CSCI 1510

- Factoring / RSA & DLOG / CDH / DDH Assumptions (continued)
- Key Exchange Definition & Construction
- Public-key Encryption Definitions
- El Gamal / RSA Encryption
- Trapdoor Permutation

Factoring Assumption

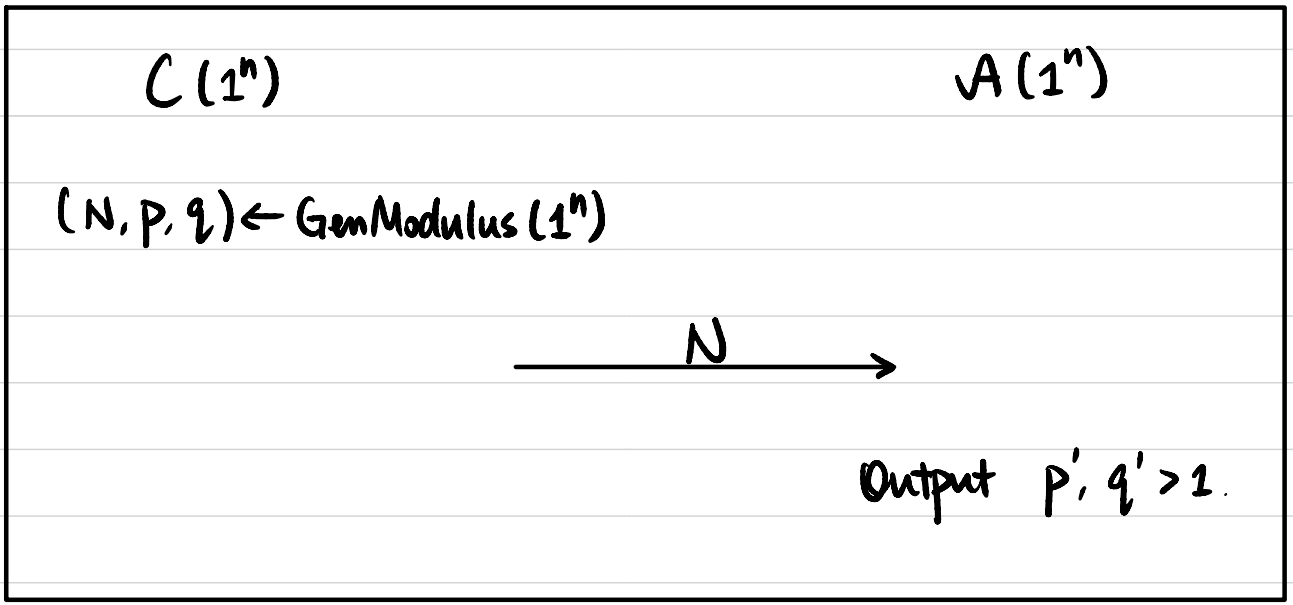
randomly sample → primality test

GenModulus (1^n): PPT algorithm, generates (N, p, q) ← How to generate?

p, q: n-bit primes, $p \neq q$. $N = p \cdot q$

Def Factoring is hard relative to GenModulus if

$\forall$ PPT A, $\exists$ negligible function $\varepsilon(\cdot)$ s.t. $\Pr[p' \cdot q' = N] \leq \varepsilon(n)$.



Factoring ⇒ OWF (GenModulus)

RSA Assumption

GenModulus (1^n): generates (N, p, q). p, q: n-bit primes, $p \neq q$. $N = p \cdot q$

GenRSA(1^n):

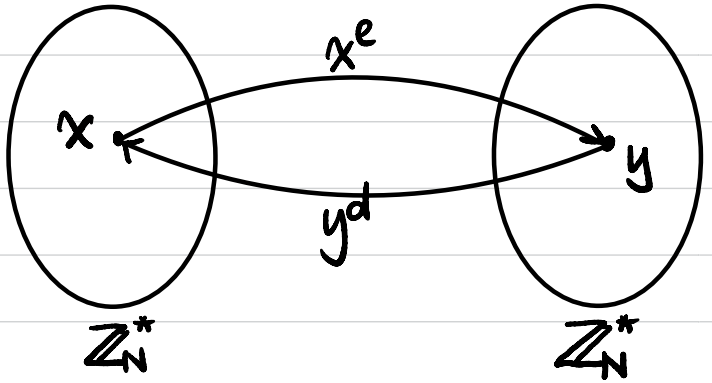
$(N, p, q) \leftarrow$ GenModulus (1^n)

$\phi(N) := (p-1)(q-1)$

Choose $e > 1$ s.t. $\gcd(e, \phi(N)) = 1$

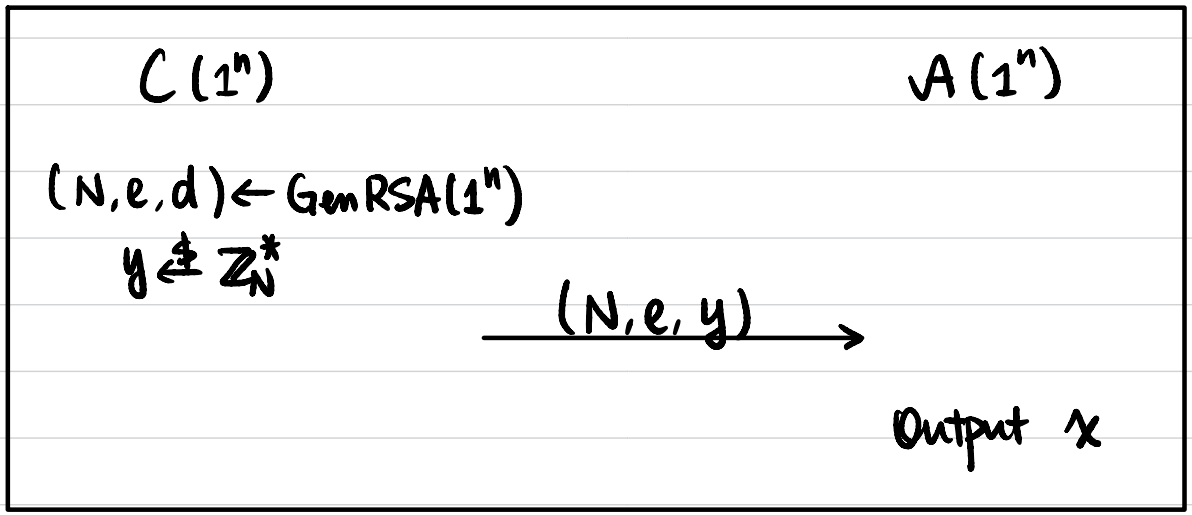
Compute $d := e^{-1} \bmod \phi(N)$

Output (N, e, d)



Def The RSA problem is hard relative to GenRSA if

$\forall$ PPT $\mathcal{A}$, $\exists$ negligible function $\varepsilon(\cdot)$ s.t. $\Pr[x^e = y \bmod N] \leq \varepsilon(n)$.



