Cryptology I (spring 2014) Prof. Dr. Dominique Unruh

Exercise Sheet 6

Out: May 7, 2014 Due: May 22, 2014

Problem 1: ElGamal FDH

Bob studied the RSA-FDH construction. He notices that RSA-FDH essentially does the following: To sign a message m, it decrypts $H(m)$ using textbook RSA, and to check a signature σ, it encrypts σ and compares the result with $H(m)$.

This lead him to the following idea: Instead of textbook RSA, he uses ElGamal in the construction of FDH, because ElGamal is more secure (it is EF-CMA secure).

Why is the resulting scheme "ElGamal-FDH" bad?

Problem 2: Random oracle model

Write down the definition of IND-CPA security in the random oracle model (for symmetric encryption schemes).

Problem 3: One-way functions

Which of the following are one-way functions? Why (short argument, no proof)? (You may assume that the RSA assumption holds. And that E_{AES} is a PRF.)

Remember that to break a one-way function, it is sufficient to find some preimage, not necessarily the "true" one that was fed into the one-way function.

(a) $f(x) := 0$ for all $x \in \{0,1\}^\eta$.

(b) $f(x) := x_1 \dots x_{\eta/2}$ for $x \in \{0,1\}^\eta$.

(c) $f(N, e, x) := (N, e, x^e \bmod N)$ where the domain of f is the set of all (N, e, x) where N is an RSA modulus, e is relatively prime to N, and $x \in \{0, \dots, N-1\}$.

(d) $f(N, e, x) := x^e \bmod N$.

(e) $f(k, x) := E_{AES}(k, x)$.

(f) $f(x) := g(x) \| g(x)$ where g is a one-way function.

Note: Here (and in (g)), the question is whether f would be a one-way function for *every* one-way function g.

(g) $f(x) := g(g(x))$ where g is a one-way function.

Hint: The first thought here might be wrong. Remember that a one-way function g might not be surjective. E.g., the first half of $g(x)$ might always consist of zeroes.

Problem 4: Merkle-Damgård and the ROM

In the lecture, I explained the random oracle heuristic which suggests to model a hash function as a random oracle. It should be added that a (preferable) refinement of this heuristic is to model the compression function itself as a random oracle, and to model the hash function as some function constructed based on that compression function (using, e.g., Merkle-Damgård). The reason behind this is that constructions like Merkle-Damgård do not produce functions that behave like random functions (even if the underlying compression function is a random function).

Give an example why a hash function H constructed using the Merkle-Damgård construction should not be modeled as a random oracle. More precisely, find a cryptographic scheme which is secure when H is a random oracle (no security proof needed), but which is insecure when H is a Merkle-Damgård construction (even if the compression function is a random oracle).

Hint: You will not have to invent a new construction. A suitable example has already been discussed at some point in the lecture, you just have to identify it.

Problem 5: Security proof in the ROM [Bonus problem]

This is a bonus problem.

Fix a hash function $H : \{0,1\}^* \to \{0,1\}^\eta$. We define the following block cipher with message and key space $\{0,1\}^\eta$.:

- **Encryption:** To encrypt $m \in \{0,1\}^\eta$ under key k, choose a random $r \in \{0,1\}^\eta$ and return the ciphertext $c := (r, m \oplus H(k\|r))$.
- **Decryption:** To decrypt $c = (r, c')$ with key k, compute and return $m := H(k\|r) \oplus c'$.

Below is a proof that this encryption scheme is IND-CPA secure in the random oracle model. Fill in the gaps. (The length of the gaps is unrelated to the length of the text to be inserted.)

Proof. In the first game, we just restate the game from the IND-CPA security definition (in the random oracle model).

Game 1. [1] ◇

To show that the encryption scheme is IND-CPA secure, we need to show that

$$|\Pr[b = b' : \mathit{Game}\ 1] - \tfrac{1}{2}| \text{ is negligible} \tag{1}$$

As a first step, we replace the random oracle.

Game 2. Like Game 1, except that we define the random oracle H differently: [2] ◇

We have $\Pr[b = b' : Game\ 1] = \Pr[b = b' : Game\ 2]$.

One can see that the adversary cannot guess the key k (where k is the key used for encryption in Game 2), more precisely, the following happens with negligible probability: "The adversary invokes $H(x)$ with $x = k\|r'$ for some r'." (We omit the proof of this fact.)

Let r_0 denote the value r that is chosen during the execution of $c \leftarrow E^H(k, m_b)$ in Game 2. Consider the following event: "Besides the query $H(k\|r_0)$ performed by $c \leftarrow E^H(k, m_b)$, there is another query $H(x)$ with $x = k\|r_0$ (performed by the adversary or by the oracle $E^H(k, \cdot)$." This event occurs with negligible probability. Namely, the adversary make such $H(x)$ queries with negligible probability because [3], and the oracle $E^H(k, m_b)$ makes such $H(x)$ queries with negligible probability because [4].

Thus, the response of the $H(k\|r_0)$-query performed by $c \leftarrow E^H(k, m_b)$ is a random value that is used nowhere else (except with negligible probability). Thus, we can replace that value by some fresh random value.

Game 3. Like Game 2, except that we replace $c \leftarrow E^H(k, m_b)$ by $r_0 \xleftarrow{\$} \{0,1\}^\eta$, $h^* \xleftarrow{\$} \{0,1\}^\eta$, $c \leftarrow (r_0, m_b \oplus h^*)$. ◇

We have that $|\Pr[b = b' : Game\ 2] - \Pr[b = b' : Game\ 3]|$ is negligible.

To get rid of m_b in Game 3, we use the fact that h^* is chosen uniformly at random and XORed on m_b. That is, we can replace $m_b \oplus h^*$ by [5].

Game 4. Like Game 3, except that we replace $c \leftarrow (r_0, m_b \oplus h^*)$ by [6]. ◇

Notice that b is not used in Game 4, thus we have that $\Pr[b = b' : Game\ 4] =$ [7].

Combining the equations we have gathered, (1) follows. □