

# 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

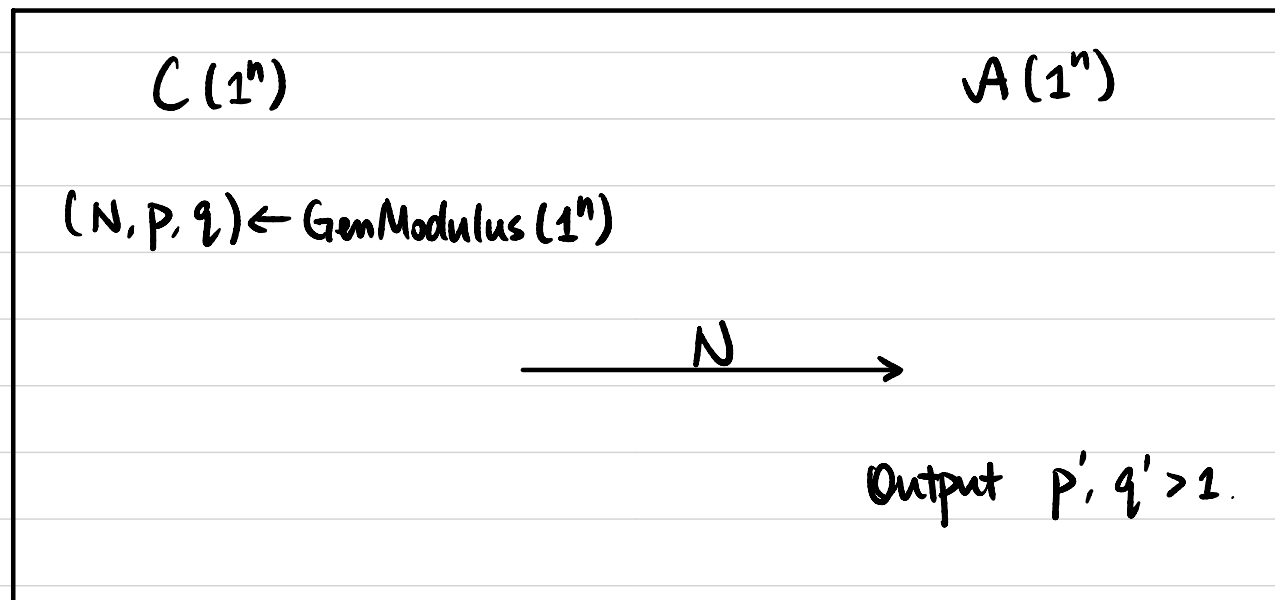
randomly sample  $\rightarrow$  primality test

GenModulus ( $1^n$ ): PPT algorithm, generates  $(N, p, q) \leftarrow$  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  $\epsilon(\cdot)$  s.t.  $\Pr[p' \cdot q' = N] \leq \epsilon(n)$ .



Factoring  $\Rightarrow$  OWF (GenModulus)

## RSA Assumption

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

GenRSA ( $1^n$ ):

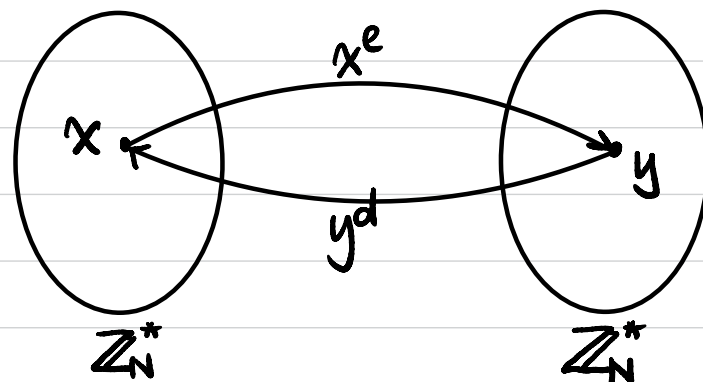
$(N, p, q) \leftarrow \text{GenModulus}(1^n)$

