Lecture Notes 19

47 A Zero-Knowledge Interactive Proof for Graph 3-Coloring

The goal of the next couple of lectures is to show that every language in $\mathcal{NP}$ has a zero knowledge interactive proof. We begin with the graph 3-colorability problem.

47.1 Graph 3-colorability

**Definition:** Let $G = (V, E)$ be a simple graph. A 3-coloring of $G$ is a function $\psi : V \rightarrow \{1, 2, 3\}$ such that for all $(u, v) \in E$, $\psi(u) \neq \psi(v)$.

That is, each node is labeled with one of three colors such that no edge connects two nodes of the same color.

**Definition:** A graph $G$ is 3-colorable if there is a 3-coloring of $G$. The language $G3C$ is the set of 3-colorable graphs.

**Fact** $G3C$ is $\mathcal{NP}$-complete.

47.2 The protocol

The protocol makes use of a commitment scheme. For now, assume a family of functions $\{C_s | s \in \{0, 1\}^n\}_{n \in \mathbb{N}}$, where $C_s(\sigma) \in \{0, 1\}^*$ for each $s \in \{0, 1\}^n$ and $\sigma \in \{1, 2, 3\}$. $C_s(\sigma)$ is said to be the commitment of the sender using coins $s$ to the value $\sigma$. $C_s(\sigma)$ can be computed in polynomial time given $s$ and $\sigma$. We desire that the commitment scheme satisfy two properties:

- **Secrecy** The commitment $C_s(\sigma)$ to $\sigma$ reveals a negligible amount of information about $\sigma$. In other words, the receiver of the commitment cannot distinguish commitments to any of the three colors with non-negligible advantage over random guessing.

- **Unambiguity** If $C_s(\sigma) = C_{s'}(\sigma')$, then $\sigma' = \sigma$. In other words, given a string $c$, there is at most one $\sigma$ for which it is a valid commitment.

Formal properties and construction of more general commitment schemes are given in section 48.

The interactive proof for $G3C$ is given in Figure 47.1.

**Explanation.** In step 1 of Figure 47.1, the prover randomly permutes the colors in the 3-coloring $\psi$ to produce a new 3-coloring $\phi$ of $G$. It commits to each color $\phi(v)$ for $v \in V$ with the commitment sequence $\bar{c}$ and sends $\bar{c}$. The verifier checks that $\phi$ is a 3-coloring by asking the prover to reveal the colors at the two endpoints of a randomly chosen edge $(u, v)$. The prover does so in step 3. In step 4, the verifier checks that the colors at $u$ and $v$ were revealed correctly and that they are different.

If both $P$ and $V$ follow this protocol, $V$ always accepts, establishing completeness. If $G$ is not 3-colorable, then any 3-coloring $\phi$ committed to by a cheating prover $P^*$ in step 1 will have at least
Common input: Simple graph $G = (V, E)$, where $V = \{1, \ldots, n\}$.

<table>
<thead>
<tr>
<th>Prover $P$</th>
<th>Verifier $V$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Private input: 3-coloring $\psi$ of $G$.</td>
<td></td>
</tr>
<tr>
<td>1. Random permutation $\pi$ over ${1, 2, 3}$. $\phi = \pi \circ \psi$ is also 3-coloring of $G$.</td>
<td></td>
</tr>
<tr>
<td>Random $s_1, \ldots, s_n \in {0, 1}^n$. Compute $c_i = C_{s_i}(\phi(v)) \forall v \in V$. $\bar{c} = (c_1, \ldots, c_n)$. $\bar{c}$</td>
<td></td>
</tr>
<tr>
<td>2. $\overset{(u,v)}{\leftarrow}$ Random $(u, v) \in E$.</td>
<td></td>
</tr>
<tr>
<td>3. $r_u = (s_u, \phi(u)), r_v = (s_v, \phi(v))$.</td>
<td></td>
</tr>
<tr>
<td>4. Let $\hat{s}_u, \hat{\sigma}_u = r_u$. Let $\hat{s}_v, \hat{\sigma}<em>v = r_v$. Check $c_u = C</em>{\hat{s}_u}(\hat{\sigma}<em>u)$. Check $c_v = C</em>{\hat{s}_v}(\hat{\sigma}_v)$. Check $\hat{\sigma}_u \neq \hat{\sigma}_v$. Accept iff all checks succeed.</td>
<td></td>
</tr>
</tbody>
</table>

Figure 47.1: Interactive proof for graph 3-colorability.

one edge whose endpoints are colored the same. With probability $1/|E|$, $V$ chooses this edge in step 2. Whatever values $P^*$ sends in step 3 will fail $V$’s one of $V$’s checks, either on correctly opening $c_u$ or $c_v$, or it will finds that $u$ and $v$ are colored the same. Hence, $V$ will reject with probability at least $1/|V|$.

