ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2021

Prof. R. Wattenhofer
Tejaswi Nadahalli and Ard Kastrati

Computational Thinking
Solutions to Exercise 5 (Cryptography)

1 Nonce Reuse

In the ElGamal digital signature scheme, recall that, for a random nonce x, $r = g^x \mod p$, $s_1 = (x \cdot h(m_1) - k_s \cdot r) \mod p-1$. The signature is (s_1, r). If a different message m_2 is signed with the same nonce/keypair, we have a signature (s_2, r) with $s_2 = (x \cdot h(m_2) - k_s \cdot r) \mod p-1$. Subtracting the two signatures, we have $s_2 - s_1 = x \cdot (h(m_2) - h(m_1)) \mod p-1$, giving $x = \frac{s_2 - s_1}{h(m_2) - h(m_1)} \mod p-1$. The attacker can then substitute the value of x and $r = g^x \mod p$ in the equation for either s_1 or s_2 to recover k_s.

2 Cryptographic Hash Functions

- $h_3(x)$ is not collision-resistant in general. By setting $h_1(x) = h_2(x)$, we get $h_3(x) = 0$ and thus clearly it is not collision-resistant.
- $h_4(x)$ is collision resistant. Let $h_4(x) = h_4(y)$ with $x \neq y$ a collision of h_4. Then, it follows that $x_0; h_1(x) = y_0; h_1(y)$. In particular, this means that $h_1(x) = h_1(y)$ and thus x and y is a collision for h_1. But h_1 is assumed to be collision-resistant and therefore those are hard to find.

3 IND-CPA

- Correctness.
  Decrypt'(Encrypt'(m, k_p), k_s) = Decrypt'((h_m, Encrypt(m, k_p)), k_s) = Decrypt(Encrypt(m, k_p), k_s) = m.
- The main idea is that a cryptographic hash function is hard to invert but it is not an encryption scheme and thus leaks information about the message. The adversary can always win the IND-CPA game: First, the adversary picks two messages m_0, m_1 with $h(m_0) \neq h(m_1)$ and sends those to the challenger. Then, when the adversary receives the ciphertext $(c_1, c_2) = (h(m_b)$,Encrypt$(m_b, k_p))$ from the challenger, she simply compares $h(m_b)$ to $h(m_0)$ and $h(m_1)$ and always outputs the correct bit.