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

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

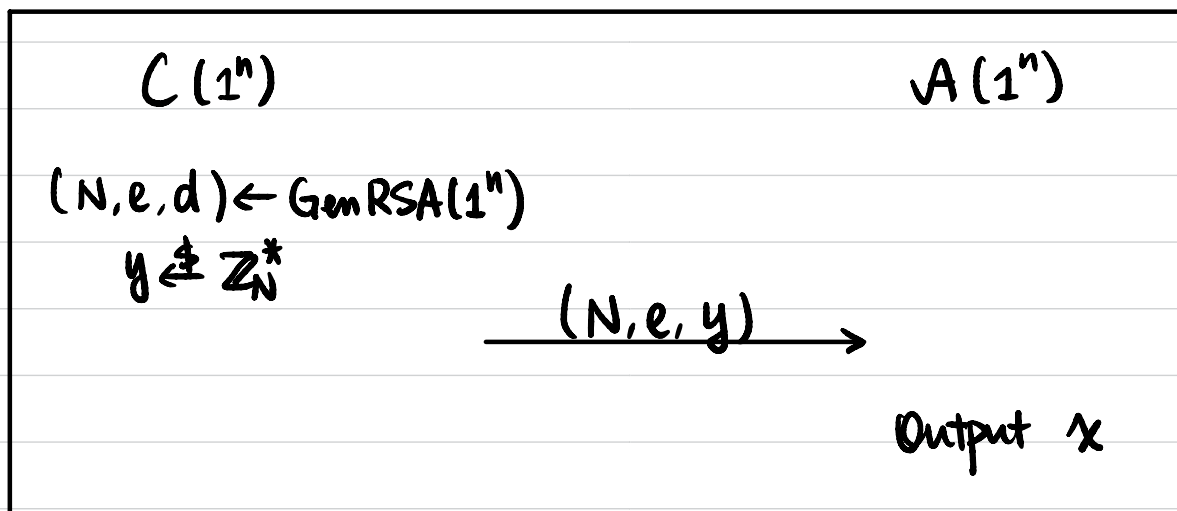
Compute  $d := e^{-1} \pmod{\phi(N)}$

Output  $(N, e, d)$



Def The RSA problem is hard relative to GenRSA if

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



RSA  $\Rightarrow$  Factoring

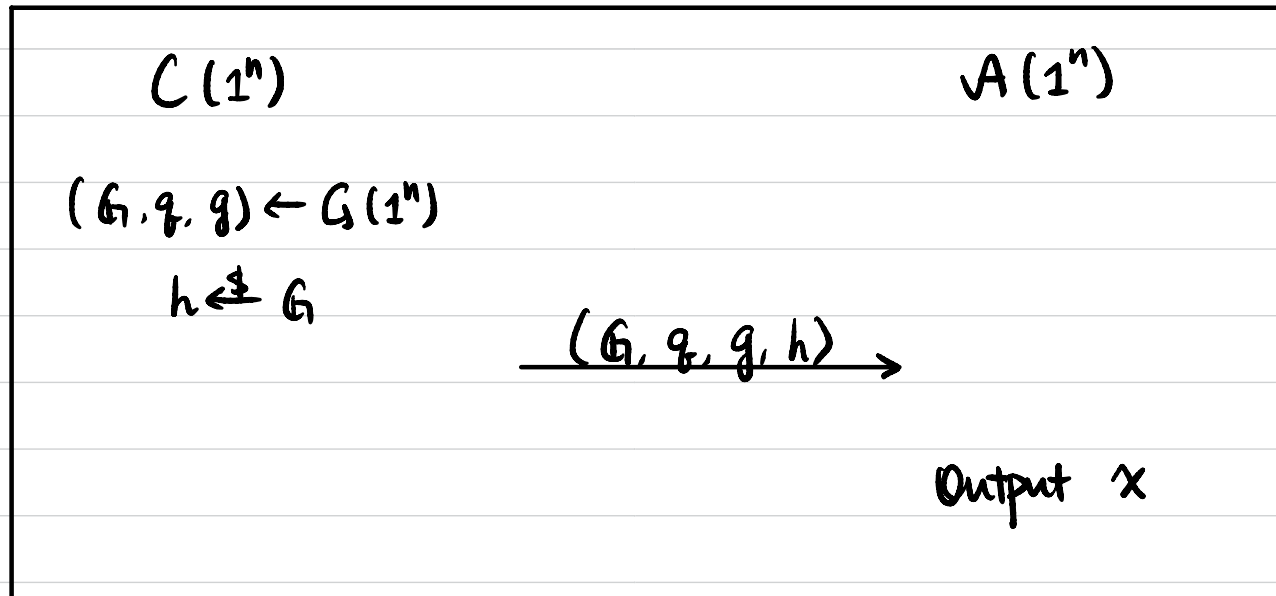
# Discrete-Log Assumption

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

description of a cyclic group  $G$  of order  $q$  with generator  $g$ .  
↑  
n-bit integer

