
TD 5 – Public-key encryption

Exercise 1.*Generalities*

1. Let $(\text{Gen}, \text{Enc}, \text{Dec})$ be a public-key encryption scheme for fixed-length messages. Prove that an unbounded adversary that is given the public key pk and a ciphertext $c \leftarrow \text{Enc}_{pk}(m)$ can compute m with success probability 1.
2. Let $(\text{Gen}, \text{Enc}, \text{Dec})$ be an IND – CPA secure public-key encryption scheme. Assume that the length of a ciphertext c is uniquely determined by the length of the plaintext m .
 - i. Why must $|c|$ be larger than $|m|$?
 - ii. Let $\ell = |c| - |m|$. Give a time bound t such that the advantage $\text{Adv}_{\text{Enc}}^{\text{IND-CPA}}(t)$ is 1.
 - iii. More generally, give a lower bound on $\text{Adv}_{\text{Enc}}^{\text{IND-CPA}}(t)$ in terms of t and ℓ .
3. We have seen that ElGamal encryption scheme is malleable: Given $c \leftarrow \text{Enc}_{pk}(m)$ and any $\alpha \in G$, it is possible to compute c' such that $\text{Dec}_{sk}(c') = \alpha \cdot m$ without knowing m nor sk .
 - i. Recall how to build c' from c .
 - ii. In the previous construction, the first component of c' is the same as the first component of c . An observer may find this suspicious. Show how to build c'' such that $\text{Dec}_{sk}(c'') = \alpha \cdot m$, such that c and c'' share no component.

Exercise 2.*ElGamal re-encryption*

Let G be a cyclic group of *prime* order q , with generator g .

1.
 - i. Recall how ElGamal encryption scheme works: Describe the three algorithms Gen, Enc and Dec.
 - ii. Prove its correction.
 - iii. Recall under which hypothesis is the scheme IND – CPA secure. Is it IND – CCA secure?

We define a variant of ElGamal encryption where the ciphertext for a message m is the pair $c = (m \cdot g^y, h^y)$ where $h = g^x$ is the public key and $y \leftarrow \{0, \dots, q - 1\}$ uniformly.

2.
 - i. Describe the decryption algorithm for this variant, and analyze its complexity. *Hint.* The private key can be inverted modulo q (using which algorithm?).
 - ii. Justify that this variant has the same security as the original scheme.

We are interested in *re-encryption*. As an example, imagine a user Alice that has two distinct email addresses on a same server (say a professional one and a personal one). Each mailbox is encrypted with the variant of ElGamal encryption scheme, using the same group G and same generator g . Alice has two pairs of keys (sk_1, pk_1) and (sk_2, pk_2) . In the first mailbox, each email is encrypted with pk_1 and in the second one with pk_2 . Alice would like the server to be able to move an encrypted email c_1 from the first mailbox to the second one: For, the server must *re-encrypt* c_1 with the second public key. Formally, given c_1 , the server must compute c_2 such that $\text{Dec}_{sk_1}(c_1) = \text{Dec}_{sk_2}(c_2)$.

3. Propose an obvious solution if the server knows the private keys of Alice. What is the drawback?
4. Alice provides a *re-encryption key* $r_{1 \rightarrow 2} = sk_2 \cdot sk_1^{-1} \pmod q$ to the server.
 - i. Why does this key give no information on sk_1 and sk_2 to the server?
 - ii. Prove that the server can re-encrypt a ciphertext c_1 into a ciphertext c_2 such that $\text{Dec}_{sk_1}(c_1) = \text{Dec}_{sk_2}(c_2)$, using only the re-encryption key.