This problem set is due on: May 3, 2005.

Problem 1 - Perfectly Hiding Commitment

Definition:
A two-round perfectly-hiding commitment scheme is a triple of efficient algorithms $(GEN, COM, VER)$ satisfying the following properties.

Correctness: For all security parameters $k$ and inputs $\alpha$,
\[
\Pr[g \leftarrow GEN(1^k); (c, d) \leftarrow COM(g, \alpha) : VER(g, c, d, \alpha) = TRUE] = 1
\]

Binding: For all $k$, and for any probabilistic polynomial-time cheating committer $C^*$:
\[
\Pr[g \leftarrow GEN(1^k); (c, d_1, d_2, \alpha_1, \alpha_2) \leftarrow C^*(g) : \newline
VER(g, c, d_1, \alpha_1) = VER(g, c, d_2, \alpha_2) = TRUE \land \alpha_1 \neq \alpha_2] < \text{negligible}(k)
\]

Perfect Hiding: For all $k$, and all inputs $\alpha$ and $\beta$ the following distributions are identical:
\[
\langle g \leftarrow GEN(1^k); (c, d) \leftarrow COM(g, \alpha) : (g, c) \rangle = \langle g \leftarrow GEN(1^k) ; (c, d) \leftarrow COM(g, \beta) : (g, c) \rangle
\]

Protocol:
Consider the following two-round protocol for committing to a $k$-bit value, $\alpha$. The algorithm $GEN$ randomly selects $(p, g, h)$ subject only to the following conditions: (1) $p$ is a $k + 1$-bit prime number and (2) $g$ and $h$ are generators of $Z_p^*$. The algorithm $COM$ on input $(p, g, h)$ and $\alpha$ selects a random $t \in Z_p^*$ and outputs the commitment message $c = g^t h^\alpha \mod p$ and the decommitment message $t$. The algorithm $VER$ on input $(p, g, h), c, t$ and $\alpha$ outputs $TRUE$ if and only if $c = g^t h^\alpha \mod p$.

Prove: the above protocol is, in fact, a perfectly-hiding commitment scheme.
Problem 2 - Zero-Knowledge in Parallel

Let \((\text{GEN}, \text{COM}, \text{VER})\) be a perfectly hiding commitment scheme. Here we provide a five-round proof system for ISO.\(^1\) with negligible soundness error.

1. The prover selects \(g \leftarrow \text{GEN}(1^k)\) and sends \(g\) to the verifier.
2. The verifier chooses a \(k\)-bit random string \(r\), selects \((c, d) \leftarrow \text{COM}(g, r)\) and sends \(c\) to the prover.
3. The prover randomly selects \(k\) graphs \(C_1, \ldots, C_k\) such that each \(C_i\) is isomorphic to \(G\) and sends \(C_1, \ldots, C_k\) to the verifier.
4. The verifier sends \(d\) and \(r\) to the prover.
5. If \(r = \text{VER}(g, c, d)\) then for each graph \(C_i\) the prover sends the verifier a random isomorphism mapping \(G\) to \(C_i\) if the \(i\)th bit of \(r\) is 0 and a random isomorphism mapping \(H\) to \(C_i\) if the \(i\)th bit of \(r\) is 1.

**Prove:** the above protocol is, in fact, a zero-knowledge proof system for ISO.

Problem 3 - Hiding and Binding

**Prove or Disprove:** There exists a bit commitment scheme which is both perfectly hiding and perfectly binding.

*Note:* A perfectly hiding commitment scheme is defined in problem 1. A commitment scheme is perfectly binding if the binding condition holds with respect to all cheating commiters (as opposed to only those running in probabilistic polynomial-time). Encryption is an example of a perfectly binding commitment scheme.

Problem 4 - Proofs of Knowledge

Let \(L\) be a language in \(\text{NP}\) and for \(x \in L\) let \(W_x\) be the set of \(\text{NP}\)-witnesses for \(x\). Informally, \((P, V)\) is a ZK proof of knowledge for \(L\) if on common input \(x\), \(P\) convinces \(V\) that he knows an element of \(W_x\) and yet interacting with \(P\) provides \(V\) provides \(P\) with no knowledge other than that \(x \in L\). (In particular, \(V\) learns nothing about which element of \(W_x\) the prover knows!)

Provide a formal definition of a zero-knowledge proof of knowledge and explain why your definition captures informal notion above.

\(^1\)The language of all pairs of graphs \((G, H)\) such that \(G\) is isomorphic to \(H\).