Lecture Notes 20

49 Further Remarks on Bit-Commitment

49.1 Prediction Versus Distinguishability

In the last lecture, although the formal secrecy definition for bit-commitment is in terms indistinguishability, I was talking informally as if it were the same as unpredictability. This isn't quite right, since the notion of indistinguishability does not depend on any assumed probability distribution on v, whereas a definition of secrecy based on unpredictability requires an assumed underlying distribution.

Recall that indistinguishability is introduced in section 26 of lecture notes 10. Unpredictability is introduced in the context pseudorandom generators in section 30 of lecture notes 12. I review these concepts in terms of bit-commitment.

Given a commitment scheme $C_s(v)$, a *predictor* is a p.p.t. algorithm A that tries to guess v given $C_s(v)$. It *succeeds* on a given run iff it outputs v. Since A itself is probabilistic, A has a success probability $p(s, v)$ given by

$$p(s, v) \stackrel{\text{df}}{=} \Pr[A(C_s(v)) = v].$$

Its overall success probability on v with security parameter n is $p'_n(v) = p(U_n, v)$, that is, we average $p(s, v)$ over all s of length n. Now, it is tempting to say that secrecy holds if $p'_n(V) = 1/2 + \epsilon(n)$ for some negligible function $\epsilon(n)$, where V is a random variable ranging over $\{0, 1\}$. This definition works if V is uniformly distributed. However, when V is not uniformly distributed, we may be able to correctly guess V with success probability significantly greater than $1/2$ based solely on knowledge of the distribution, and this ability to predict does not indicate a weakness in the commitment scheme. For example, if $\Pr[V = 1] = 2/3$, then the strategy of always guessing $V = 1$ is correct $2/3$ of the time, that is, $p'_n(V) = 2/3$.

What we want instead is a definition based on the information about v that is provided by the commitment $c = C_s(v)$. Namely, we'll say that the commitment scheme satisfies secrecy if knowledge of c increases an adversary A's success probability at correctly guessing v by only a negligible amount over what it could do without knowing c.

Formally, a bit-commitment scheme $C_s(v)$ is *unpredictable* iff for all p.p.t. algorithms A and all distributions V over $\{0, 1\}$,

$$\Pr[A(C_{U_n}(V)) = V] \leq \max\{\Pr[V = 0], \Pr[V = 1]\} + \epsilon(n)$$

where $\epsilon(n)$ is a negligible function.

Coming back to indistinguishability, the general definition of secrecy given in section 48 of lecture notes 19, when applied to a simple one-way communication scheme $C_s(v)$, just says that the probability ensembles

$$\{C_{U_n}(0)\}_{n\in\mathbb{N}} \quad \text{and} \quad \{C_{U_n}(1)\}_{n\in\mathbb{N}}$$

are computationally indistinguishable.

We invite the reader to think about how to show that an unpredictable bit-commitment scheme satisfies indistinguishability and vice versa.

49.2 Bit-commitment based on quadratic residuosity

The Goldwasser-Micali quadratic residue cryptosystem QR can be used for bit commitment. The idea is that any quadratic residue modulo n is a commitment to 0, and any quadratic non-residue with Jacobi symbol 1 is a commitment to 1.

Commit phase Using the random tape s, the sender $S(v)$ computes the commitment $C_s(v)$ as follows: The sender first generates a QR key consisting of $n = pq$ for two primes p and q and a number y such that $\left(\frac{y}{p}\right) = \left(\frac{y}{q}\right) = -1$. This implies that y has Jacobi symbol $\left(\frac{y}{n}\right) = 1$ and that y is not a quadratic residue modulo n. (Note that these two facts hold even if p and/or q are composite.) The sender then generates a random string x of length n, computes $c = (x^2 \bmod n)$ if $v = 0$ and $c = (x^2 y \bmod n)$ if $v = 1$, and sends c to the receiver. Thus, c is a quadratic residue iff $v = 0$.

Reveal phase Sender sends (v, s). Receiver checks that $c = C_s(v)$.

Secrecy is based on the (presumed) indistinguishability of random quadratic residues from random quadratic non-residues with Jacobi symbol 1. Unambiguity is perfect since a given number c either is or is not a quadratic residue modulo n.

Note that there is a subtle issue here when a probabilistic primality testing algorithm is used in finding the primes p and q, for there is a non-zero probability that it will fail and result in numbers p and q that are not prime. Nevertheless, y cannot be a quadratic residue modulo n since if it were, then we would have $\left(\frac{y}{s}\right) = 1$ for every prime divisor s of n, but $\left(\frac{y}{p}\right) = -1$ implies that $\left(\frac{y}{s}\right) = -1$ for some prime divisor s of p.

49.3 Extensions to bit-commitment

The commitment protocol needed for G3C is a little more general in two respects than what we have discussed so far.

1. The G3C protocol needs commitment to one of three colors – a ternary value. The extension is straightforward. Namely, express the ternary value as a two-bit binary string and separately commit each of the two bits.

2. The proof that the G3C protocol is zero knowledge requires a commitment scheme with non-uniform secrecy. Namely, secrecy is required with respect to all families of polynomial size circuits. Non-uniform commitment schemes can be constructed (using the same constructions) assuming the existence of non-uniform one-way functions.