RSA $\Rightarrow$ Factoring

Discrete-Log Assumption

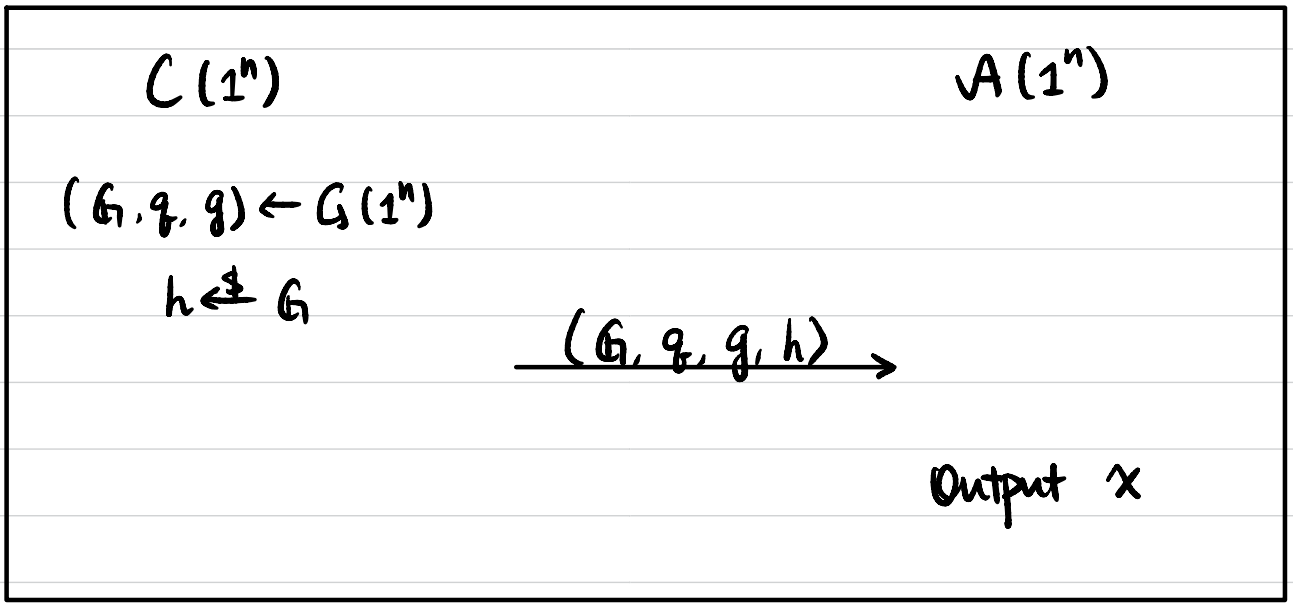
$G(1^n)$: PPT algorithm, generates (G, q, g)

description of a cyclic group G of order q with generator g.

(q: n-bit integer)

Def Discrete-Log (DLOG) is hard relative to G if

$\forall$ PPT A, $\exists$ negligible function $\varepsilon(\cdot)$ s.t. $\Pr[\, g^x = h \,] \leq \varepsilon(n)$.



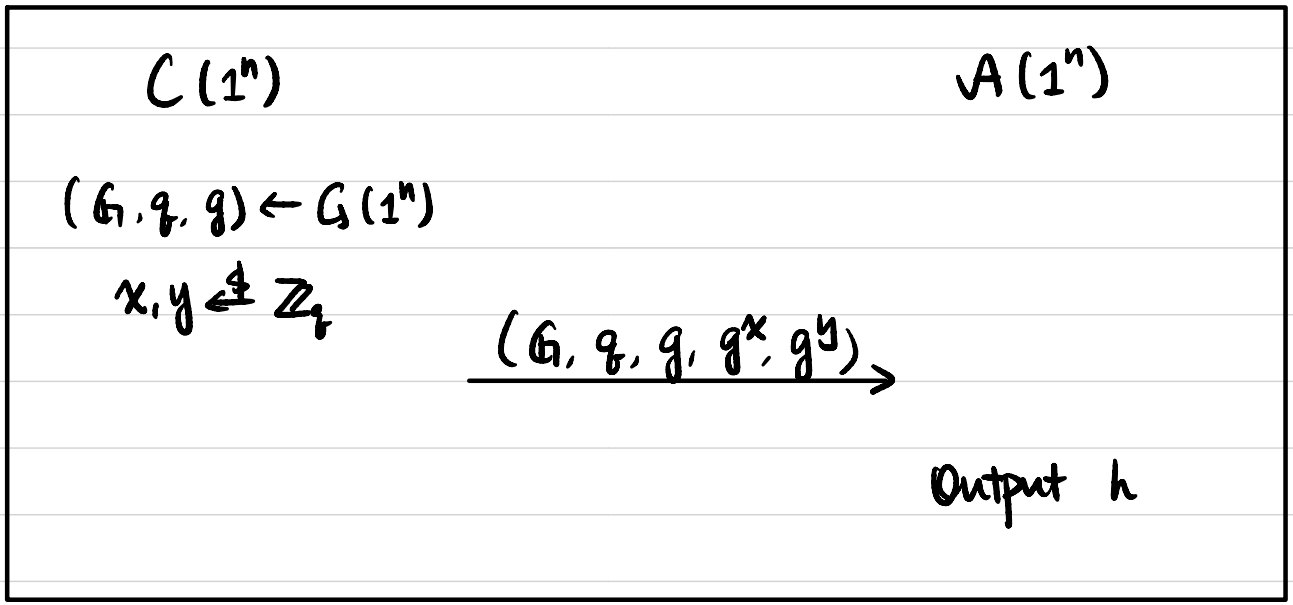
DLOG $\Rightarrow$ CRHF

Computational Diffie-Hellman (CDH) Assumption

$G(1^n)$: PPT algorithm, generates (G, q, g)

Def CDH is hard relative to G if

$\forall$ PPT A, $\exists$ negligible function $\varepsilon(\cdot)$ s.t. $\Pr[h = g^{xy}] \leq \varepsilon(n)$.



