, and statistical zero knowledge (SZK).

We would ideally like to show statements of the form "object A does not imply object B". Unfortunately, we probably can't do this. This statement logically means object A exists and object B doesn't, but it turns out that we can't prove things like this without also showing $\mathsf{Pol} \neq \mathsf{NP}$. The other problem is that for most of these cryptographic objects, we believe they exist but that you won't get it from obfuscation and one way functions.

1 Black Box Separation

The idea here is the rule out "natural" constructions. More specifically, this means black box constructions.

As an example, we can show that one way functions do not imply collision resistance. (Again, this isn't exactly the theorem that is proved. What is proved is a black box separation.)

We'll imagine we live in a world with an oracle in the sky that can return truly random outputs when queried. Alice, Bob, and an adversary all have access to this oracle. This immediately gives a one way function, since Alice can query the oracle O on some input x, and the adversary will only have a negligible probability of guessing x given $O(x)$.

We denote by H^O a circuit with AND, OR, and NOT gates in addition to special oracle gates that can query this truly random oracle. We create a second oracle ColFind that takes as input a circuit that makes oracle queries and finds a collision. Every party will have access to ColFind.

We have to be careful, however, to ensure that ColFind doesn't leak any information about the random oracle. To ensure this, it finds collisions as follows: it chooses x_0 at random, and then chooses a random x_1 that makes $H^O(x_0) = H^O(x_1)$.

In this world, we don't have collision resistant hash functions, since we can break all of them with ColFind. But it turns out we have one way functions, because it turns out the random oracle one way function still works (though we won't prove this).

2 Black Box vs. iO

Applying this black box separation technique doesn't seem to work with iO because applications of iO are not black box. This is because iO takes circuits as input.

It turns out it's not too hard to come up with an approach that gets around this. We will instead think about iO for oracle-aided circuits. So we consider oracles such as a one way function $G(r)$ and $PRF(k, \cdot)$. The adversary has access to these oracles but doesn't know what we query them on.

Here, iO just means that if we have $C_1^O \equiv C_2^O$, then their obfuscations are indistinguishable.

We can come up with a world where we have this definition of iO and one-way functions, but we don't have collision resistance.

Obfuscation is done slightly differently. The obfuscator Obf takes in the circuit and some randomness r and produces a string $L \leftarrow \mathsf{Obf}(C^O, r)$. In order to makes sense of L, we additionally have an evaluation oracle $\mathsf{Eval}(L, x)$ that inverts the obfuscation to recover C^O and then runs it on x and returns the output.

So the adversary has access to Eval and Obf and O. it turns out that with some effort we can find a collision finding oracle that doesn't break Eval, Obf, and O.

With iO and one way functions, we can get PKE, multiparty NIKE, ORE, and a few other cryptographic tools. So we know immediately that these tools cannot imply collision resistant hash functions, since it would contradict our result here that iO and one way functions cannot give collision resistant hash functions.

In addition, FHE and PIR (private information retrieval) do imply CRHF, so we won't be able to construct these from iO and OWFs in a black box way.

3 Statistical Zero Knowledge

Statistical zero knowledge is similar to regular zero knowledge, except we change the requirement so that even an unbounded time verifier cannot distinguish the simulated view from the real view.

We are given two circuits, C_0, C_1. We have the promise that the statistical distance SD is either small or large: $SD(C_0(r), C_1(r)) \in [0, 1/3] \cup [2/3, 1]$. The goal is to figure out which is the case.

Fact: Most tools in crypto imply SZK hardness. Multilinear maps, LWE, and factoring all imply this.

Theorem 1. *There are no black box constructions of hard problems in SZK from* iO *and OWF.*

Here we're not trying to *build* anything. We just need to come up with a hard problem.

So we have an SZKSolver that takes in O, Obf, Eval and C_0, C_1. We want to solve statistical zero knowledge but we don't want to break any of O, Obf, Eval.

Our first attempt is to compute $\delta = SD(C_0, C_1)$. We just have our oracle run in possibly exponential time and just try to compute the statistical distance. If $\delta < 1/2$, output 0 and if $\delta > 1/2$, output 1.

The problem with this is this breaks the OWF. The problem is that if we're very slightly under $1/2$ and then we change the distribution just slightly, we'll detect that we have gone over $1/2$. If we want to invert r, then we can set up C_0, C_1 such that $\delta < 1/2$ if O has a preimage of r and $\delta \geq 1/2$ otherwise.

Our second try is that if C_0 and C_1 don't obey promise, reject. The problem is that we have a sharp threshold at $1/3$ and $2/3$, and based on that behavior we can still attack the OWF in a similar fashion.

The correct thing to do is to choose a random $t \in [1/3, 2/3]$. We output 0 if $\delta < t$ and 1 if $\delta \geq t$. This random threshold is picked for every random pair of circuits C_0, C_1, which rules out the possibility of binary searching for t.

So we can see that this doesn't trivially break the one way function. If we work harder we can show that this doesn't break Obf.

So we have a world with obfuscation and one way functions, but we don't have hard problems because our SZK solver solves them.

Again, since we can get PKE, NIKE, ORE, PPAD hardness from iO + OWF, these cannot imply SZK hardness.

At the end of class, we noted that a lot of crypto tools imply hardness in $\mathsf{NP} \cap \mathsf{coNP}$.