1 Definitions of Zero Knowledge

Last class we showed a zero knowledge proof for Graph-Isomorphism. Now we will show that there exist zero knowledge proofs for every language in $NP$. Logically, this implies that anything that you can prove in a classical manner you can also prove in zero knowledge.

Previously, we defined perfect zero knowledge, which is a very strong form. Today, we relax that definition as follows.

**Definition 1 (Zero Knowledge)** Let $(P, V)$ be a interactive proof for $L \in NP$, with witness relation $R_L$. $(P, V)$ is zero knowledge if for all probabilistic polynomial time machines $V^*$ there exists an expected PPT $S$ such that for all nonuniform PPT $D$ there exists a negligible function $\varepsilon$ such that $\forall x \in L, w \in R_L(x), z \in \{0, 1\}^*$, $D$ distinguishes the following distributions with probability $\varepsilon(|x|)$:

$$\{\text{View}_{V^*}[P(x, w) \Leftrightarrow V^*(x, z)], \{S(x, z)\}\}.$$

Perfect zero knowledge is exactly the same except that it requires the two distributions to be identical rather than simply indistinguishable.

An alternative definition is to replace $\text{VIEW}_{V^*}$ with $\text{OUTPUT}_{V^*}$. The two definitions are equivalent, since the output is included in the view and since $V^*$ could simply output its view.

There is also a stronger notion of zero knowledge known as black-box zero knowledge.

**Definition 2 (Zero Knowledge)** Let $(P, V)$ be a interactive proof for $L \in NP$, with witness relation $R_L$. $(P, V)$ is zero knowledge if there exists an expected PPT $S$ such that for all probabilistic polynomial time machines $V^*$ and for all nonuniform PPT $D$ there exists a negligible function $\varepsilon$ such that $\forall x \in L, w \in R_L(x), z \in \{0, 1\}^*, r \in \{0, 1\}^*$, $D$ distinguishes the following distributions with probability $\varepsilon(|x|)$:

$$\{\text{View}_{V^*}[P(x, w) \Leftrightarrow V^*(x, z)], \{S^{V^*}(x, z)\}\}.$$

2 Commitment Schemes

We want to show how to construct zero-knowledge proofs for a large class of languages; in order to do so, we need to introduce a new class of cryptographic primitives known as commitment schemes. A commitment can be thought of as the digital equivalent of a
physical locked box. It consists of a two-phase interactive protocol between two parties \( S, R \); in the commit phase, the sender commits to a value \( v \) (puts it in a locked box) and in the reveal phase the sender reveals the value of \( v \). An observer should not be able to determine the value \( v \) from the commitment (when it is in the locked box), and the sender should only be able to reveal one value when it opens the box.

**Definition 3 (Commitment Scheme)** 
\( \text{Com} \) is a commitment scheme if \( \text{Com} \) is polynomial time and there exists a polynomial \( \ell \) such that the following two properties hold:

1. **Hiding:** For every nonuniform PPT \( D \) there exists a negligible function \( \varepsilon \) such that for all \( n \in \mathbb{N} \), \( v_0, v_1 \in \{0, 1\}^n \), \( D \) distinguishes the following distributions with probability at most \( \varepsilon(n) \):
   \[
   \{ r \leftarrow \{0, 1\}^{\ell(n)} : \text{Com}(v_0, r) \}, \{ r \leftarrow \{0, 1\}^{\ell(n)} : \text{Comm}(v_1, r) \}.
   \]

2. **Binding:** For all \( v_0, v_1 \in \{0, 1\}^n \), \( r_0, r_1 \in \{0, 1\}^{\ell(n)} \), if \( v_0 \neq v_1 \) then \( \text{Com}(v_0, r_0) \neq \text{Com}(v_1, r_1) \).

Commitment schemes can be constructed from OWPs (or OWFs):

**Lemma 4** If one-way permutation exist, then there exist (perfectly binding) commitment schemes.

**Proof.** We begin by constructing a single-bit commitment scheme. Let \( f \) be the assumed one-way permutation, and let \( h \) be a hard-core predicate for \( f \). We define a commitment scheme by:

\[
\text{Com}(b; r) = (f(r), h(r) \oplus b).
\]

To decommit, the sender reveals \( r \).

Binding follows immediately; given a commitment \( (x, y) \), since \( f \) is a one-way permutation there exists a unique string \( r \) such that \( f(r) = x \), therefore there exists a unique \( b \) such that \( h(r) \oplus y = b \).

Hiding follows from the assumption that \( h \) is a hard-core predicate: assume for contradiction that there exists a n.u. PPT \( D \) and a polynomial \( p \), such that for infinitely many \( n \in \mathbb{N} \), \( D \) distinguishes \( r \leftarrow \{0, 1\}^n : (f(r), h(r) \oplus 0) \) and \( r \leftarrow \{0, 1\}^n : (f(r), h(r) \oplus 1) \) w.p. \( 1/p(n) \). By the prediction lemma, there exist a machine \( A \) such that

\[
\Pr[m \leftarrow 0, 1, r \leftarrow \{0, 1\}^n : A(f(r), h(r) \oplus m) = m] \geq \frac{1}{2} + \frac{1}{2p(n)}.
\]

We can now use \( A \) to construct a machine \( A_0 \) that predicts the hard-core predicate \( h \): \( A_0 \) on input \( (f(r), y) \) picks \( c \leftarrow 0, 1 \), computes \( m = A(f(r), (y \oplus c)) \), and outputs \( c \oplus m \).
Observe that,

\[ Pr[r \leftarrow 0, 1^n : A_0(f(r)) = h(r)] = Pr[r \leftarrow 0, 1^n : c \leftarrow 0, 1 : A(f(r), c) \oplus c = h(r)] = Pr[r \leftarrow 0, 1^n : m \leftarrow 0, 1 : A(f(r), h \oplus b(r)) = m] \geq \frac{1}{2} + \frac{1}{2p(n)}. \]

Therefore the proposed 1-bit commitment scheme is satisfies the hiding property.