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

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



DLOG  $\Rightarrow$  CRHF

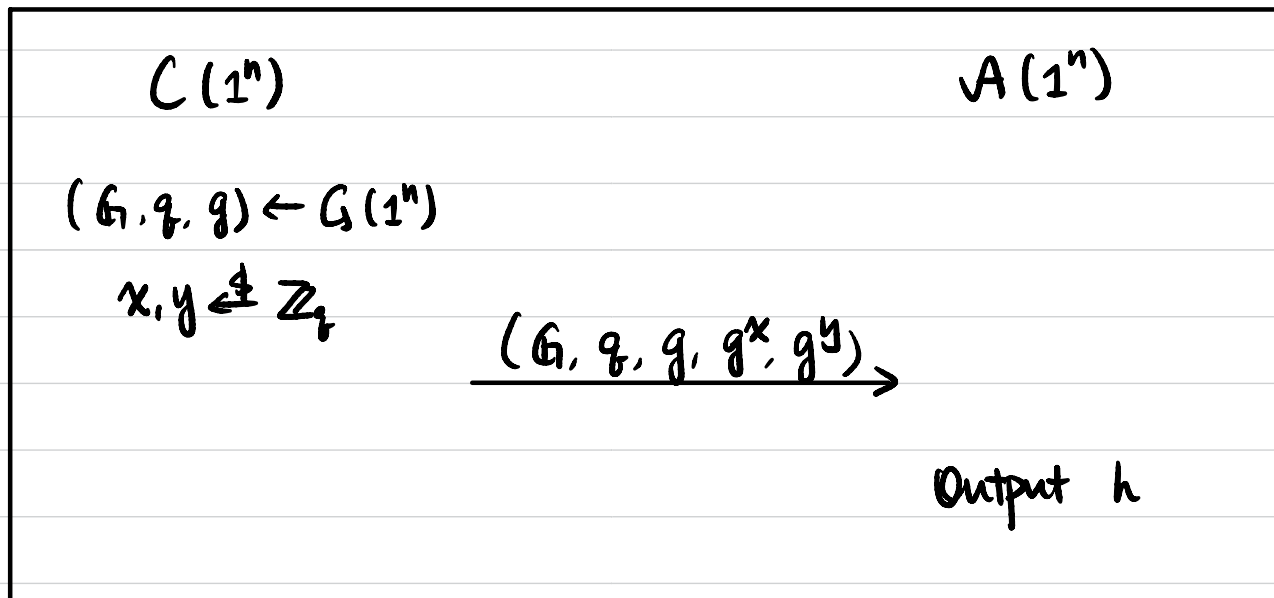


# Computational Diffie-Hellman (CDH) Assumption

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

