Exercise 9.1

Task

Theorem

If the DDH problem is hard, then the El Gamal cryptosystem is CPA secure.

Problem

Prove the above theorem.

1. Construct an algorithm `B` that uses `Adv` as subroutine and attempts to solve DDH. Concretely, this means that `B` gets as input $G, \alpha, \alpha^a, \alpha^b, \alpha^c$, and must eventually output a guess, either c is random or $c = ab$.
2. Show that your algorithm achieves advantage at least ϵ. The conclusion is that if `Adv` is polynomial time and ϵ is not negligible, the existence of `B` demonstrates that DDH cannot be hard, we have a contradiction, and so such an adversary cannot exist.

Solution

We can prove the theorem by contradiction. Let's assume that an adversary `Adv` that plays the CPA security game with an advantage at least ϵ exists. Then, we'll show that we can construct a polynomial time algorithm that answers the DDH problem using that adversary. If we can do this we'll get that the DDH problem is not hard, which will be a contradiction.

1. Construct an algorithm to solve DDH

We have an adversary `A`, which generates a message `m` and the CPA oracle `O`. `O` either encrypts the message it receives under El Gamel or encrypts a random message of the same length and sends it back to `A`. This is all things we know, now let's see how we can adapt this adversary-oracle situation to solve DDH.

We replace the oracle with our custom implementation `O'`, which will interact with the adversary `A` in the same way but at the end will answer the DDH problem based on the adversary's result.

1. `O'` is given at the start $G, \alpha, \alpha^a, \alpha^b, \alpha^c$. `O'` chooses α^a as its public key and sends it to `A`. (In the notes the public key is noted as α^r where r is uniformly chosen in Z_t)
2. `A` generates a message m and sends that message to `O'`.
3. `O'` returns to `A` the tuple $(\alpha^b, \alpha^c \times m)$
4. Now `A` does its magic and returns either `real` or `ideal`.
    1. If `A` outputs `real` it means `A` thinks that the message has been properly encrypted (and `A` can break that encryption), which would be the case if $\alpha^c = \alpha^{ab}$, so we output `YES` for DDH.
    2. If `A` outputs `ideal` it means `A` thinks the message is random garbage, which would be the case when $\alpha^c \neq \alpha^{ab}$ (i.e. c is some random number). We output `NO` for DDH.

2. Show that the algorithm achieves advantage at least ϵ

The advantage of the above algorithm (which maps directly to the CPA definition) would be:

$$\mathrm{Adv} = |\mathrm{P[real|real]} - \mathrm{P[real|ideal]}| =$$

$$|\mathrm{P}[\mathrm{real}|\alpha^c = \alpha^{ab}] - \mathrm{P}[\mathrm{real}|\alpha^c : \mathrm{c \ is \ random}]| =$$

We know that in the case $\mathrm{real|real}$ A has an advantage ϵ, hence $\mathrm{P[real|real]} = \frac{1}{2} + \epsilon$. In the $\mathrm{real|ideal}$ case, the adversary has no advantage because c in α^c is uniformly chosen in Z_t and therefore reveals no information whatsoever. Therefore $\mathrm{P[real|ideal]} = \frac{1}{2}$. From that, we get that the advantage of the above algorithm is the same ϵ

$$\mathrm{Adv} = |\mathrm{P[real|real]} - \mathrm{P[real|ideal]}| = |\frac{1}{2} + \epsilon - \frac{1}{2}| = \epsilon$$