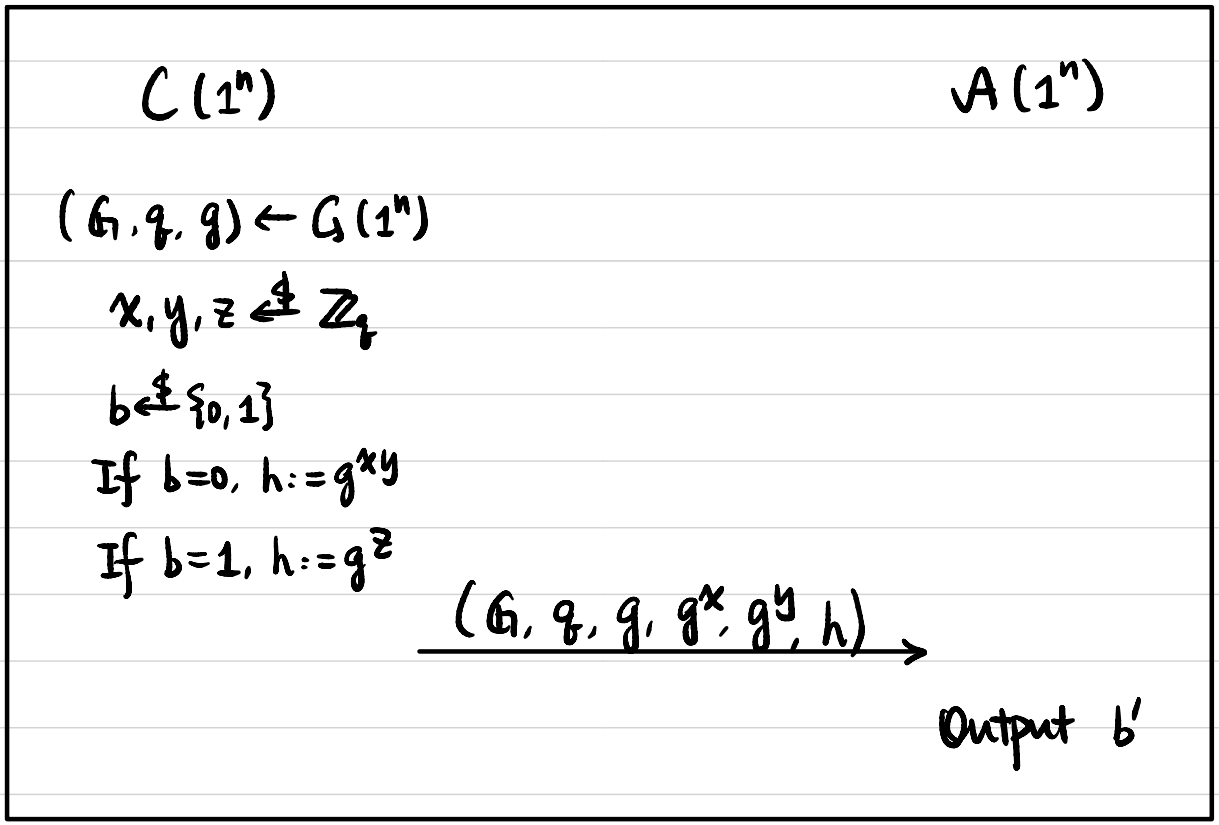
CDH ⇒ DLOG

Decisional Diffie-Hellman (DDH) Assumption

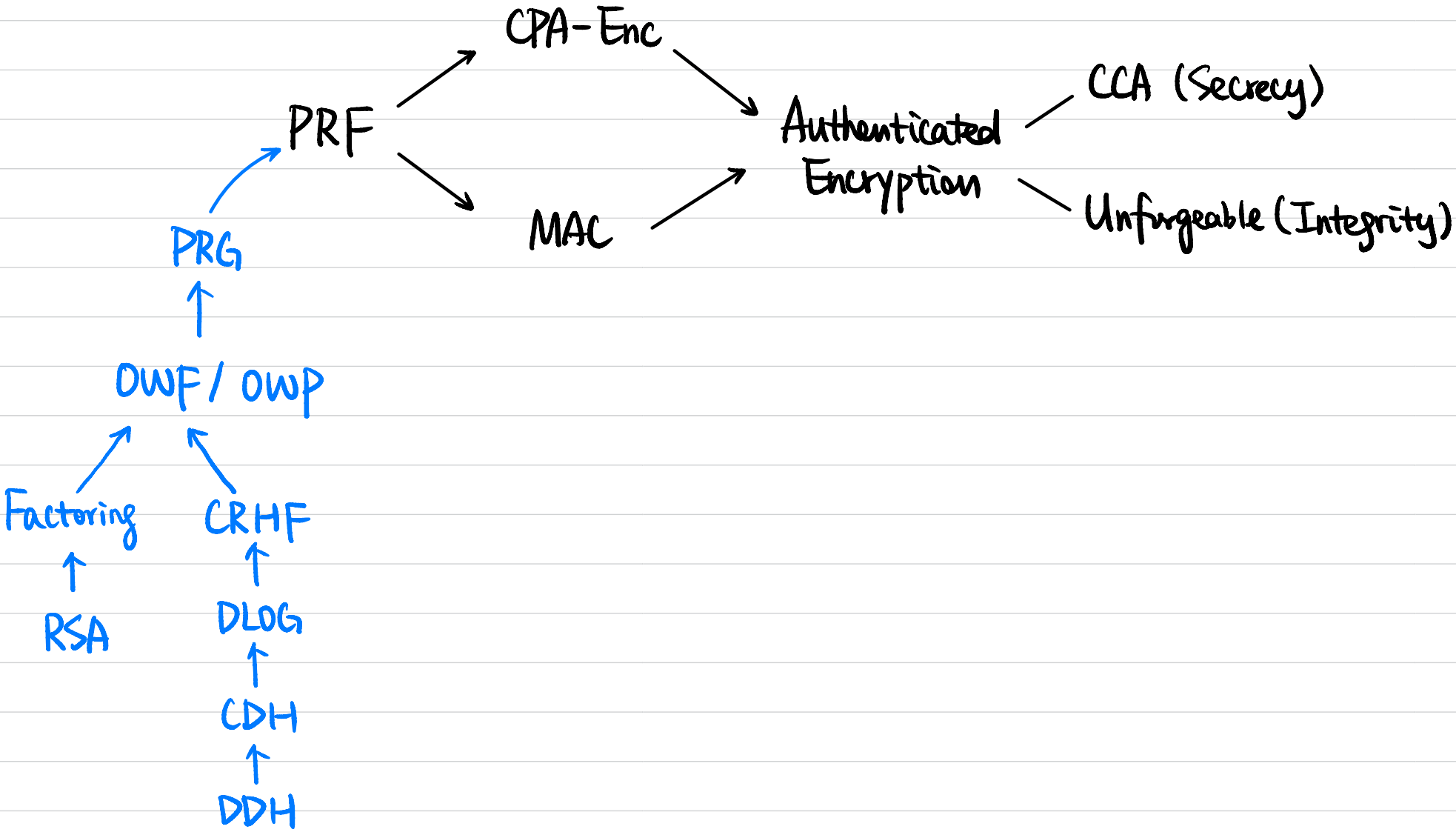
$\mathcal{G}(1^n)$: PPT algorithm, generates (G, q, g)