Def CDH is hard relative to  $G$  if

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



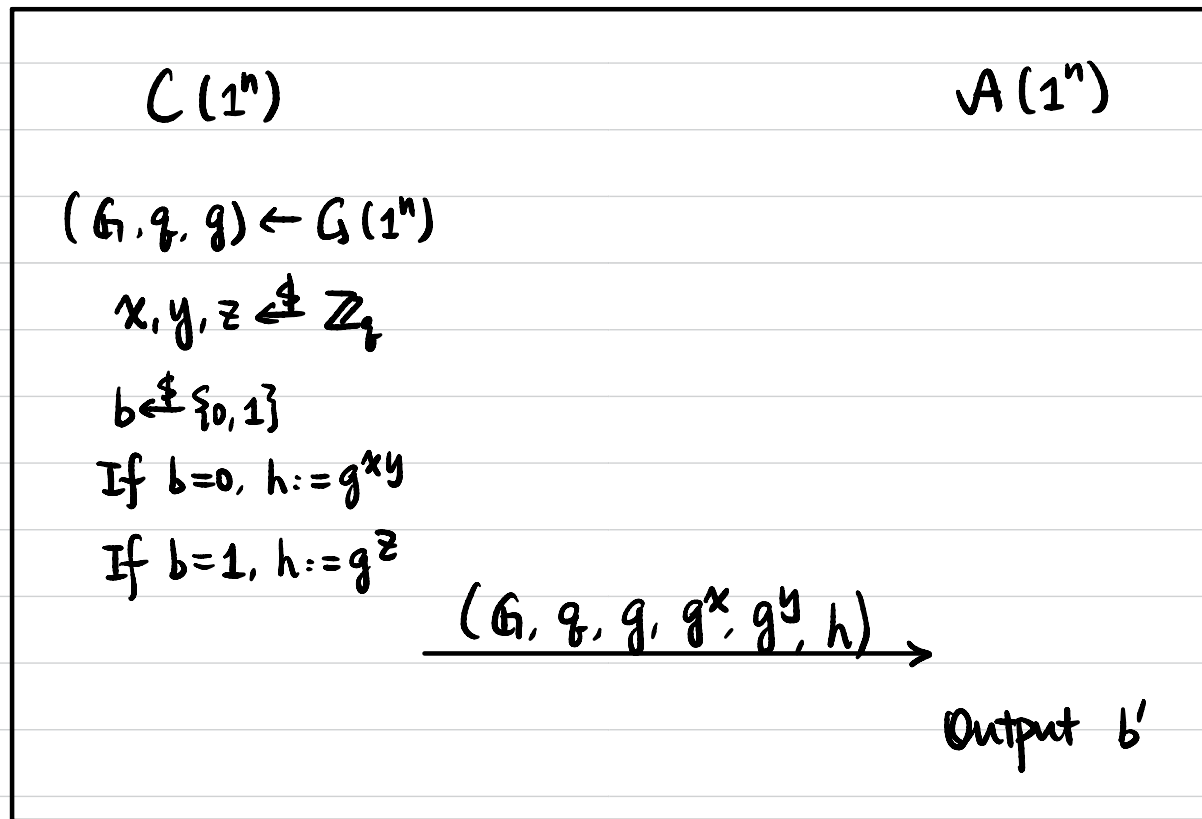
CDH  $\Rightarrow$  DLOG

# Decisional Diffie-Hellman (DDH) Assumption

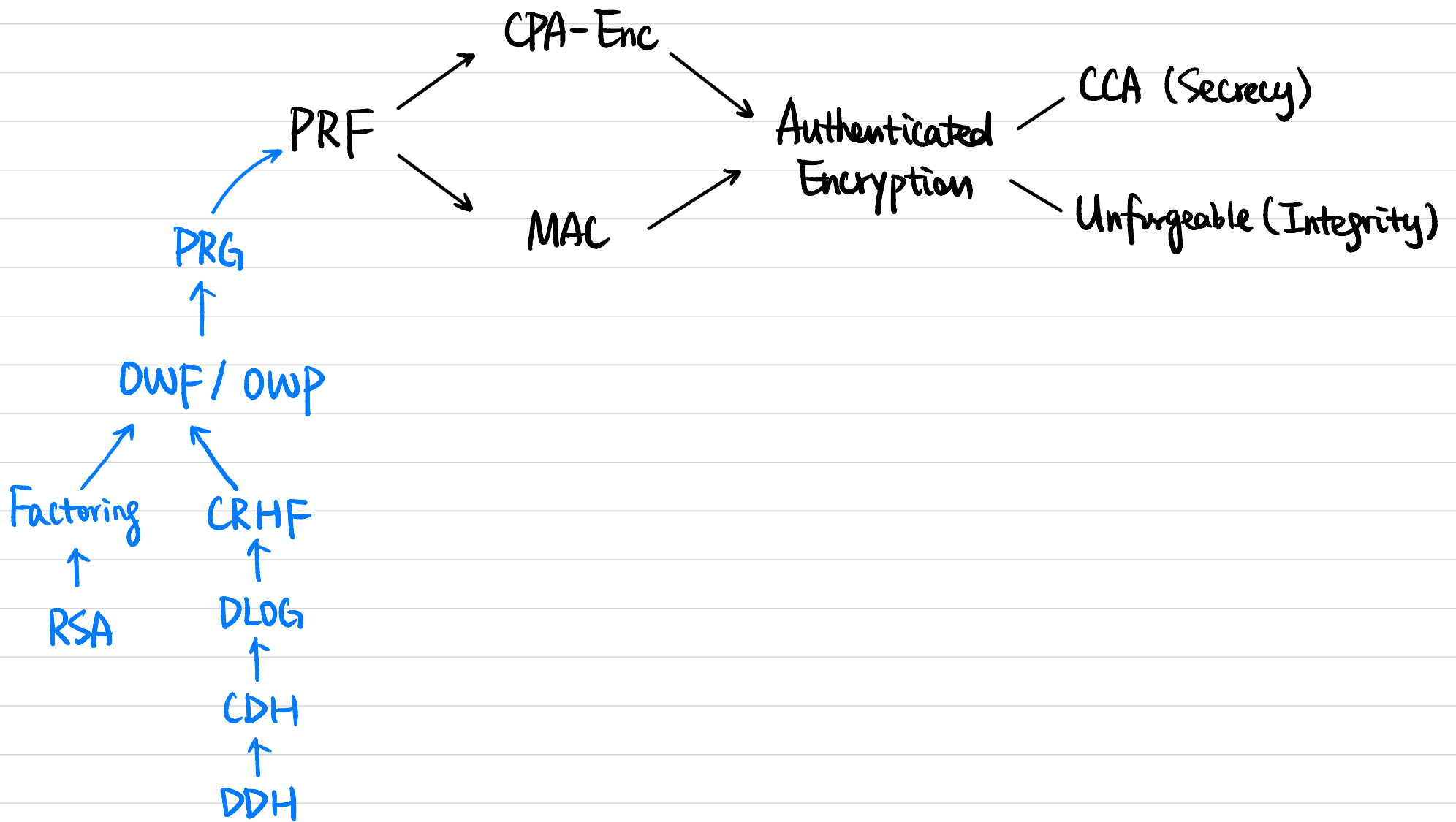
$G(1^n)$ : PPT algorithm, generates  $(G, q, g)$

