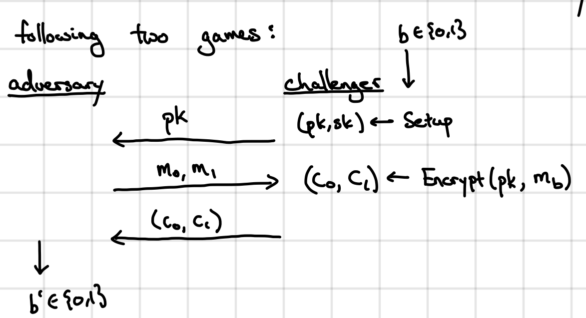
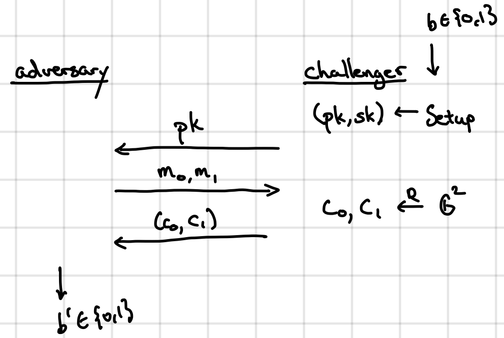
Security: If DDH holds in $\mathbb{G}$, then ElGamal is semantically secure.

Proof. Consider following two games:

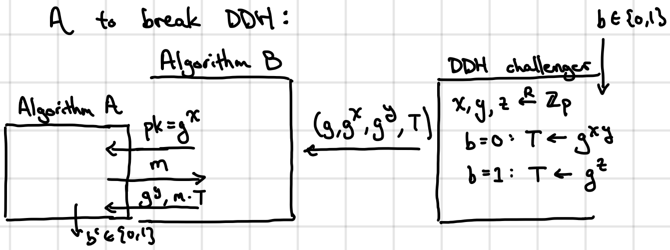




adversary's advantage in guessing b is 0 here since (c_0, c_1) is independent of (m_0, m_1)!

Claim: these two games are indistinguishable under DDH

Proof. Suppose there exists efficient $\mathcal{A}$ that can distinguish $(c_0, c_1) \leftarrow$ Encrypt(pk, m) from $(c_0, c_1) \xleftarrow{R} \mathbb{G}^2$. We use $\mathcal{A}$ to break DDH:



Observe: x is uniform over $\mathbb{Z}_p$ so g^x is a properly-generated public key (for ElGamal)

if $T = g^{xy}$, then $(g^y, T \cdot m) = (g^y, g^{xy} \cdot m)$ which is the output of Encrypt(pk, m) with randomness y — this is exactly the distribution where $\mathcal{A}$ sees Encrypt(pk, m)

if $T = g^z$, then $(g^y, g^z \cdot m)$ is uniform over $\mathbb{G}^2$ (since y, z are sampled independently of each other and of m) — this is exactly the distribution where $\mathcal{A}$ sees $(c_0, c_1) \xleftarrow{R} \mathbb{G}^2$

distinguishing advantage of $\mathcal{B}$ = distinguishing advantage of $\mathcal{A}$

Equivalent view: Under DDH, g^{xy} looks uniform even given g, g^x, g^y, so an ElGamal ciphertext looks indistinguishable (to an efficient adversary) from a OTP encryption

What if we want to encrypt longer messages? [or messages that is not a group element]

- Hybrid encryption (key encapsulation [KEM]):
  - Use PKE scheme to encrypt a secret key
  - Encrypt payload using secret key + authenticated encryption

  called key encapsulation

  PKE. Encrypt(pk, k) "header" [slow]

  AE. Encrypt(k, m) "payload" [fast]

  secret-key operations much much faster than public-key operations!
- How to derive key from group element?

  Same as in key-exchange: hash the group element to a bit-string (symmetric key)

  e.g., Hash-ElGamal: Encrypt(pk, m): $y \xleftarrow{R} \mathbb{Z}_p$

  $$c = (g^y, m \oplus H(g, h, g^y, h^y))$$

  as before, can also rely on CDH + ideal hash function (random oracle)

  $H: \mathbb{G}^4 \rightarrow \{0,1\}^n$

Vanilla ElGamal described above is <u>not</u> CCA-secure!

Ciphertexts are malleable: given $ct = (g^y, h^y \cdot m)$, can construct ciphertext $(g^y, h^y \cdot m \cdot g)$ which decrypts to message $m \cdot g$

↳ directly implies a CCA attack

Several approaches to get CCA security from DH assumptions:

- Cramer-Shoup (CCA-security from DDH) - based on hash-proof systems
- Fujisaki-Okamoto transformation (using an ideal hash function + CDH)
- Make stronger assumption ("interactive" CDH + use ideal hash function):

We do <u>not</u> know of any groups where CDH believed to be hard, but interactive CDH is easy.

"CDH is hard even given access to a DDH oracle"

also called strong DH assumption

  - Setup: $x \xleftarrow{R} \mathbb{Z}_p$, $h \leftarrow g^x$ $\quad$ pk: h, sk: x
  - Encrypt(pk, m): $y \xleftarrow{R} \mathbb{Z}_p$ $\quad k \leftarrow H(g, g^x, g^y, h^y)$ $\quad ct' \leftarrow \mathrm{Enc}_{AE}(k, m)$

    $c \leftarrow (g^y, ct')$

    symmetric authenticated encryption scheme
  - Decrypt(sk, c): $k \leftarrow H(g, g^x, c_0, c_0^x)$

    $m \leftarrow \mathrm{Dec}_{AE}(k, c_1)$

Essentially ElGamal where key derived from hash function