To commit to an arbitrary value \( v \in \{0, 1\}^n \), simply use the 1-bit commitment scheme to commit to each bit. Binding is again immediate, and hiding follows from the hybrid lemma.

### 3 \( NP \subseteq ZK \)

Having constructed commitment schemes, it is possible to prove the existence of zero-knowledge proofs for general classes of problems, specifically for any language in \( NP \).

**Theorem 5** If one-way functions exist, then every language in \( NP \) has a zero-knowledge proof.

For simplicity, we will show how to prove this result based on one-way permutations (the protocol is simpler – three rounds instead of four rounds).

**Theorem 6** If one-way permutations exist, then every language in \( NP \) has a zero-knowledge proof.

**Proof.** The proof proceeds in two steps. First, we will give a zero-knowledge proof for 3COLOR. We will then reduce the original language \( L \) to 3COLOR using Cook’s reduction (which ensures that when we reduce an instance \( x \in L \) to an instance \( x' \in L_{3COLOR} \) we can also reduce the witness \( w \) to a witness \( w' \in R_{3COLOR}(x) \) and run the zero knowledge proof for 3COLOR on inputs \( x', w' \).

Recall that 3COLOR is the language consisting of 3-colorable graphs using a standard encoding. \( X = (V, E), c_i \in \{0, 1, 2\}, n = |V|, w = c_0, c_1, \ldots, c_n. \)

A zero knowledge proof of 3COLOR can be defined as follows.

Assume let \( C \) be the commitment scheme constructed from a one-way permutation as defined above. Let \( G(V, E) \) be a graph such that \( V = \{1, \ldots, n\} \) and let \( \pi \) describe a coloring of \( G \).

1. The prover \( P \) uniformly selects a random permutation \( \pi \) over \( \{1, 2, 3\} \). For each \( i = 1, \ldots, n \), \( P \) sends the commitment \( C(\pi(\phi(i))) \) to the verifier \( V \).
2. The verifier \( V \) uniformly selects a random edge \( e \in E \) and sends it to \( P \).
3. Upon receiving \( e = (i, j) \in E \), \( P \) decommits to the \( i^{th} \) and \( j^{th} \) values sent in Step 1.

4. \( V \) verifies that the decommitted values \( \phi(i), \phi(j) \) are different elements of \( \{1, 2, 3\} \) and that they match the commitments received in Step 1.

Recall that there are three independent properties that need to be considered: completeness, soundness, and zero knowledge.

**Completeness:** If \( G \in \text{3COLOR} \) and \( \phi \) is a valid coloring, then it is clear that \( P \) will always be able to reveal satisfactory values for \( \phi(i) \) and \( \phi(j) \), therefore \( V \) will accept the proof.

**Soundness:** If \( G \not\in \text{3COLOR} \), then \( \pi \) is not a valid 3-coloring. Therefore there must be at least one edge \( e = (i, j) \in E \) such that \( \phi(i) = \phi(j) \). Since \( V \) chooses the edge in Step 2 uniformly at random, the chance that he will choose an invalid edge is at least \( \frac{1}{|E|} \), and if he chooses that edge it will be impossible for \( P \)'s decommitted values to pass \( V \)'s verification.

**Zero Knowledge:** Given a (possibly cheating) verifier \( V^* \) (which is required to be a probabilistic polynomial-time machine), we can construct a simulator \( M_{V^*} \) as follows. Given input \((x, y)\) where \( x \) is some encoding of a graph \( G \), \( M_{V^*} \) randomly assigns colors to the vertices of \( G \) and writes down the commitments to these colors. \( M_{V^*} \) then simulates \( V^* \) to choose an edge \( e \in E \) and writes down the result. If the two vertices corresponding to the chosen edge have different colors, then \( M_{V^*} \) decommits to the colors and writes down the result. If the two vertices have the same color, \( M_{V^*} \) rewinds and tries again. The probability that the two vertices have the same color is \( \frac{1}{3} \), therefore the expected number of tries before a valid transcript is obtained is 3. Since each try takes polynomial time and since a constant number of attempts is needed, \( M_{V^*} \) runs in polynomial time as desired. Since both the color assignments of \( M_{V^*} \) and the permutations of \( P \) are chosen uniformly at random, the probability distributions that result from the interactive proof system \((P(x), V^*(x, y))\) and the simulator \( M_{V^*}(x, y) \) are indistinguishable.

The given protocol has non-negligible soundness error. In order to reduce this, we can repeat this proof in sequence, however this increases the number of rounds to super-constant. While it would be nice to simply repeat the proof in parallel instead, the zero-knowledge property is not necessarily maintained under parallel composition. We will learn more about these issues, and about possible ways around them, in the next lecture.