The construction of the simulator $M^*$ to show that this protocol is zero knowledge is deferred to the next lecture.

### 48 Bit-Commitment Schemes

A bit-commitment scheme is a pair of probabilistic polynomial-time interactive Turing machines $(S, R)$ called the sender and receiver, respectively. The common input is a security parameter $1^n$. The sender’s private input is a bit $v$. The sender’s commitment to $v$ is the receiver’s view $(r, \bar{m})$ of its interaction with $S$, where $r$ is the receiver’s random coins and $\bar{m}$ is the sequence of messages received from $S$.

Fix $n$ and let $\sigma \in \{0, 1\}$. We say a receiver view $(r, \bar{m})$ is a possible $\sigma$-commitment if, for some string $s$, $\bar{m}$ describes the messages received by $R$ when $R$ uses local coins $r$, $S$ uses local coins $s$, and $S$ has private input $\sigma$. The view is ambiguous if it is both a possible 0-commitment and a possible 1-commitment.

Here are the requirements for the commit phase of a bit-commitment scheme:

**Input specification** The common input is a security parameter $1^n$. The sender’s private input is a bit $v$.

**Secrecy** For all probabilistic polynomial-time interactive Turing machines $R^*$ interacting with $S$,
the probability ensembles
\[\{(S(0), R^*)(1^n)\}_{n \in \mathbb{N}} \quad \text{and} \quad \{(S(1), R^*)(1^n)\}_{n \in \mathbb{N}}\]

are computationally indistinguishable. The notation \(S(v), R^*(x)\) as used here means the random variable describing the receiver’s view in a joint computation of \(S\) and \(R^*\) on common input \(x\), where \(S\) has private input \(v\). (Recall the definition of computational indistinguishability in section 26 of lecture notes 10.)

**Unambuguity** For all but a negligible fraction of the receiver’s local coins \(r\), there is no sequence of sender messages \(\tilde{m}\) for which the receiver’s view \((r, \tilde{m})\) is ambiguous.

In the **reveal phase**, the sender opens the commitment \((r, \tilde{m})\) by revealing the secret bit \(v\) and the sequence \(s\) of local coins that it used during the commit phase. Upon receiving \((v, s)\), the receiver re-executes the joint computation of the commit phase, simulating \(S(v)\) using local coins \(s\), and simulating \(R\) with local coins \(r\). It then checks that the sequence of messages \(\tilde{m}'\) sent by \(S\) in the simulation matches the sequence \(\tilde{m}\) from the commitment and accepts iff they agree.

### 48.1 Commitment based on a one-way permutation

Let \(f : \{0, 1\}^* \to \{0, 1\}^*\) be a one-way permutation, and let \(b : \{0, 1\}^* \to \{0, 1\}\) be a hard core predicate for \(f\). A commitment scheme is easily derived from \(f\) and \(b\).

**Commit phase** Let \(1^n\) be the common input and \(v\) the sender’s private input. The sender chooses a uniformly distributed binary string \(s\) of length \(n\) and sends a single message \(m = C_s(v) = (f(s), b(s) \oplus v)\) to the receiver. The receiver does nothing during the commit phase (and hence uses no local coins). The sender’s commitment to \(v\) is just \(m\).

**Reveal phase** To open \(m\), the sender sends the pair \((v, s)\). The receiver checks that \(m = C_s(v)\).

Unambiguity is immediate since \(f\) is a permutation. Hence, if \(m = (y, \tau)\) for some string \(y\) and \(\tau \in \{0, 1\}\), then \(m\) is a commitment only to the value \(v = b(s) \oplus \tau\), where \(s = f^{-1}(y)\) is the unique inverse of \(y\) under \(f\).

Secrecy follows from the fact that \(b\) is a hard-core predicate for \(f\). Here’s a sketch of the proof of secrecy.

