Exercises Week 1

Jordi Nadeu Ferran

Exercise 1

Let $n = pq$, with p, q primes. The goal of this exercise is to show that knowing $\varphi(n)$ is equivalent to knowing p, q. In particular, you need to show the following:

- Show that given n and $\varphi(n)$ it is possible to find p and q without factorizing.
- Show that given n, p, q it is possible to compute $\varphi(n)$.

(a) Finding p and q given n and $\varphi(n)$

The Euler's function is defined as:

$$\varphi(n) = (p-1)(q-1) \tag{1}$$

Given $n = pq$ and $\varphi(n)$, we solve for p and q as follows:

$$\begin{aligned} p + q &= n - \varphi(n) + 1 \\ pq &= n \end{aligned}$$

This gives the quadratic equation:

$$x^2 - (n - \varphi(n) + 1)x + n = 0 \tag{2}$$

Solving for x:

$$p, q = \frac{(n - \varphi(n) + 1) \pm \sqrt{(n - \varphi(n) + 1)^2 - 4n}}{2} \tag{3}$$

Then knowing this we can determine p and q without direct factorization.

(b) Computing $\varphi(n)$ given n, p, and q

By definition,

$$\varphi(n) = (p-1)(q-1) \tag{4}$$

which follows directly from knowing p and q.

Exercise 2

Let (Enc, Dec) be a deterministic asymmetric encryption scheme, such that given a public key pk and a message m, the encryption $Enc(m, pk)$ is unique. Show that (Enc, Dec) is not IND-CPA secure. To do this we show how an attacker A can win the IND-CPA game below:

1. Setup phase: $pp \leftarrow \text{Setup}(\lambda)$
2. Key generation: $(pk, sk) \leftarrow \text{KeyGen}(\lambda)$
3. Choose a random bit: $b \leftarrow \{0, 1\}$
4. Attacker chooses messages: $(m_0, m_1) \leftarrow A^{OEnc}(\lambda, pk)$
5. Encryption: $ct \leftarrow Enc(pk, m_b)$
6. Attacker guesses b: $b' \leftarrow A^{OEnc}(pk, ct, m_0, m_1)$
7. Return $b = b'$.

An attacker A in the IND-CPA game can:

1. Choose two messages m_0 and m_1.
2. Receive the encryption $c = \text{Enc}(m_b, pk)$.
3. Compute $\text{Enc}(m_0, pk)$ and $\text{Enc}(m_1, pk)$.
4. Compare c with both computed values to determine b.

Since encryption is deterministic, the attacker correctly guesses b with probability 1, violating IND-CPA security.

Exercise 3

Let $(g, h = g^x)$ be a public key for the ElGamal encryption scheme. Let (g^r, mh^r) be an encryption of m.

Show that if an adversary knows r, then it can find m.

Given an ElGamal encryption (g^r, mh^r) and knowing m, an attacker can recover the secret key:

$$h = g^x \Rightarrow mh^r = mg^{xr} \tag{5}$$

Rearranging:

$$\frac{mh^r}{m} = g^{xr} \tag{6}$$

Taking logarithms:

$$x = \frac{\log_g(mh^r) - \log_g m}{r} \tag{7}$$

Then knowing this we can say knowing m allows an adversary to compute x and break the encryption.

Exercise 4

Let $(g, h = g^x)$ be a public key for the ElGamal signature scheme. Suppose that a signer uses the same k twice, i.e., it produces two signatures (r, σ_1) and (r, σ_2) for two different messages m_1, m_2 using the same k (and thus the same r) for both of them. What happens?

If the same random value k is reused in two ElGamal signatures (r, σ_1) and (r, σ_2) for different messages m_1 and m_2, then:

$$\sigma_1 = (m_1 - xr)k^{-1}$$
$$\sigma_2 = (m_2 - xr)k^{-1}$$

Subtracting both equations:

$$\sigma_1 - \sigma_2 = (m_1 - m_2)k^{-1} \tag{8}$$

Solving for k:

$$k = (m_1 - m_2)(\sigma_1 - \sigma_2)^{-1} \tag{9}$$

Once k is known, the secret key x can be found:

$$x = \frac{m_1 - k\sigma_1}{r} \tag{10}$$

Then knowing this we can say reusing k completely breaks the ElGamal signature scheme.