Prof. Martin Hofmann
Dr. Ulrich Schöpp

Ludwig-Maximilians-Universität München
Institut für Informatik
Summer 2017

Cryptography

Exercise Sheet 10

Exercise 10-1 Prove that hardness of the Decisional Diffie-Hellman problem (DDH) implies hardness of the Discrete Logarithm Problem.

Solution (Sketch): Let $\mathcal{A}$ be an adversary for the Discrete Logarithm Problem. We construct an adversary $\mathcal{B}$ for the DDH problem, using $\mathcal{A}$ as a subroutine. The assumption about the hardness of the DDH problem will give us

$$|P(\mathcal{B}(G, q, g, g^x, g^y, g^z) = 1) - P(\mathcal{B}(G, q, g, g^x, g^y, g^{xy}) = 1)| \leq negl(n).$$

We construct $\mathcal{B}$ such that this inequality allows us to bound the success probability of $\mathcal{A}$.

When started, $\mathcal{B}$ is given G, q, g, g^x, g^y and h as input and then checks if $h = g^{xy}$ as follows. It gives G, q, g and g^x to $\mathcal{A}$, which returns a number x'. By computing $g^{x'}$, $\mathcal{B}$ can verify if x' is indeed the logarithm of g^x. If not, $\mathcal{B}$ returns 0. Otherwise we must have $x' = x$. Then, $\mathcal{B}$ calculates $(g^y)^x$ and returns 1 if this is equal to h. Otherwise $\mathcal{B}$ return 0.

We next analyse the succeess probability of $\mathcal{B}$. First, $P(\mathcal{B}(G, q, g, g^x, g^y, g^z) = 1) = negl(n)$ since $\mathcal{B}$ returns 1 only if $z = xy$ and z is chosen at random. Second, $P(\mathcal{B}(G, q, g, g^x, g^y, g^{xy}) = 1)$ is the probability that $\mathcal{A}$ gives the right answer x for the request g^x. Since x is random, this is just the success probability of $\mathcal{A}$ in the discrete logarithm experiment, i.e.

$$P(\mathcal{B}(G, q, g, g^x, g^y, g^z) = 1) = P(\mathsf{DLog}_{\mathcal{A},G}(n) = 1)$$

Overall, we get

$$|P(\mathcal{B}(G, q, g, g^x, g^y, g^z) = 1) - P(\mathcal{B}(G, q, g, g^x, g^y, g^{xy}) = 1)| = P(\mathsf{DLog}_{\mathcal{A},G}(n) = 1) + negl(n)$$

By assumption that the DDH problem is hard, the left-hand-side is neglible, so $P(\mathsf{DLog}_{\mathcal{A},G}(n) = 1)$ must be neglible.

Exercise 10-2 The RSA hardness assumption states that $P(\mathsf{RSA\text{-}inv}_{\mathcal{A}}(n) = 1) \leq negl(n)$ for any probilistic polynomial time adversary $\mathcal{A}$.

Show that the RSA hardness assumption implies that factoring is hard in the following sense: No probabilistic polynomial time adversary $\mathcal{B}$ can succeed in the following factoring experiment with non-neglible probability.

1. Randomly generate two primes p and q and let $N := p \cdot q$.
2. Adversary $\mathcal{B}$ is given N and returns two numbers p' and q'.
3. The adversary succeeds in the experiment if $p' \cdot q' = N$.

Solution (Sketch): We show the contraposition this time: Suppose there is an adversary $\mathcal{B}$ that solves factoring with non-neglible probability. We construct from $\mathcal{B}$ an adversary $\mathcal{A}$ that solves RSA-inv with non-neglible probability.

