Notes for Lecture 27 (draft)

Summary

In this lecture we begin the construction and analysis of a zero-knowledge protocol for the 3-coloring problem. Via reductions, this extends to a protocol for any problem in NP. We will only be able to establish a weak form of zero knowledge, called "computational zero knowledge" in which the output of the simulator and the interaction in the protocol are computationally indistinguishable (instead of identical). It is considered unlikely that NP-complete problem can have zero-knowledge protocols of the strong type we defined in the previous lectures.

As a first step, we will introduce the notion of a *commitment scheme* and provide a construction based on any one-way permutation.

1 Commitment Scheme

A commitment scheme is a two-phase protocol between a *Sender* and a *Receiver*. The Sender holds a message m and, in the first phase, it picks a random key K and then "encodes" the message using the key and sends the encoding (a *commitment* to m) to the Receiver. In the second phase, the Sender sends the key K to the Receiver can *open* the commitment and find out the content of the message m.

A commitment scheme should satisfy two security properties:

- **Hiding.** Receiving a commitment to a message m should give no information to the Receiver about m;
- **Binding.** The Sender cannot "cheat" in the second phase and send a different key K' that causes the commitment to open to a different message m'.

It is impossible to satisfy both properties against computationally unbounded adversaries. It is possible, however, to have schemes in which the Hiding property holds against computationally unbounded Receivers and the Binding property holds (under appropriate assumptions on the primitive used in the construction) for bounded-complexity Senders; and it is possible to have schemes in which the Hiding property

holds (under assumptions) for bounded-complexity Receivers while the Binding property holds against any Sender. We shall describe a protocol of the second type, based on one-way permutations. The following definition applies to one-round implementations of each phase, although a more general definition could be given in which each phase is allowed to involve multiple interactions.

Definition 1 (Computationally Hiding, Perfectly Binding, Commitment Scheme) *A Perfectly Binding and (t, ϵ)-Hiding Commitment Scheme for messages of length ℓ is a pair of algorithms (C, O) such that*

- **Correctness.** *For every message m and key K,*

$$O(K, C(K, m)) = m$$

- (t, ϵ)**-Hiding.** *For every two messages $m, m' \in \{0,1\}^\ell$, the distributions $C(K, m)$ and $C(K, m')$ are (t, ϵ)-indistinguishable, where K is a random key, that is, for every algorithm A of complexity $\leq t$,*

$$|\,\mathbb{P}[A(C(K, m)) = 1] - \mathbb{P}[A(C(K, m')) = 1]| \leq \epsilon$$

- **Perfectly Binding.** *For every message m and every two keys K, K',*

$$O(K', C(K, m)) \in \{m, FAIL\}$$

In the following we shall refer to such a scheme (C, O) as simply a (t, ϵ)-secure commitment scheme.

Given a one-way permutation $f : \{0,1\}^n \to \{0,1\}^n$ and a hard-core predicate P, we consider the following construction of a one-bit commitment scheme:

- $C(K, m) := f(K), m \oplus P(K)$
- $O(K, (c_1, c_2))$ equals $FAIL$ if $f(K) \neq c_1$, and $P(K) \oplus c_2$ otherwise.

Theorem 2 *If P is a (t, ϵ)-secure hard core predicate for f, then the above construction is a $(t - O(1), 2\epsilon)$-secure commitment scheme.*

There is a generic way to turn a one-bit commitment scheme into a commitment scheme for messages of length ℓ (just concatenate the commitments of each bit of the message, using independent keys).

Theorem 3 *Let (O, C) be a (t, ϵ)-secure commitment scheme for messages of length k such that $O(\cdot, \cdot)$ is computable in time r. Then the following scheme $(\overline{C}, \overline{O})$ is a $t - O(r \cdot \ell), \epsilon \cdot l)$-secure commitment scheme for message of length $k \cdot \ell$:*

- $\overline{C}(K_1, \ldots, K_\ell, m) := C(K_1, m_1), \ldots, C(K_\ell, m_\ell)$
- $\overline{O}(K_1, \ldots, K_\ell, c_1, \ldots, c_\ell)$ *equals $FAIL$ if at least one of $O(K_i, c_i)$ outputs $FAIL$; otherwise it equals $O(K_1, c_1), \ldots, O(K_\ell, c_\ell)$.*

There is also a construction based on one-way permutations that is better in terms of key length.