Def DDH is hard relative to $\mathcal{G}$ if

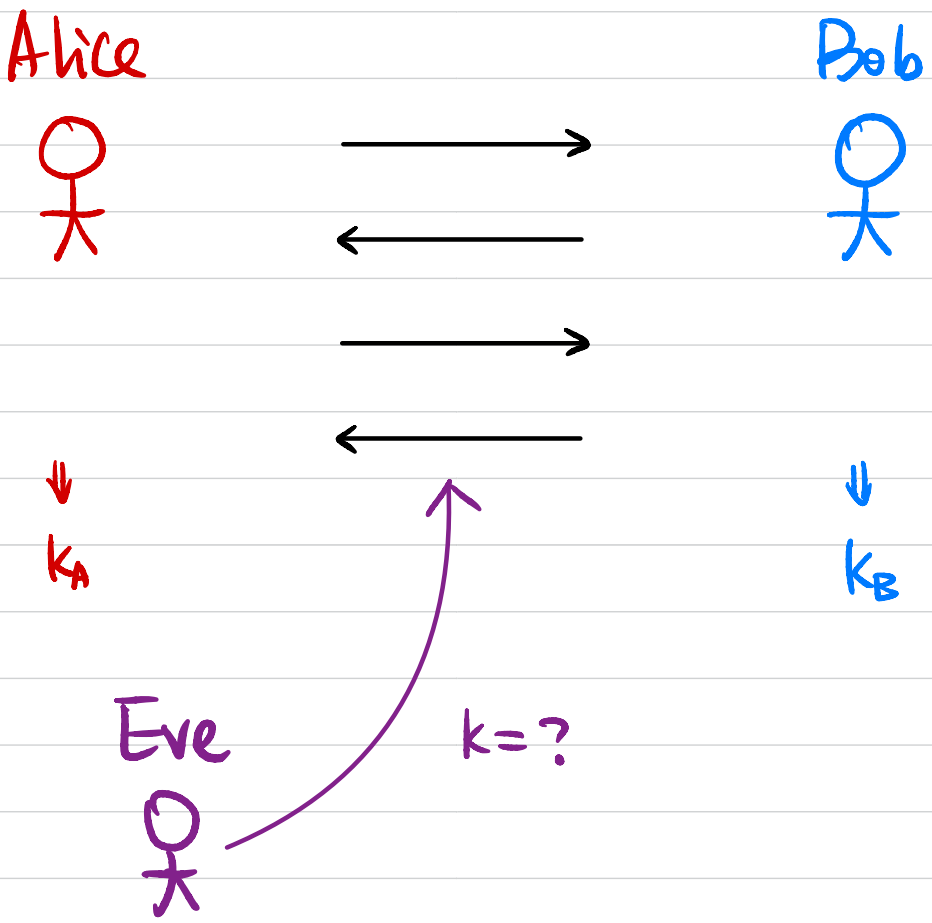
$\forall$ PPT $\mathcal{A}$, $\exists$ negligible function $\varepsilon(\cdot)$ s.t. $\Pr[b = b'] \leq \frac{1}{2} + \varepsilon(n)$.



DDH $\Rightarrow$ CDH


CPA-Enc
PRF
MAC
Authenticated Encryption
CCA (Secrecy)
Unforgeable (Integrity)
PRG
OWF / OWP
Factoring
CRHF
RSA
DLOG
CDH
DDH

Key Exchange



- **Correctness:** $k = k_A = k_B$
- **Security (Informally):** Eve listening on the channel shouldn't be able to guess k.