50 Proof that G3C Construction is Zero Knowledge

As usual, to prove zero knowledge, we construct a simulator M^* for V^*'s view of its interaction with P_{G3C}. Assume that $q(n)$ is a polynomial that bounds V^*'s running time. Here's what M^* does on input $G = (V, E)$:

- Uniformly and independently select $e_1, \ldots, e_n \in \{1, 2, 3\}$.
- Choose random strings $s_1, \ldots, s_n \in \{0, 1\}^n$.
- Set $d_i = C_{s_i}(e_i)$.

- Run V^* with G as common input, random $r \in \{0,1\}^{q(n)}$ on its random coins tape, and $(d_1, \ldots, d_n)$ on its incoming message tape.
- Let (u, v) be V^*'s outgoing message. [Force it to some edge if (u, v) is not produced or not an edge of G.]
- If $e_u \neq e_v$, then halt with output $(G, r, (d_1, \ldots, d_n), (s_u, e_u, s_v, e_v))$. Otherwise, halt with output $\perp$.

Claim 1 $\Pr[M^*(G) = \perp] \leq \frac{1}{3} + \frac{1}{p(|V|)} \leq \frac{1}{2}$.

Proof: $M^*(G)$ outputs $\perp$ iff V^* chooses an edge (u, v), both of whose endpoints are colored the same. If the choice of edge were completely independent of the coloring, then the probability of the two endpoints being colored the same (assuming each vertex is colored uniformly and independently from $\{1, 2, 3\}$) would be exactly $1/3$. In fact, the choice of edge is not independent of the coloring since V^* has access to the commitments of the coloring. We are thus led to analyze the probabilities in greater detail.

Let $p_{u,v}(G, r, (e_1, \ldots, e_n))$ be the probability (over all choices for $s_1, \ldots, s_n$) that V^* on $(G, r, (C_{s_1}(e_1), \ldots, C_{s_n}(e_n)))$ replies with (u, v).

Subclaim *Assume that C_s satisfies non-uniform secrecy. Then for every positive polynomial $p(\cdot)$, every sufficiently large graph $G = (V, E)$, every $r \in \{0,1\}^{q(n)}$, every edge $(u, v) \in E$, and every two sequences $\alpha, \beta \in \{1, 2, 3\}^n$, it holds that*

$$|p_{u,v}(G, r, \alpha) - p_{u,v}(G, r, \beta)| \leq \frac{1}{p(n)}.$$

Proof: The proof of the subclaim is by a reducibility argument. Assuming the subclaim is false, we construct a distinguisher for the commitment scheme that violates non-uniform secrecy. (See textbook for details.) ■

To finish the proof of the claim, one must calculate probabilities that (u, v) is chosen given $\bar{e} \in \{1, 2, 3\}^n$ by analyzing the number of illegally colored edges in each $\bar{e}$. Again, we refer the reader to the textbook for details. ■

Claim 2 *For every probabilistic polynomial-time algorithm A, every positive polynomial $p(\cdot)$, and all sufficiently large graphs $G = (V, E)$,*

$$|\Pr[A(M^*(G)) = 1 \mid M^*(G) \neq \perp] - \Pr[A(\text{view}_{V^*}^{P_{\text{G3C}}}(G)) = 1]| < \frac{1}{p(|V|)}.$$

We omit the rather complicated proof and refer the interested reader to the textbook.

51 Polynomial Time Reducibility

Definition: Let L_1, L_2 be languages. L_1 *is polynomial-time reducible to* L_2, written $L_1 \leq_{\text{p}} L_2$, if there exists a polynomial-time computable function f such that for all x,

$$x \in L_1 \iff f(x) \in L_2.$$

A language L' is *$\mathcal{NP}$-complete* if $L \leq_{\mathrm{p}} L'$ for every $L \in \mathcal{NP}$. It follows that if L' could be decided in polynomial time, then every language $L \in \mathcal{NP}$ could be decided in polynomial time. This would show that $\mathcal{P} = \mathcal{NP}$, settling one of the most important questions of computer science.

The language G3C is known to be $\mathcal{NP}$-complete, so for every $L \in \mathcal{NP}$, there exists a polynomial-time function f_L such that $L \leq_{\mathrm{p}}$ G3C via f_L. In addition, there is a function g_L that maps witnesses for L to witnesses for G3C. Let R_L be the defining polynomial-time relation for L. (See section 3.5 of lecture notes 1.) Thus, if $(x, w) \in R_L$, then $g_L(x, w)$ is a 3-coloring of the graph $f_L(x)$.

52 A Zero Knowledge Interactive Proof for Every $L \in \mathcal{NP}$

We now come to the main theorem of this unit.

Theorem 1 (4.4.11 of textbook) *Suppose that there exists a commitment scheme satisfying the (non-uniform) secrecy and unambiguity requirements. Then every language in $\mathcal{NP}$ has an auxiliary-input zero-knowledge proof system. Furthermore, the prescribed prover in this system can be implemented in probabilistic polynomial time provided it gets the corresponding $\mathcal{NP}$-witness as auxiliary input.*

Proof: We describe the auxiliary-input interactive proof system for a language L. The common input is a string $x \in L$. The prover receives a witness w on its (private) auxiliary input. Both parties compute $G = f_L(x)$. The prover computes $\psi = g_L(x, w)$, a 3-coloring of G. The prover and verifier then jointly run as a subprotocol the auxiliary-input zero-knowledge interactive proof system for G3C on common input G and prover's auxiliary input ψ to prove that $G \in$ G3C. V accepts $x \in L$ iff it accepts $G \in$ G3C in the subprotocol. ■