2 A Protocol for 3-Coloring

We assume we have a (t, ϵ)-secure commitment scheme (C, O) for messages in the set $\{1, 2, 3\}$.

The prover P takes in input a 3-coloring graph $G = ([n], E)$ (we assume that the set of vertices is the set $\{1, \ldots, n\}$ and use the notation $[n] := \{1, \ldots, n\}$) and a proper 3-coloring $\alpha : [n] \to \{1, 2, 3\}$ of G (that is, α is such that for every edge $(u, v) \in E$ we have $\alpha(u) \neq \alpha(v)$). The verifier V takes in input G. The protocol, in which the prover attempts to convince the verifier that the graph is 3-colorable, proceeds as follows:

- The prover picks a random permutation $\pi : \{1, 2, 3\} \to \{1, 2, 3\}$ of the set of colors, and defines the 3-coloring $\beta(v) := \pi(\alpha(v))$. The prover picks n keys $K_1, \ldots, K_n$ for (C, O), constructs the commitments $c_v := C(K_v, \beta(v))$ and sends $(c_1, \ldots, c_n)$ to the verifier;
- The verifier picks an edge $(u, v) \in E$ uniformly at random, and sends (u, v) to the prover;
- The prover sends back the keys K_u, K_v;
- If $O(K_u, c_u)$ and $O(K_v, c_v)$ are the same color, or if at least one of them is equal to $FAIL$, then the verifier rejects, otherwise it accepts

Theorem 4 *The protocol is complete and it has soundness error at most* $(1 - 1/|E|)$.

Repeating the protocol k times sequentially reduces the soundness error to $(1 - 1/|E|)^k$; after about $27 \cdot |E|$ repetitions the error is at most about 2^{-40}.

3 Simulability

We now describe, for every verifier algorithm V^*, a simulator S^* of the interaction between V^* and the prover algorithm.

The basic simulator is as follows:

Algorithm S^*_{1round}

- Input: graph $G = ([n], E)$
- Pick random coloring $\gamma : [n] \to \{1, 2, 3\}$.
- Pick n random keys $K_1, \ldots, K_n$
- Define the commitments $c_i := C(K_i, \gamma(i))$
- Let (u, v) be the 2nd-round output of V^* given G as input and $c_1, \ldots, c_n$ as first-round message
- If $\gamma(u) = \gamma(v)$, then output FAIL
- Else output $((c_1, \ldots, c_n), (u, v), (K_u, K_v))$

And the procedure $S^*(G)$ simply repeats $S^*_{1round}(G)$ until it provides an output different from $FAIL$.

It is easy to see that the output distribution of $S^*(G)$ is always *different* from the actual distribution of interactions between P and V^*: in the former, the first round is almost always a commitment to an invalid 3-coloring, in the latter, the first round is always a valid 3-coloring.

We shall prove, however, that the output of $S^*(G)$ and the actual interaction of P and V^* have *computationally indistinguishable* distributions provided that the running time of V^* is bounded and that the security of (C, O) is strong enough.

For now, we prove that $S^*(G)$ has efficiency comparable to V^* provided that security of (C, O) is strong enough.

Theorem 5 *Suppose that (C, O) is $(t+O(nr), \epsilon/(n \cdot |E|))$-secure and C is computable in time $\leq r$ and that V^* is a verifier algorithm of complexity $\leq t$.*

Then the algorithm S^*_{1round} as defined above has probability at most $\frac{1}{3} + \epsilon$ of outputting $FAIL$.

The proof of Theorem 5 relies on the following result.

Lemma 6 *Fix a graph G and a verifier algorithm V^* of complexity $\leq t$.*

Define $p(u, v, \alpha)$ to be the probability that V^* asks the edge (u, v) at the second round in an interaction in which the input graph is G and the first round is a commitment to the coloring α.

Suppose that (C, O) is $(t + O(nr), \epsilon/n)$-secure, and C is computable in time $\leq r$.

Then for every two colorings α, β and every edge (u, v) we have

$$|p(u, v, \alpha) - p(u, v, \beta)| \leq \epsilon$$