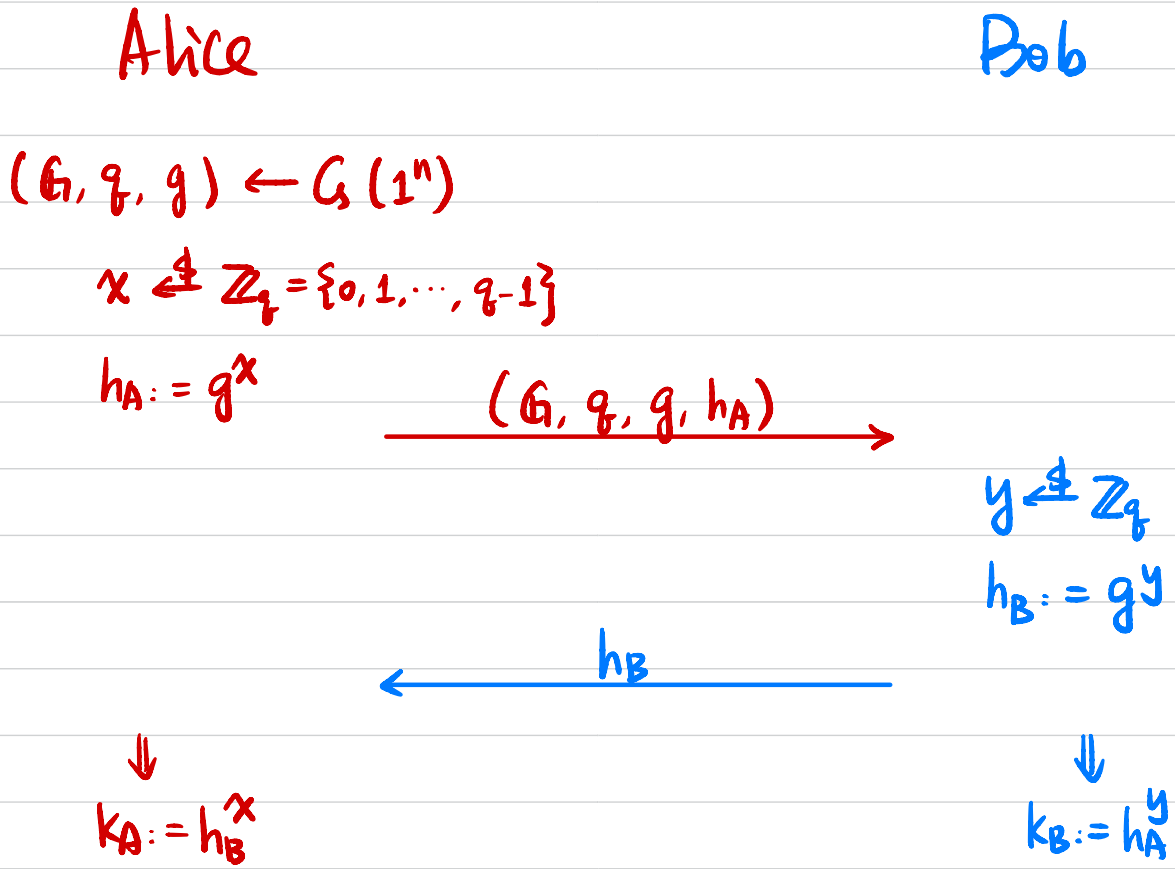
Key Exchange: Security

Def A key exchange protocol Π is secure if

$\forall$ PPT A, $\exists$ negligible function $\varepsilon(\cdot)$ s.t. $\Pr[b = b'] \leq \frac{1}{2} + \varepsilon(n)$.

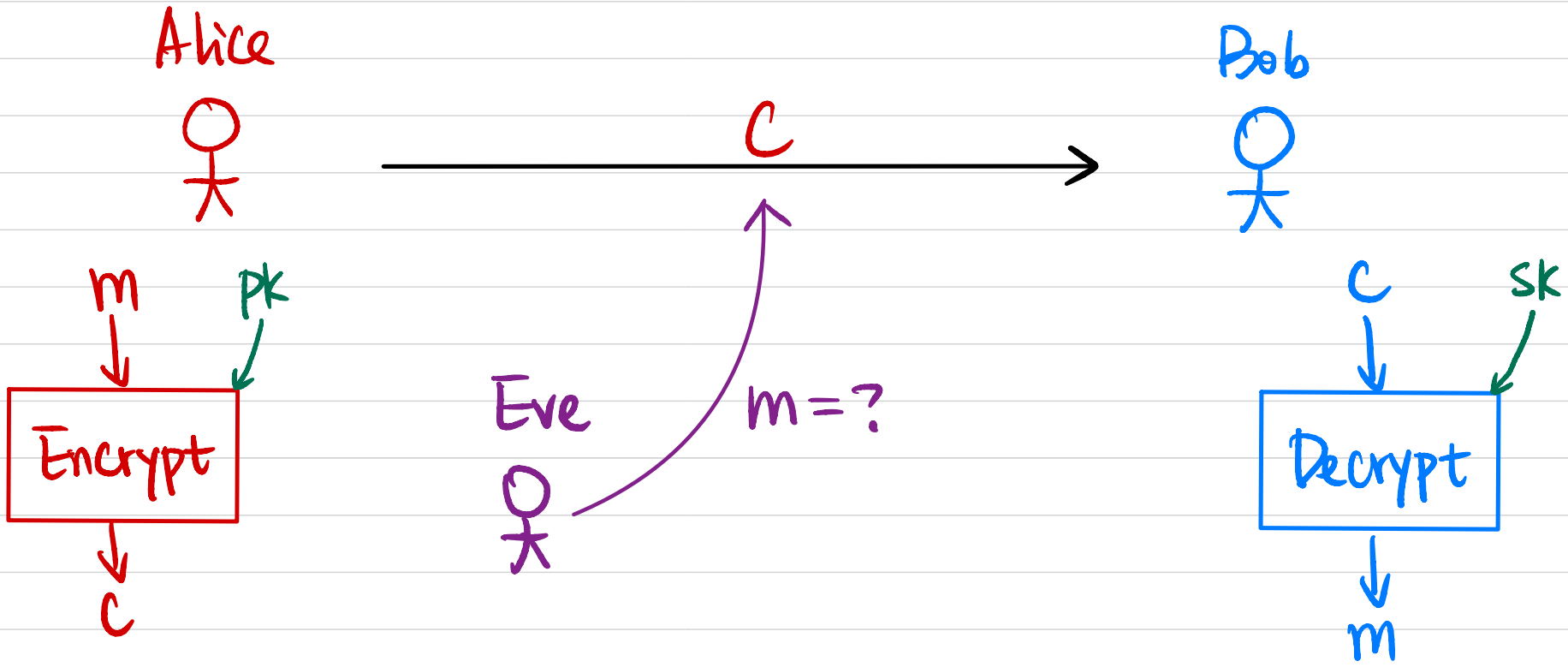
$C(1^n)$	$A(1^n)$
Two parties holding 1^n execute Π.	
$\Rightarrow$ transcript T containing all the messages & a key k output by each party.	
$b \xleftarrow{\$} \{0,1\}$	
If $b=0$, $\hat{k} := k$	
If $b=1$, $\hat{k} \xleftarrow{\$} \{0,1\}^n$	
$\xrightarrow{(T, \hat{k})}$	
	output b'

Diffie-Hellman Key Exchange



Thm If DDH is hard relative to G, then this is a secure key exchange protocol.

