Cryptography, winter term 16/17: Sample solution to assignment 8

Cornelius Brand, Marc Roth

Exercise 8.1 (Be nice to your tutors and TAs, 1 Bonus Point) Write the name and matriculation number of every author as well as number, time slot and the name of the tutor of your tutorial group on the first page of your solution. Furthermore write in english and staple *all* sheets of your solution.

Exercise 8.2 (Hash functions, 2+2+2 Points) Let (Gen_1, H_1) and (Gen_2, H_2) be hash functions from which *at least one* is collision-resistant. Decide for the following constructions whether the resulting hash function is *necessarily* collision-resistant [1] and prove your answer.

(a) $H_a^{(s_1,s_2)}(x) := H_1^{s_1}(x)||H_2^{s_2}(x)$

(b) $H_b^{(s_1,s_2)}(x) := H_1^{s_1}(H_2^{s_2}(x))||H_2^{s_2}(H_1^{s_1}(x))$

(c) $H_c^{(s_1,s_2)}(x) := H_1^{s_1}(H_2^{s_2}(x)||x)||H_2^{s_2}(H_1^{s_1}(x)||x)$

Solution 8.2 (Hash functions, 2+2+2 Points) For reasons of notations we do not write the key explicitely, as it is fixed and known by the adversary.

(a) H_a is collision-resistant: As $H_a(x) = H_a(y)$ implies $H_1(x) = H_1(y)$ and $H_2(x) = H_2(y)$ any adversary that finds a collision for H_a can be used to construct an adversary that finds a collision for both H_1 or H_2.

(b) H_b is not collision-resistant: Assuming H_1 is the constant zero function (for all keys), it follows that $H_b(x) = 0||H_2(0)$ for any x.

(c) H_c is collision-resistant: To see this we assume that there is a ppt adversary that finds a collision x,y with non-negligible probability. We have $H_c(x) = H_c(y)$ which implies

$$H_1(H_2(x)||x) = H_1(H_2(y)||y) \tag{1}$$

and

$$H_2(H_1(x)||x) = H_2(H_1(y)||y)\,. \tag{2}$$

As $(H_2(x)||x) \neq (H_2(x)||x)$ and $(H_1(x)||x) \neq (H_1(x)||x)$ we found collisions for both H_1 and H_2. Therefore, the adversary can be used to construct an adversary that finds a collision for both hash functions which contradicts the fact that at least one of H_1 and H_2 is collision-resistant.

[1] In each case we assume that Gen runs Gen_1 and Gen_2 to obtain a key (s_1, s_2).

Exercise 8.3 (Random-Oracle model, 2+2+2 Points) Let (Gen, H) be a collision-resistant hash function with inputs of arbitrary size. We define a MAC for arbitrary-length messages by

$$\text{Mac}_{s,k}(m) = H^s(k||m)\,.$$

(a) Show that this is not a secure MAC if H is constructed by the Merkle-Damgard transform from an arbitrary collision-resistant hash function h. (We assume that s is known to the attacker.)

(b) Show that this MAC is secure if H is modeled as a random oracle.
Hint: You do *not* need to prove this by hand. Instead, use a property of random oracles that was introduced in the lecture and the canonical construction of a MAC (Theorem 4.6).

(c) Explain *briefly* the consequences of (a) and (b) for the soundness of the Random-Oracle model.

Solution 8.3 (Random-Oracle model, 2+2+2 Points) For reasons of notations we do not write the key explicitely, as it is fixed and known by the adversary.

(a) Let h be the collision-resisitant hash function from which H is constructed by applying the Merkle-Damgard transform. We show that the MAC is not secure by constructing an adversary: We first query an arbitrary message m of length n and obtain

$$t = \text{Mac}_k(m) = H(k||m) = h(h(0^n||k)||m)\,.$$

The adversary outputs $m' = m||t$ and $t' = h(t||t)$. Now it holds that

$$\begin{aligned}\text{Mac}_k(m') &= H(k||m')\\ &= H(k||m||t)\\ &= h(h(h(0^n||k)||m)||t)\\ &= h(t||t)\\ &= t'\end{aligned}$$

It follows that the adversary wins with probability 1, which is certainly not negligible.

(b) It is known from the lecture that the function $F_k(m) = H(k||m)$ is a PRF if H is modeled as a random oracle. Therefore, applying Theorem 4.6, the MAC is secure.

(c) The results above illustrate that a proof of security in the Random-Oracle model does not guarantee security for every practical instance. It is mere evidence. However, a proof of security in the Random-Oracle model is better than no proof at all.

Exercise 8.4 (One-way functions and hard-core predicates, 2+2 Points) Let f be a length-preserving one-way function. Define the function $g(x_1, x_2) = (x_1, f(x_2))$, where $|x_1| = |x_2|$.

(a) Show that g is one-way as well.

(b) Show that if f has a hard-core predicate, then so has g.

Solution 8.4 (One-way functions and hard-core predicates, 2+2 Points) (a) Suppose this was not the case. Obviously, g is computable in polynomial time if f is, so there is a probabilistic polynomial-time algorithm $\mathcal{A}$ and a positive polynomial $q(n)$ such that

$$\Pr[\mathsf{Invert}_{\mathcal{A},g}(n) = 1] \geq 1/q(n).$$

We construct a ppt-algorithm $\hat{\mathcal{A}}$ as follows: When $\hat{\mathcal{A}}$ is given 1^n and y as input, it uniformly chooses $x \leftarrow \{0,1\}^n$, and then simulates $\mathcal{A}$ on input 1^{2n} and (x, y). The output of $\hat{\mathcal{A}}$ is then simply the second half of the output of $\mathcal{A}$. Whenever $\mathcal{A}$ successfully inverts g on (x, y), i.e. it outputs (x, x') with $f(x') = y$, then $\hat{\mathcal{A}}$ successfully inverts y. Hence, the success probability of $\hat{\mathcal{A}}$ is at least that of $\mathcal{A}$, and hence non-negligible. Thus, f is not one-way, a contradiction.

(b) Let hc be a hard-core predicate of f. We claim that $\mathsf{hc}'(x_1, x_2) := \mathsf{hc}(x_2)$ is a hard-core predicate for g. Assume this is not the case. Again, hc' is obviously computable in polynomial time if hc is. Hence, there is probabilistic polynomial-time algorithm $\mathcal{A}$ and a positive polynomial $q(n)$, such that

$$\Pr[\mathcal{A}(1^n, g(x_1, x_2)) = \mathsf{hc}'(x_1, x_2)] \geq 1/2 + 1/q(n)\,.$$

By definition, this means

$$\Pr[\mathcal{A}(1^n, (x_1, f(x_2))) = \mathsf{hc}(x_2)] \geq 1/2 + 1/q(n)\,.$$

We construct again a ppt-algorithm $\hat{\mathcal{A}}$ as follows: When $\hat{\mathcal{A}}$ is given 1^n and $f(x)$ as input, it uniformly chooses $x_1 \leftarrow \{0,1\}^n$, and then simulates $\mathcal{A}$ on input 1^{2n} and $(x_1, f(x))$. Then,

$$\Pr[\hat{\mathcal{A}}(1^n, f(x)) = \mathsf{hc}(x)] = \Pr[\mathcal{A}(1^{2n}, (x_1, f(x))) = \mathsf{hc}'(x_1, x)] \geq 1/2 + 1/q(n)$$

showing that hc is not a hard-core predicate, a contradiction.