Initially, $\mathcal{A}$ is given N, e and $x^e \mod N$. Then, $\mathcal{A}$ uses $\mathcal{B}$ as a subroutine and gives N to $\mathcal{B}$. As an adversary in the factoring experiment, $\mathcal{B}$ produces two numbers p' and q'. Next, $\mathcal{A}$ checks that these are a factorization of N. If they are not, then $\mathcal{A}$ returns a random value. Otherwise, $\mathcal{A}$ can compute $\phi(N) = (p'-1)(q'-1)$ and with thiw can use the extended Euclidean Algorithm to compute d with $e \cdot d = 1 \mod \phi(N)$. Using $\mathcal{A}$ can compute $(x^e)^d \mod N$, which is x, which it then returns.

Suppose $\mathcal{B}$ succeeds with probability s. Then $\mathcal{A}$ succeeds whenever $\mathcal{B}$ as its subroutine succeeds, i.e. also with probability s (since N, e and x are chosen independently). So if $\mathcal{B}$ succeeds with nonnegligible probability, then so does $\mathcal{A}$. The contraposition is: If $\mathcal{A}$ succeeds with neglible probability, then so does $\mathcal{B}$, which is what we had to show.

Exercise 10-3 For any given pseudorandom function F, one can attempt to define a hash function (Gen, H) by letting $\mathsf{Gen}(n)$ be a random string s of length n and defining $H_s(x) := F_s(x)$. Would this definition always produce a collision-resistant hash function?

Solution (Sketch): This definition would not always produce a collision-resistant hash function. We construct a counter example F'.

Let F be an arbitrary PRF and modify it as follows:

$$F'(k,x) = \begin{cases} \vec{0} & \text{if } |k-x| \leq 1 \text{ (difference as binary numbers)} \\ F(k,x) & \text{otherwise} \end{cases}$$

First we show that F' is still a PRF. By assumption we know

$$|P(D^f(n)=1) - P(D^{F_k}(n)=1)| \leq negl(n)$$

for any distinguisher D.

We analyse $P(D^{F'_k}(n)=1)$. Consider first the computation of $D^{F_k}(n)$. The distinguisher D is a probabilistic polynomial time algorithm, so it can make at most polynomially many different requests to the oracle function, say with arguments $x_1, \ldots, x_m$. If none of these values is in $\{k-1, k, k+1\}$, then we could replace F_k with F'_k without affecting the computation. Write X for the event that one of the $x_1, \ldots, x_m$ is in $\{k-1, k, k+1\}$. Since k is random, the probability of x_i being in this set is $3/2^n$. Since the number m is polynomial in n, the overall probability of X is neglible in n, so $P(X) = negl(n)$.

We have $P(D^{F'_k}(n) = b) \leq P(D^{F'_k}(n) = b \mid \overline{X}) + P(X)$ (recall Exercise 2-1-a). But $P(D^{F'_k}(n) = b \mid \overline{X}) = P(D^{F_k}(n) = b)$, so we have $P(D^{F'_k}(n) = b) \leq P(D^{F_k}(n) = b) + negl(n)$.

This gives us $|P(D^{F'_k}(n) = 1) - P(D^{F_k}(n) = 1)| \leq negl(n)$[1].

Together with the assumption, this gives us $|P(D^f(n) = 1) - P(D^{F'_k}(n) = 1)| \leq negl(n)$, i.e. F' is also a PRF.

But F' is not collision-resistant. To construct a collision, an adversary is given k. The adversary can reliably construct a collision by choosing k and either $k - 1$ or $k + 1$ as its values.

[1]For this it suffices to show $P(D^{F_k}(n) = 1) - negl(n) \leq P(D^{F'_k}(n) = 1) \leq P(D^{F_k}(n) = 1) + negl(n)$. The right inequality follows directly from what we have shown by letting $b = 1$. The other inequality can be seen by using the inequality we have shown with $b = 0$ as follows: $P(D^{F'_k}(n) = 1) = 1 - P(D^{F'_k}(n) = 0) \geq 1 - (P(D^{F_k}(n) = 0) + negl(n)) = P(D^{F_k}(n) = 1) - negl(n)$.