1 Commitments

Last time we have showed that given the electronic equivalent of a “physical” envelope – or a commitment, and the existence of one way functions, every language in NP has a zero knowledge (ZK) proof. We start by formally defining what a commitment is, and how one can it be constructed. The first thing that comes to mind is why not use encryption to provide a mechanism for commitments? Private key encryption schemes (like the OTP) cannot be used since different keys give different encryptions of the same message, and in the reveal phase the sender may trick the receiver by providing a different key. Public key encryption can be used if the key is handed over in the first round along with the commitment, however we try to come up with a commitment scheme that requires weaker assumptions than the existence of public key encryption schemes.

**Definition 1 (1-bit Commitment)** Comm is a commitment scheme if it is poly-time and the following two hold:

1. **Binding (perfect):** For all \( r_0, r_1 \in \{0,1\}^n \), \( \text{Comm}(0, r_0) \neq \text{Comm}(1, r_1) \), where \( r_0 \) and \( r_1 \) may be equal. This denotes perfect binding commitments.

2. **Hiding (computational):**

   \[
   \{ r \leftarrow \{0,1\}^n : \text{Comm}(0, r) \}_n \approx \{ r \leftarrow \{0,1\}^n : \text{Comm}(1, r) \}_n
   \]

   This means that commitment to 0 is computationally indistinguishable from the commitment to 1.

One wants to construct such a commitment. Consider the first attempt: given the bit \( b \) and the key \( r \), and let \( f \) be a one way permutation, \( \text{Comm}(b, r) = f(b, r) \). This scheme is bad since it may be the case that \( f \) leaks the first half of its input and uses only the second, therefore being trivial to distinguish between the two possible values of the bit \( b \).

**Theorem 1** Assuming the existence of OWP (one way permutations), there exists a 1-bit commitment scheme.

**Proof.** Let the scheme be: \( \text{Comm}(b, r) = f(r), b \oplus h(r) \) where \( f \) is the OWP and \( h \) is a hard-core predicate for \( f \).

The scheme is trivially perfectly binding since the following facts hold:
• Since $f$ is a permutation, $\forall r_0, r_1, r_0 \neq r_1$ it is the case that $Comm(b, r_1) \neq Comm(b, r_2)$ irrespective of the value of $b$.

• Since $h(r)$ is a deterministic predicate, $h(r) \oplus 0 \neq h(r) \oplus 1$.

Consider the hiding property, we define the following distributions (hybrids):

- $H_0 = \{ r \leftarrow \{0,1\}^n : f(r), (0 \oplus h(r)) \}$
- $H_1 = \{ r \leftarrow \{0,1\}^n; s \leftarrow \{0,1\} : f(r), (0 \oplus s) \}$
- $H_2 = \{ r \leftarrow \{0,1\}^n; s \leftarrow \{0,1\} : f(r), (1 \oplus s) \}$
- $H_3 = \{ r \leftarrow \{0,1\}^n : f(r), (1 \oplus h(r)) \}$

$H_0 \approx H_1$ by the same argument in the proof we previously had for the $PRG$ with 1-bit expansion, likewise $H_2 \approx H_3$. And since $H_1 = H_2$, by the polyjump lemma $H_0 \approx H_3$ and the scheme satisfy the computational hiding property as well; therefore the scheme is a 1-bit commitment scheme.

\[ \square \]

## 2 ZK proofs of knowledge (ZKPOK)

It is known that sequential composition does work, however parallel composition or repetition may or may not be ZK. There exist problems for which runs in parallel leaks the witness.

We consider a new type of proof, one in which if the prover convinces the verifier, then it must be the case that the prover knows the witness. For example consider the ZK proof that $y = f(x)$ where $f$ is a OWP, $f : \{0,1\}^n \rightarrow \{0,1\}^n$ and the witness relation $R_L = \{ y, x : f(x) = y \}$. Consider the ZK proof that $y \in L$ where $L$ is defined by $R_L$; then since for every $y$, $\exists x$, and if $P$ is silent, and $V$ accepts we have a vacuous valid ZK proof.

Proof of knowledge: one wants to prove that given a particular $y$ one can find the corresponding $x$.

**Definition 2 (Proof of knowledge)** Let $(P, V)$ be an interactive proof for language $L$ with witness relation $R_L$. $(P, V)$ is said to be a proof of knowledge for $R_L$ if $\exists$ a p.p.t. extractor $E$ such that $\forall$ turing machines $P'$, $\exists$ a negligible function $\epsilon$ such that $\forall x, z \in \{0,1\}^*$
\[ \Pr[E_{P'}(x,z) \in R_L(x)] \geq \Pr[\text{Out}_V[P'(x,z) \leftrightarrow V(x)] = \text{accept}] - \epsilon(|x|) \]

**Theorem 2** Assuming the existence of one way functions, there exists a ZK proof of knowledge (ZKPOK) for every language in NP.

**Proof idea:**
Show that G3C (graph 3-coloring) is a ZKPOK, actually the proof will be for graph-iso. Basically what we do is show that \( n \) sequential repetitions of graph-iso is a proof of knowledge, and we do so by finding an extractor \( E \).

Recall the ZK proof for graph-iso:
- The input is the graphs \( G_1 \) and \( G_2 \), the prover \( P \) has the permutation \( \sigma : \sigma(G_1) = G_2 \).
- \( P \) picks a random permutation \( \pi \) and sends \( \pi(G_1) \) to the verifier \( V \).
- \( V \) randomly chooses the bit \( b \leftarrow \{0,1\} \) and sends it back to \( P \).
- If \( b = 0 \), \( P \) sends \( \pi \) to \( V \), otherwise it sends \( (\pi^{-1} \circ \sigma) \) to \( V \).
- \( V \) checks if \( \pi(G_1) = H \) or if \( (\pi^{-1} \circ \sigma)(H) = G_2 \).

Let \( E \) be the extractor with oracle access to \( P' \), where \( P' \) is a prover TM for graph-iso. Since \( E \) has oracle access to \( P' \) it can rewind and restart it such that \( P' \) will be again in the state when it provides the same \( H \) – which is the same permutation of \( G_1 \), and therefore \( E \) can ask for both \( \pi \) and \( (\pi^{-1} \circ \sigma) \) and if the answers are correct it can extract \( \sigma \). \( E \) works as follows:
- If \( P' \) answers for both \( b = 0 \) and \( b = 1 \) correctly, then find \( \sigma \) and stop.
- If \( P' \) answers none correctly then abort.
- If \( P' \) answers only one correctly then continue to the next iteration. \( P' \) will return a different permutation of \( G_1 \), call it \( H' \), and since \( E \) can remember the prefix that leads \( P' \) to this state it can again force a response for both \( b = 1 \) and \( b = 0 \) given the same \( H' \).
- \( E \) can perform \( n \) sequential runs of the graph-iso protocol.

Note that if \( E \) aborts then \( P'(x,z) \) convinces \( V \) with probability 0. Otherwise the extractor \( E \) only fails if in none of the interactions \( P' \) answers for both \( b = 1 \) and \( b = 0 \) correctly. This means that \( P'(x,z) \) convinces \( V \) only with probability \( 2^{-n} \) (since the protocol is allowed only \( n \) runs of graph-iso). Therefore if \( P'(x,z) \) convinces \( V \) with probability \( > 2^{-n} \) it holds that \( \Pr[E_{P'}(x,z) \in R_L(x)] = 1 \). \( \blacksquare \)