Suppose some probabilistic polynomial-time algorithm \(D(m)\) is able to distinguish commitments to 0 from commitments to 1 with non-negligible probability \(\epsilon(n)\). Formally

\[|\Pr[D(f(U_n), b(U_n) \oplus 1) = 1] - \Pr[D(f(U_n), b(U_n)) = 1]| \geq \epsilon(n),\]

where \(U_n\) is a uniformly distributed random variable over \(\{0, 1\}^n\). Without loss of generality, we may assume that the output of \(D\) is either 0 or 1, and we may drop the absolute value brackets and assume that

\[\Pr[D(f(U_n), b(U_n) \oplus 1) = 1] - \Pr[D(f(U_n), b(U_n)) = 1] \geq \epsilon(n).\]

We construct an algorithm \(A'\) that on input \(y = f(s)\) correctly outputs \(b(s)\) with non-negligible advantage \(\epsilon'(n)\) over random guessing. Formally,

\[\Pr[A'(f(U_n)) = b(U_n)] \geq \frac{1}{2} + \epsilon'(n)\]
$A'(y)$ chooses $\tau \in \{0, 1\}$ uniformly at random, constructs $m = (y, \tau)$, computes $\sigma = D(m)$ and outputs $\sigma \oplus \tau$.

From the proof of unambiguity above, $m = (y, \tau)$ is a commitment to $v = \tau \oplus b(s)$, where $s = f^{-1}(y)$. Hence, $b(s) = \tau \oplus v$. Thus, if $m$ is a commitment to $v$ and $D(m)$ outputs $v$, then $A'(y)$ correctly outputs $b(s)$. Moreover, because $\tau$ is chosen at random, $m$ is equally likely to be a commitment to 0 or a commitment to 1.

We leave to the reader the task of showing that $A'(f(s))$ has an $e'(n)$ advantage at guessing $b(s)$ for some non-negligible function $e'(n)$. This contradicts the assumption that $b$ is hard-core for $f$. Hence, the assumed distinguisher $D$ does not exist and the commit phase satisfies the secrecy condition.

### 48.2 Commitment based on a pseudorandom generator

Although the commitment scheme of section 48.1 is simple, it assumes the existence of one-way permutations. This is a possibly stronger assumption than the existence of one-way functions, for the problem of constructing a one-way permutation assuming only the existence of one-way functions is still open. However, it is known that pseudorandom generators can be constructed assuming only the existence of one-way functions. We now construct a bit-commitment scheme based on a pseudorandom generator, showing that commitment schemes exist if one-way functions exist.

Let $G(s)$ be a pseudorandom generator with expansion factor $\ell(n) = 3n$. (See section 29 of lecture notes 12)

**Commit phase** Let $1^n$ be the common input and $v$ the sender’s private input. The receiver chooses $r \in \{0, 1\}^{3n}$ uniformly at random and sends $r$ to the sender. The sender chooses $s \in \{0, 1\}^n$ uniformly at random, computes

$$m = \begin{cases} G(s) & \text{if } v = 0 \\ G(s) \oplus r & \text{if } v = 1 \end{cases}$$

and sends $m$ to the receiver. The sender’s commitment to $v$ is the receiver view $(r, m)$.

**Reveal phase** To open $(r, m)$, the sender sends the pair $(v, s)$. The receiver checks that either $v = 0$ and $m = G(s)$ or $v = 1$ and $m = G(s) \oplus r$.

The proof of the secrecy condition is another reducibility argument. Assuming there is a distinguisher between commitments to 0 and commitments to 1, one constructs a distinguisher between $G(U_n)$ and $U_{3n}$, contradicting the assumption that $G$ is a pseudorandom generator. Details are in the textbook.

The proof of unambiguity is more interesting. This commitment scheme does not have perfect unambiguity. For example, if $r = 0$, then the receiver view $(r, G(s))$ is a commitment to both 0 and 1. More generally, if there exist $s_0, s_1$ such that $G(s_0) = G(s_1) \oplus r$, then the receiver view $(r, G(s_0)) = (r, G(s_1) \oplus r)$ is ambiguous. Otherwise, $(r, m)$ is unambiguous for all receiver views $(r, m)$.

Call a value $r$ bad if $r = G(s_0) \oplus G(s_1)$ for some $s_0, s_1$ and good otherwise. There are $(2^n)^2 = 2^{2n}$ pairs $(s_0, s_1)$, where $s_0, s_1 \in \{0, 1\}^n$, and each of them gives rise to one bad value $r = G(s_0) \oplus G(s_1)$. All of the other $2^{3n}$ possible values for $r$ are good. Hence, the probability of the receiver choosing a bad $r$ is exponentially small – only $2^{2n}/2^{3n} = 1/2^n$, which is a negligible function.