

Homework 9

Instructor: Anna Lysyanskaya

Due: April 18, 2007

Problem 1: Collision Resistant Hash Functions

Let $\{G, h_K\}$ be a family of collision-resistant hash functions, where G is the key generation algorithm that generates a key K , and $h_K : D_K \mapsto \{0, 1\}^*$ is the function indexed by key K , with domain D_K . Recall that a function is collision-resistant if the following are satisfied:

Non-triviality : for all keys $K \in G$, for any input x , $|x| > k$, $h_K(x)$ is always shorter than x .

Collision-resistance : for all PPT adversaries $\{A_k\}$, there exists a negligible function $\nu(k)$ such that

$$\Pr[K \leftarrow G(1^k); (x_1, x_2) \leftarrow A_k(K) : h_K(x_1) = h_K(x_2) \wedge x_1, x_2 \in D_K] = \nu(k)$$

(recall that D_K is the domain of the function h_K .)

- Let $\{G, h_K\}$ be a family of collision-resistant hash functions, where D_K , the domain of the function, consists of all strings of length up to $L(k)$, and $L(k) > k$ is a polynomial. Let $h'_K(x) = h_K(h_K(x))$. Is $\{G, h'_K\}$ a family of collision-resistant hash functions?
- Suppose you have a collision-resistant hash function that reduces its input by a little bit, for example from $k+1$ bits to k bits. Your task is to design a hash function that reduces an $L > k$ bit string into a k bit string, in such a way that it is hard to find collisions. More precisely, let $\{G, h_K\}$ be a family of collision-resistant hash functions, $h_K : \{0, 1\}^{k+1} \mapsto \{0, 1\}^k$. Let $L(k) > k$ be any given polynomial. Design a collision-resistant hash function family $\{G, h_K^L\}$ such that $h_K^L : \{0, 1\}^{L(k)} \mapsto \{0, 1\}^k$.

Problem 2: Zero Knowledge Proofs

Recall that in homework 2, we examined a zero knowledge proof protocol for the language of vertex cover.

- On homework 2, we discussed a physical protocol using physical drawings and paper cups. Explain how we could implement the same protocol cryptographically using commitments.
- Recall our definition of zero knowledge. We said that a proof scheme with prover algorithm P , and verifier algorithm V is zero knowledge iff:

\forall ppt V' , \exists ppt S such that $\forall x \in L$, \forall witnesses w the following two distributions are indistinguishable:

$$D_0 = \{P(1^k, x, w) \leftrightarrow V'(1^k, x) \rightarrow \text{view} : \text{view}\}$$

$$D_1 = \{S(1^k, x) \leftrightarrow^* V'(1^k, x) \rightarrow \text{view} : \text{view}\}$$

Note that in the second experiment, S is allowed to rewind V' and run him on different inputs polynomially many times.

Describe a simulator for the vertex cover proof protocol you gave in part (a).

- Argue that the protocol given in part (a) is zero knowledge.

Problem 3: Commitments

- a. Recall the definition of commitments that we discussed in class:

Let $Commit$ be algorithm which takes input of the form $1^k, x, r$, where k is a security parameter, $x \in Dom$ is the value we will commit to, and $r \in \{0, 1\}^{l(k)}$ is the randomness used. $Commit$ is a perfectly binding commitment scheme if the following two properties hold:

Hiding \exists a randomized algorithm $FakeCom$ such that $\forall x \in Dom$, the distribution

$$D_0 = \{r \leftarrow \{0, 1\}^{l(k)}; C \leftarrow Commit(1^k, x, r) : C\}$$

is indistinguishable from the distribution

$$D_1 = \{C \leftarrow FakeCom(1^k) : C\}.$$

Perfect Binding $\forall \mathcal{A}$,

$$\Pr[C, x_1, x_2, r_1, r_2 \leftarrow \mathcal{A}(1^k) : C = Commit(1^k, x_1, r_1) = Commit(1^k, x_2, r_2) \wedge x_1 \neq x_2] = 0$$

Now consider the function $f(y)$, which first parses y as $x \circ r$, and then computes $Commit(1^k, x, r)$.

Prove that if $Commit$ is a commitment scheme as defined above, then f is a one way function. (Give a reduction).

- b. We can consider a slightly weaker definition which we call Computationally Binding Commitments. Intuitively, here we want a slightly weaker binding property, so that there is a negligible probability that the adversary will be able to produce two openings $(x_1, r_1$ and $x_2, r_2)$ for the same commitment. However, note that when we make this change, we also need to change our definition to consider families of commitment schemes keyed by some public key. Explain why.
- c. It turns out that adding a public key also allows us to get commitments which satisfy a stronger hiding property.

The resulting definition goes as follows: An pair of algorithms $Gen, Commit$ is a computationally binding perfectly hiding commitment scheme if the following two properties hold:

Hiding \exists a randomized algorithm $FakeCom$ such that $\forall x \in Dom, \forall PK$ in the output range of Gen , the distribution

$$D_0 = \{r \leftarrow \{0, 1\}^{l(k)}; C \leftarrow Commit_{PK}(1^k, x, r) : (C, PK)\}$$

is identical to the distribution

$$D_1 = \{C \leftarrow FakeCom(1^k, PK) : (C, PK)\}.$$

Computational Binding \forall ppt. \mathcal{A}, \exists negligible function $\nu(k)$ such that

$$\Pr[PK \leftarrow Gen, C, x_1, x_2, r_1, r_2 \leftarrow \mathcal{A}(1^k, PK) : C = Commit_{PK}(1^k, x_1, r_1) = Commit_{PK}(1^k, x_2, r_2) \wedge x_1 \neq x_2] = \nu(k)$$

Now consider a family of functions G, f . G will run Gen to obtain a public key PK . $f_{PK}(y)$ will first parse y as $x \circ r$, and then compute $Commit_{PK}(1^k, x, r)$.

Let's prove that if $Gen, Commit$ is a computationally binding commitment scheme, then G, f as defined above, is a family of one way functions.

Suppose we are given an adversary \mathcal{A} who with high probability can, given $f_{PK}(y)$, find y' such that $f_{PK}(y) = f_{PK}(y')$. Consider the following two cases:

- Suppose that when we give the adversary $f_{PK}(y = x \circ r)$, with nonnegligible probability, he outputs $y' = x \circ r'$ (where it is possible that $r' = r$). Then show that we can build an algorithm \mathcal{B} which breaks the Hiding property.
- Now suppose that when we give the adversary $f_{PK}(y = x \circ r)$, with nonnegligible probability, he outputs $y' = x' \circ r'$, where $x' \neq x$ (and again it is possible that $r' = r$). Then show that we can build an algorithm \mathcal{B} , which breaks the Binding property of the commitment.
- Conclude the proof