ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing



HS 2026

Prof. R. Wattenhofer
Ard Kastrati

Computational Thinking
Solutions to Exercise 5 (Cryptography)

1 Nonce Reuse

Recall that in the ElGamal digital signature scheme, we choose a random nonce $x \in \{1, \ldots, p-2\}$, let $r = g^x \mod p$ and let $s_1 = (x \cdot h(m_1) - k_s \cdot r) \mod (p-1)$. 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 $s_2 - s_1 \equiv x \cdot (h(m_2) - h(m_1)) \pmod{(p-1)}$, giving $x \equiv \frac{s_2 - s_1}{h(m_2) - h(m_1)} \pmod{(p-1)}$. 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 $h_3(x) = 0$, which is clearly not collision-resistant.
- $h_4(x)$ is collision resistant. Suppose we have $h_4(x) = h_4(y)$, with $x \neq y$ colliding for h_4. Then, it follows that $x_0; h_1(x) = y_0; h_1(y)$. In particular, this implies $h_1(x) = h_1(y)$, which means that x and y collide for h_1. Since every collision for h_4 is also a collision for h_1, we conclude that h_4 is at least as collision-resistant as h_1.

3 ElGamal Encryption

We show that Breaking-ElGamal-Encryption $\leq$ CDH. Given some ElGamal ciphertext $(c_1, c_2) = (g^x, m \cdot g^{x \cdot k_s})$ and the public key $k_p = g^{k_s}$, we create the pair $(g^a = g^x, g^b = g^{k_s})$ as a CDH problem instance and get $g^{ab} = g^{x \cdot k_s}$ by solving this instance. We can then extract m by computing

$$c_2 \cdot \left(g^{x \cdot k_s}\right)^{p-2} = m \cdot g^{x \cdot k_s} \cdot \left(g^{x \cdot k_s}\right)^{p-2} = m \cdot \left(g^{x \cdot k_s}\right)^{p-1} = m.$$

4 Active Adversary in ElGamal Encryption

a) Refer to Lemma 3.30 in the lecture notes.

b) An attacker can change (c_1, c_2) to (c_1, c_2') where $c_2' = 2^{-1} \cdot c_2 = 2^{p-2} \cdot c_2 \mod p$. After decryption, everything mod p we have

$$m' = c_2' \cdot {c_1}^{k_s \cdot (p-2)} = 2^{p-2} \cdot c_2 \cdot {c_1}^{k_s \cdot (p-2)} = 2^{p-2} \cdot 2k = 2^{p-1} \cdot k = k$$

A higher-level description: Alice sent $E(2k)$, the encryption of $2k$. The attacker intercepts the value $E(2k)$ and blocks her request. The attacker then encrypts $\frac{1}{2} = 2^{-1} = 2^{p-2} \mod p$ itself to get $E(2^{p-2})$. Finally, it multiplies $E(2k)$ with $E(2^{p-2})$ to obtain $E(k)$, which works because of the homomorphic property. The value $E(k)$ is sent as if Alice had intended it.

c) There exist encryption schemes (non-malleable schemes) which are not vulnerable against this sort of attack where an attacker modifies a ciphertext into another one that decrypts to a related plaintext. However, generally speaking, in order to achieve secure communication in the presence of active adversaries we have to have authentication. This can be achieved by using for example digital signatures. If the adversary changes any of Alice's signed messages, then it cannot forge signatures for the changed messages.