CS 120/ E-177: Introduction to Cryptography

Salil Vadhan and Alon Rosen Dec. 14, 2006

Lecture Notes 21:

Zero-Knowledge Proofs II

Recommended Reading.

- Vadhan, *Interactive & Zero-Knowledge Proofs*, from IAS/PCMI Summer School on Computational Complexity, Secs 1.1, 1.2, 2.1, 2.2.
- Goldreich, Chapter 4 (up to 4.4)

1 Zero Knowledge for NP

An NP-complete problem: GRAPH 3-COLORING.

- An (undirected) graph $G = (W, E)$ is *3-colorable* if there is a function $C : W \to \{R, Y, B\}$ such that for all $(u, v) \in E$, $C(u) \neq C(v)$.
- $3\text{COL} = \{G : G \text{ is 3-colorable}\}$.
- For every $L \in \mathbf{NP}$, there is a poly-time f such that $x \in L \Leftrightarrow f(x) \in 3\text{COL}$.
- Moreover, given any **NP** proof system for L, we can choose f such that valid **NP** proofs for $x \in L$ can be mapped in poly-time to valid 3-colorings of $f(x)$.

Cut and Choose:

- $G \in 3\text{COL} \Leftrightarrow \exists C \ \left(\bigwedge_{(u,v)\in E} C(u) \neq C(v)\right)$.
- If we randomly permute the 3 colors, each pair $(C(u), C(v))$ for $u \neq v$ reveals no information.
- Have prover 'commit' to randomized coloring C, verifier pick a random edge.

Physical Zero-Knowledge Proof: See video.

Definition 1 *A* commitment scheme *over message space* $\mathcal{P} = \bigcup_n \mathcal{P}_n$ *is a polynomial-time computable function* $\text{Com}(m, k)$ *satisfying:*

- *(Hiding) For every* $m, m' \in \mathcal{P}_n$ *such that* $\|m\| = \|m'\|$, $\text{Com}(m, K) \stackrel{c}{\equiv} \text{Com}(m', K)$, *when* $K \stackrel{R}{\leftarrow} \{0,1\}^n$.
- *(Binding) There do not exist* $m \neq m'$ *and* k, k' *such that* $\text{Com}(m, k) = \text{Com}(m, k')$.

Zero-Knowledge Proof for GRAPH 3-COLORING

Common input: A graph $G = (W, E)$ on n vertices.
Prover's input: A valid 3-coloring $C : W \to \{R, Y, B\}$ (in case $G \in \text{3COL}$)

1. P: Choose a permutation $\pi : \{R, Y, B\} \to \{R, Y, B\}$ uniformly at random, and set $C' = \pi \circ C$. For every vertex $w \in W$, choose $k_w \stackrel{R}{\leftarrow} \{0,1\}^n$ and send $z_w = \text{Com}(C'(w), k_w)$ to V.
2. V: Choose an edge $(u, v) \stackrel{R}{\leftarrow} E$, and send (u, v) to P.
3. P: Check that $(u, v) \in E$, and if so send $C'(u)$, $C'(v)$, k_u, k_v to V.
4. V: Accept if $C'(u) \neq C'(v)$, $z_u = \text{Com}(C'(u), k_u)$ and $z_v = \text{Com}(C'(v), k_v)$.

Theorem 2 *Above is a zero-knowledge proof for* GRAPH 3-COLORING.

Proof:

- Perfect completeness.
- Soundness error $1 - 1/|E|$. Reduce by repetition.

Simulator S^{V^*}, on input $G = (W, E)$:

1. Select $(u, v) \stackrel{R}{\leftarrow} E$.
2. Define a coloring C' by setting $(C'(u), C'(v))$ to be two random distinct colors in $\{R, Y, B\}$, and setting $C'(w) = R$ for all other vertices w.
3. For every $w \in W$, choose $k_w \stackrel{R}{\leftarrow} \{0,1\}^n$, and set $z_w = \text{Com}(C'(w), k_w)$.
4. Select random coin tosses r for V^*, and let $(u^*, v^*) = V^*(G, \{z_w\}_{w \in W}; r)$.
5. If $(u^*, v^*) \neq (u, v)$, output `fail`. Otherwise, output $(\{z_w\}_{w \in W}, (u, v), (k_u, k_v, C'(u), C'(v)); r)$.

Claim 3 *For every PPT* V^* *and* $G \in \text{3COL}$, *we have*

1. $S^{V^*}(G)$ *succeeds with probability at least* $1/|E| - \text{neg}(n)$*, and*

2. *The output distribution of $S^{V^*}(G)$, conditioned on success, is computationally indistinguishable from* $\mathsf{View}_{V^*}^{(P,V)}((P,V)(G))$.

Repeat $n \cdot |E|$ times to eliminate failure. ■

Corollary 4 *Every language in* **NP** *has a zero-knowledge proof.*

2 Compiling Protocols to Handle Malicious Adversaries

First Attempt. Let (A, B) be a protocol for computing $f(a, b)$ that is secure vs. honest-but-curious adversaries. Consider the following new protocol (A', B') when the two parties' inputs are a and b respectively.

1. A': Choose random coin tosses r_A for A and $k_A \stackrel{\text{R}}{\leftarrow} \{0,1\}^n$, and send $z_A = \text{Com}((a, r_A), k_A)$.
2. B': Choose random coin tosses r_B for B and $k_B \stackrel{\text{R}}{\leftarrow} \{0,1\}^n$, and send $z_B = \text{Com}((b, r_B), k_B)$.
3. A': Compute and send the first message m_1 of A, as $m_1 = A(a; r_A)$.
   Use a zero-knowledge proof to convince B' that m_1 is consistent with z_A. (Why is this an **NP** statement?)
4. B': If the zero-knowledge proof fails, abort. Otherwise, compute the first message m_2 of B, as $m_2 = B(b, m_2; r_B)$.
   Use a zero-knowledge proof to convince A' that m_2 is consistent with z_A and m_1.
5. etc.

Q: How can one still cheat in this protocol?