Assignment 4 Q4

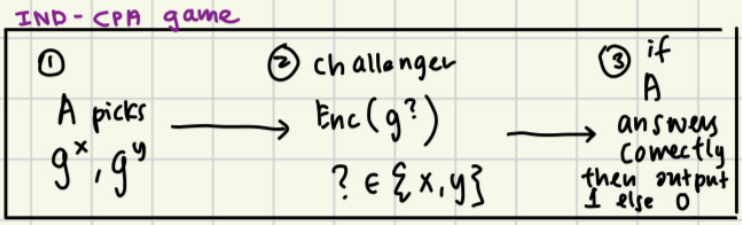
watIAM: sh2yap
student ID: 21111395

a) The encryption is trivial, as the attacker has public access to the public key, thus access to an "encryption oracle" is trivial. Attacker just needs to perform modular arithmetic to encrypt.
b) IND-CPA and IND-CCA for Elgamal PKES
    a. IND-CPA security game
        i. Setup: globally public prime p, globally public element g ∈ $\mathbb{Z}_p^*$ of large prime order q
        ii. The challenger chooses x ∈ $\mathbb{Z}_q$ and set:
            1. $k_{public} = g^x mod\ p$
            2. $k_{private} = x$
        iii. The adversary can select plaintexts m and random integer r to obtain corresponding ciphertext c = E(m) = $(g^r, m * (g^x)^r)\ mod\ p$
        iv. The adversary picks two messages m_0 and m_1 of the same length
        v. The challenger picks uniformly at random one of the messages, and encrypts it, the encrypted message is c.
        vi. The adversary guesses which is which message m_0 or m_1 corresponds to the encrypted message c.
        vii. If the guess is correct, the adversary wins.
    b. IND-CCA security game
        i. Same Setup as IND-CPA (step a.i, a.ii)
        ii. The adversary can select ciphertexts c and random integer x ∈ $\mathbb{Z}_q$, and feed into a Decryption oracle to obtain the corresponding plaintexts m. Note that, the adversary cannot ask to decrypt the challenge ciphertext.
        iii. The adversary picks two messages m_0 and m_1 of the same length
        iv. The challenger picks uniformly at random one of the messages, and encrypts it, the encrypted message is c.
        v. The challenger presents c to the adversary.
        vi. Under IND-CCA2: Adversary can "perform additional operations in polynomial time, **including calls to the oracles, for ciphertexts different than** c." (Source)
        vii. The adversary guesses which is which message m_0 or m_1 corresponds to the encrypted message c.
        viii. If the guess is correct, the adversary wins.

c) With reference to https://people.eecs.berkeley.edu/~daw/teaching/cs276-s06/l19.pdf

Assignment 4 Q4

Assume by contradiction, that we have an adversary A that breaks ElGamal IND-CPA security game, by real-or-random definition.



probability that A correctly distinguishes the real scenario

probability that A distinguishes correctly in a random simulated scenario

$$Adv\,A = \left| Pr[A^{E_{pk}}(pk) = 1] - Pr[A^{E_{pk} \circ \$}(pk) = 1] \right|$$

interaction of A with real ElGamal encryption oracle.

interaction of A with a simulated oracle that combines Elgamal encryption with a random oracle ($)

Suppose adversary A runs in time t and $Adv\,A = \varepsilon$.

We construct an algorithm B that solves DDH

Algorithm B B as follows:

B(a, b, c)

1) Run $A^{E_B}(a)$, where B's version of the encryption oracle E_B answers its one query m with $(b, c \cdot m)$

2) Output the same result as A does.

Assignment 4 Q4

There're 2 cases in which (a,b,c) can be inputted

CASE 1

(g^x, g^r, g^{rx})

A interacts with a "real" encryption oracle. So $B(g^x, g^r, g^{rx}) = A^{E_{pk}}(pk)$

i.e. the problem reduces to solving $A^{E_{pk}}(pk)$, because we can see that g^{rx} is not entirely random, and thus, if A can distinguish b/w them, A^{E_B} can also distinguish b/w them.

CASE 2:

(g^x, g^r, g^z)

Since g^z is selected uniformly at random, $g^z \cdot m$ is also a uniform random value & is indistinguishable from $g^{rx} \cdot m$

∴ making it a random oracle with $B(g^x, g^r, g^z) = A^{E_{pk=\$}}(pk)$

This turns Algorithm B that breaks El-Gamal's DDH property into one that breaks IND-CPA.

∴ if Elgamal can be broken by IND-CPA game (IND-CPA insecure) → the DDH property of Elgamal can also be broken (DDH is not hard)

∴ proved by contrapositive #

d)

d) Show that Elgamal does not satisfy IND-CCA security.

public key: g^x r is random
private key: x

1) Adversary chooses some messages $m_0 = \alpha$ and $m_1 = y$
2) Sends (m_0, m_1) to challenger
3) Challenger returns $\langle C_0, C_1 \rangle$, which is the encryption of m_b, $b \xleftarrow{\$} \{0, 1\}$
4) Adversary picks some z, then sends $\langle C_0, zC_1 \rangle$ to the decryption oracle. This works $\because zC_1$ is <u>not</u> the challenge cipher-text
5) The decryption oracle returns either $z \cdot \alpha$ or $z \cdot y$, so the attacker can know which message has been encrypted, by dividing by z.