Public-Key Encryption



Public-Key Encryption

- **Syntax:**

  A public-key encryption (PKE) scheme is defined by PPT algorithms (Gen, Enc, Dec):

  $(pk, sk) \leftarrow Gen(1^n)$

  $C \leftarrow Enc_{pk}(m)$

  $m/\perp := Dec_{sk}(c)$

- **Correctness:** $\forall (pk, sk)$ output by $Gen(1^n)$, $\forall m \ (\in M_{pk})$.

  $$Dec_{sk}(Enc_{pk}(m)) = m.$$

- **Security:** Semantic / CPA / CCA?

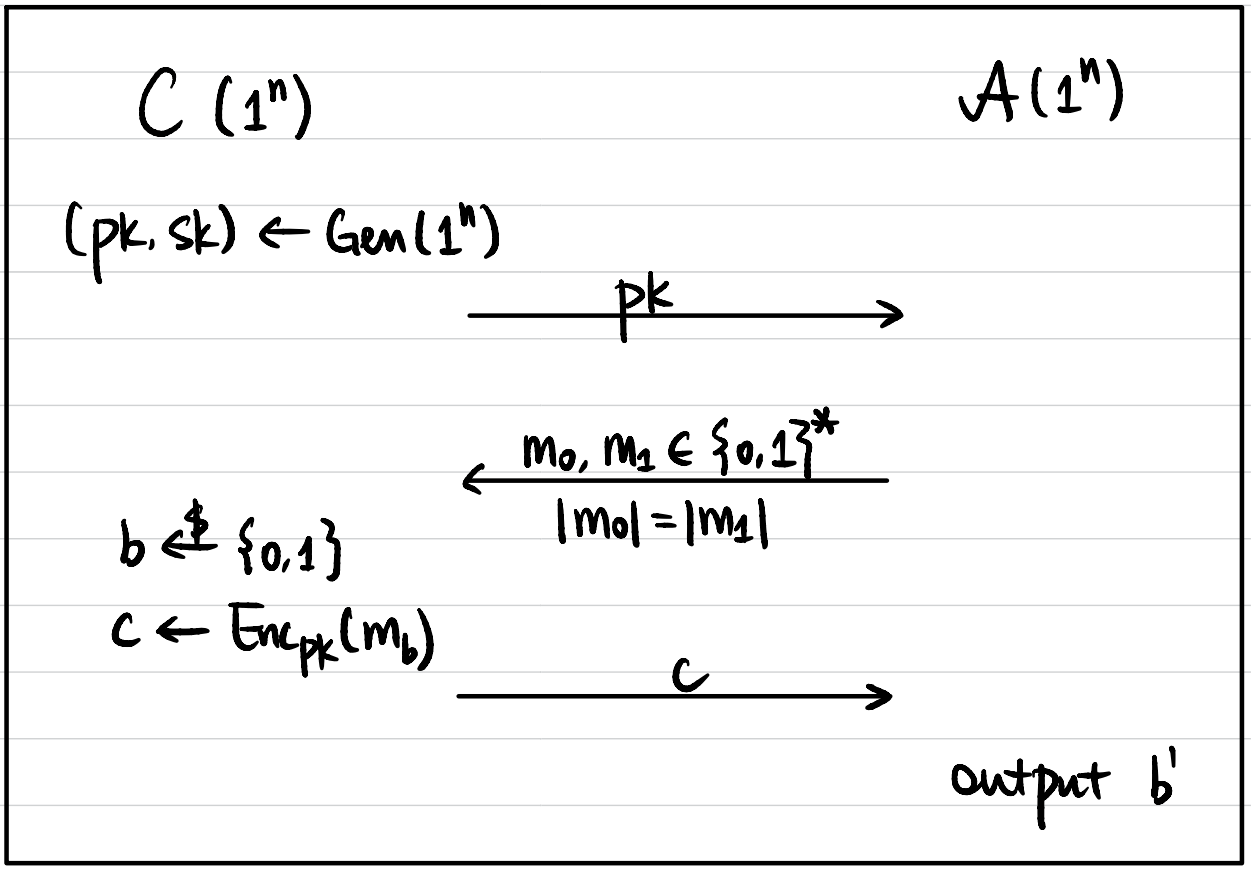
Semantic Security

Def A public-key encryption scheme (Gen, Enc, Dec) is **semantically secure** if $\forall$ PPT A, $\exists$ negligible function $\varepsilon(\cdot)$ s.t.

= CPA

$$\Pr[b = b'] \leq \frac{1}{2} + \varepsilon(n)$$

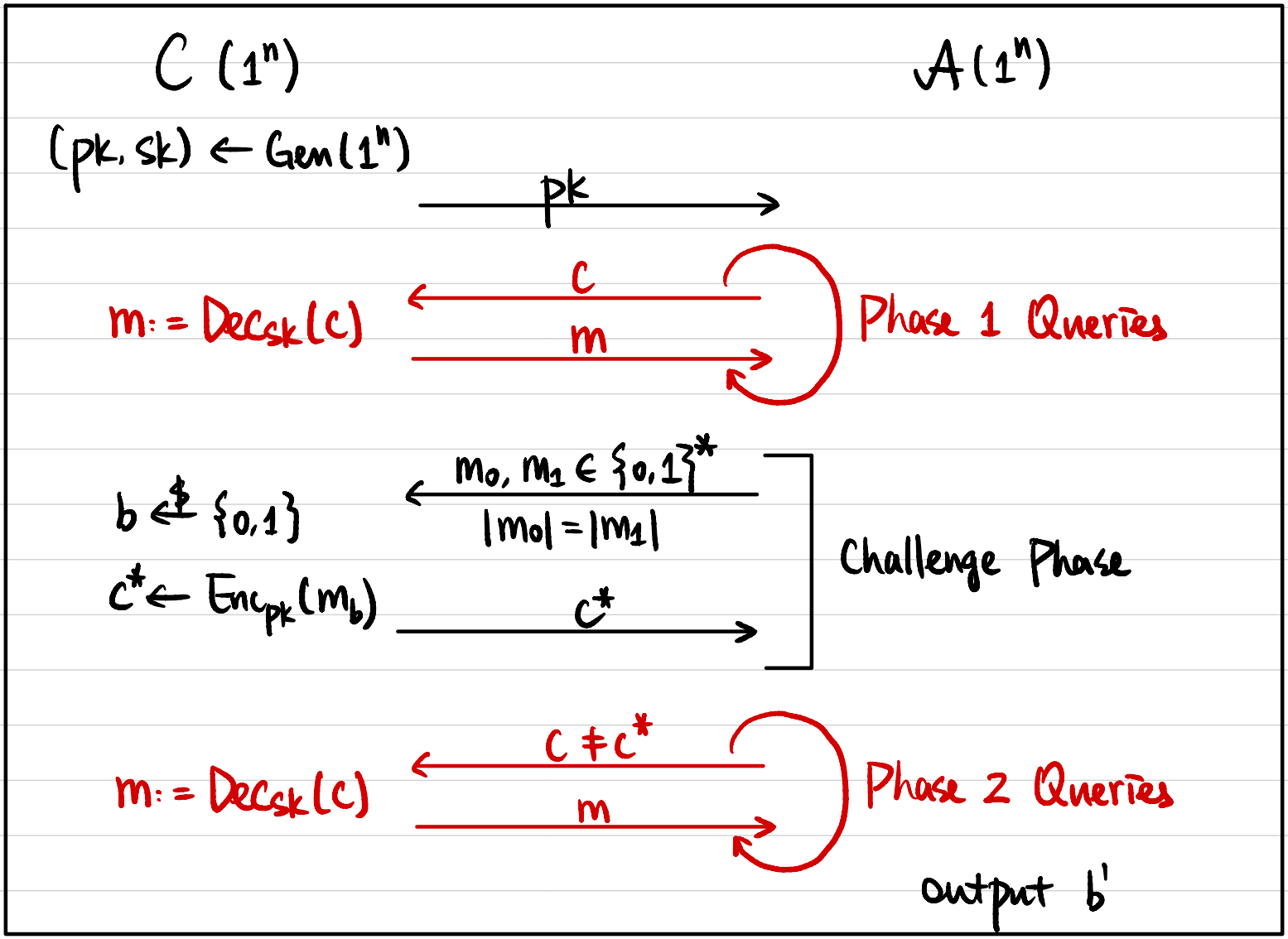
Requires randomized encryption



CPA Security?

Chosen Ciphertext Attack (CCA) Security

Def A public-key encryption scheme (Gen, Enc, Dec) is **CCA-secure** if $\forall$ PPT $\mathcal{A}$, $\exists$ negligible function $\varepsilon(\cdot)$ s.t. $\Pr[b = b'] \leq \frac{1}{2} + \varepsilon(n)$



El Gamal Encryption

- $Gen(1^n)$:

  $(G, q, g) \leftarrow G(1^n)$

  $x \xleftarrow{\$} \mathbb{Z}_q$, $h := g^x$

  $pk := (G, q, g, h)$

  $sk := x$

- $Enc_{pk}(m)$: $m \in G$

  $r \xleftarrow{\$} \mathbb{Z}_q$

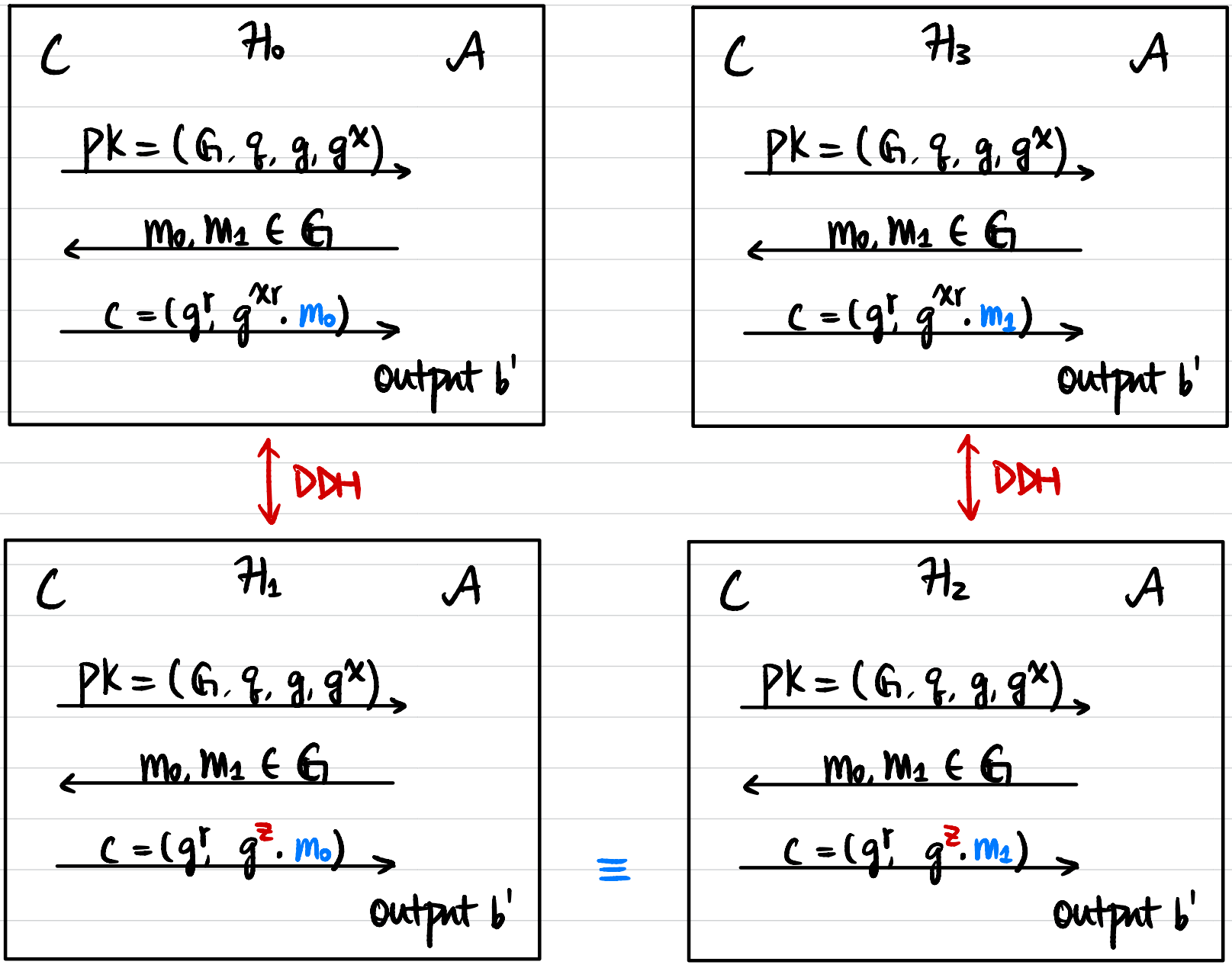
  $c := (g^r, h^r \cdot m)$

- $Dec_{sk}(c)$: $c = (c_1, c_2)$

  $m := \frac{c_2}{c_1^x}$

Thm If DDH is hard relative to G, then El Gamal encryption is CPA-secure.

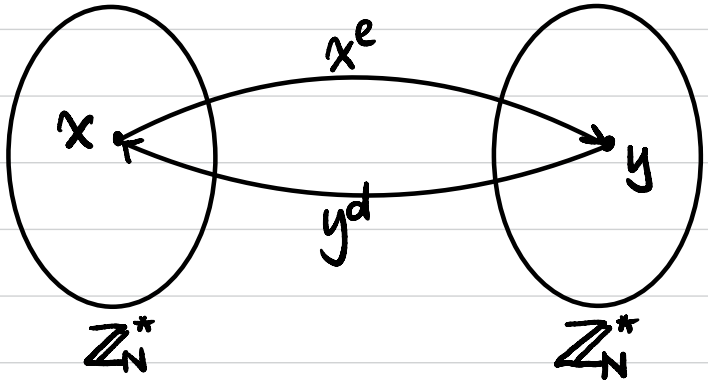
Proof Sketch



RSA-based Encryption

Plain RSA Encryption:

- $\text{Gen}(1^n)$:
  $(N, e, d) \leftarrow \text{GenRSA}(1^n)$
  $pk := (N, e)$
  $sk := (N, d)$



- $\text{Enc}_{pk}(m)$: $m \in \mathbb{Z}_N^*$
  $c := m^e \mod N$

- $\text{Dec}_{sk}(c)$: $m := c^d \mod N$

Is it CPA-secure? No!

RSA-based Encryption

Padded RSA Encryption:

- $\text{Gen}(1^n)$:

  $(N, e, d) \leftarrow \text{GenRSA}(1^n)$

  $pk := (N, e)$

  $sk := (N, d)$

- $\text{Enc}_{pk}(m)$: $m \in \{0,1\}$

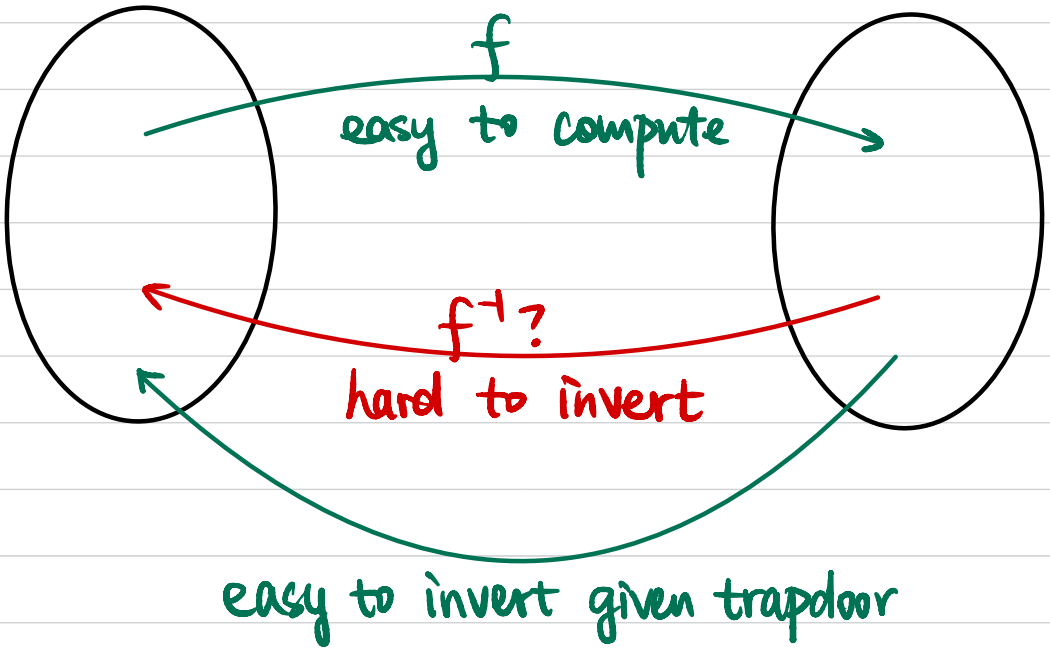
  $\hat{m} \xleftarrow{\$} \mathbb{Z}_N^*$ s.t. $\text{lsb}(\hat{m}) = m$ (lsb: least significant bit)

  $c := \hat{m}^e \bmod N$

- $\text{Dec}_{sk}(c)$: $m := \text{lsb}(c^d \bmod N)$

Thm If the RSA problem is hard relative to GenRSA, then this encryption scheme is CPA-secure.

Trapdoor Permutation



Trapdoor Permutation

Def A family $F = \{f_i : D_i \to R_i\}_{i \in I}$ is a **trapdoor permutation** if

① permutation: $\forall i \in I$, f_i is a permutation (bijection)

② easy to sample a function: $(i, t) \leftarrow Gen(1^n)$.

③ easy to sample an input: $x \leftarrow Sample(i \in I)$. x uniform in D_i.

④ easy to compute f_i: $f_i(x)$ poly-time computable $\forall i \in I, x \in D_i$.

⑤ hard to invert f_i: $\forall PPT\ A$, $\exists$ negligible function $\varepsilon(\cdot)$ s.t.

$$Pr\left[\begin{array}{l} (i,t) \leftarrow Gen(1^n), \\ x \leftarrow Sample(i) \\ y \leftarrow f_i(x) \\ z \leftarrow A(1^n, i, y) \end{array} : f_i(z) = y\right] \leq \varepsilon(n).$$

⑥ easy to invert f_i with trapdoor: $Inv(i, t, f_i(x)) = x$ $\quad (i,t) \leftarrow Gen(1^n)$, $x \in D_i$

Example: RSA trapdoor permutation