Def DDH is hard relative to  $G$  if

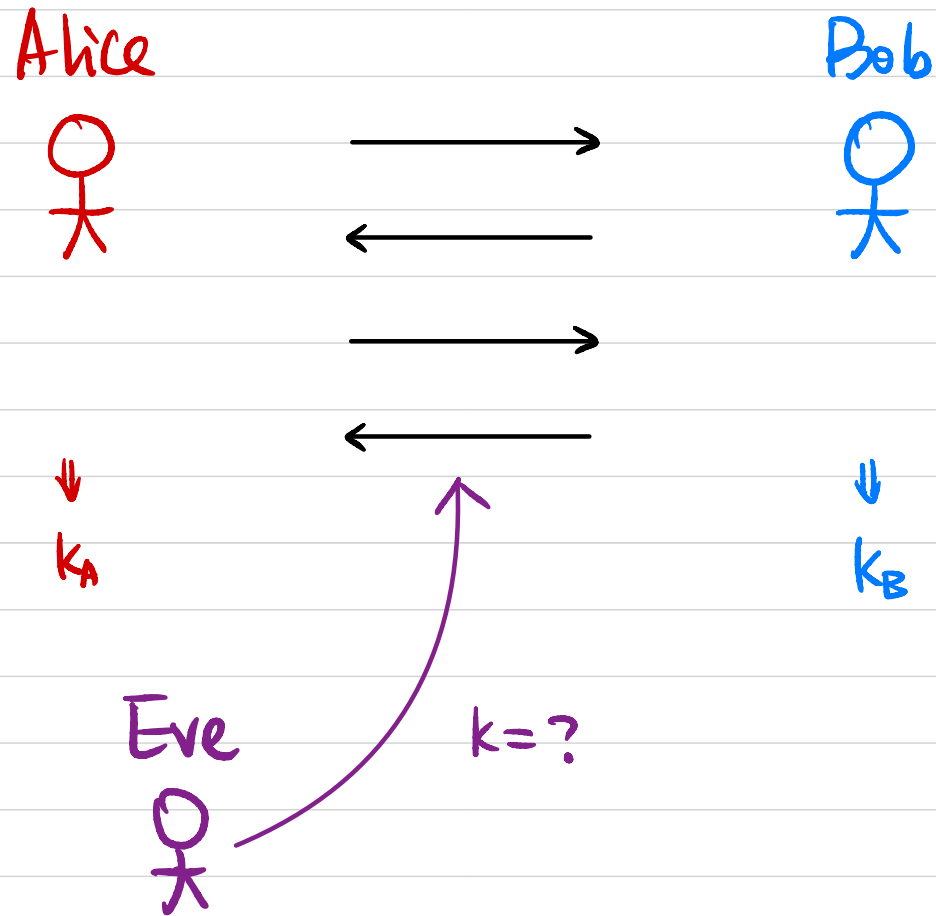
$\forall$  PPT  $A$ ,  $\exists$  negligible function  $\epsilon(\cdot)$  s.t.  $\Pr[b = b'] \leq \frac{1}{2} + \epsilon(n)$ .



DDH  $\Rightarrow$  CDH



# 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  $\Pi$  is secure if

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

$C(1^n)$

$A(1^n)$

Two parties holding  $1^n$  execute  $\Pi$ .

$\Rightarrow$  transcript  $T$  containing all the messages  
& a key  $k$  output by each party.

$b \leftarrow \{0, 1\}$

If  $b=0$ ,  $\hat{k} := k$

If  $b=1$ ,  $\hat{k} \leftarrow \{0, 1\}^n$

$(T, \hat{k}) \rightarrow$

output  $b'$

# Diffie-Hellman Key Exchange

Alice

Bob

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

$$x \leftarrow \mathbb{Z}_q$$

$$h_A := g^x$$

$$\xrightarrow{(G, q, g, h_A)}$$

$$y \leftarrow \mathbb{Z}_q$$

$$h_B := g^y$$

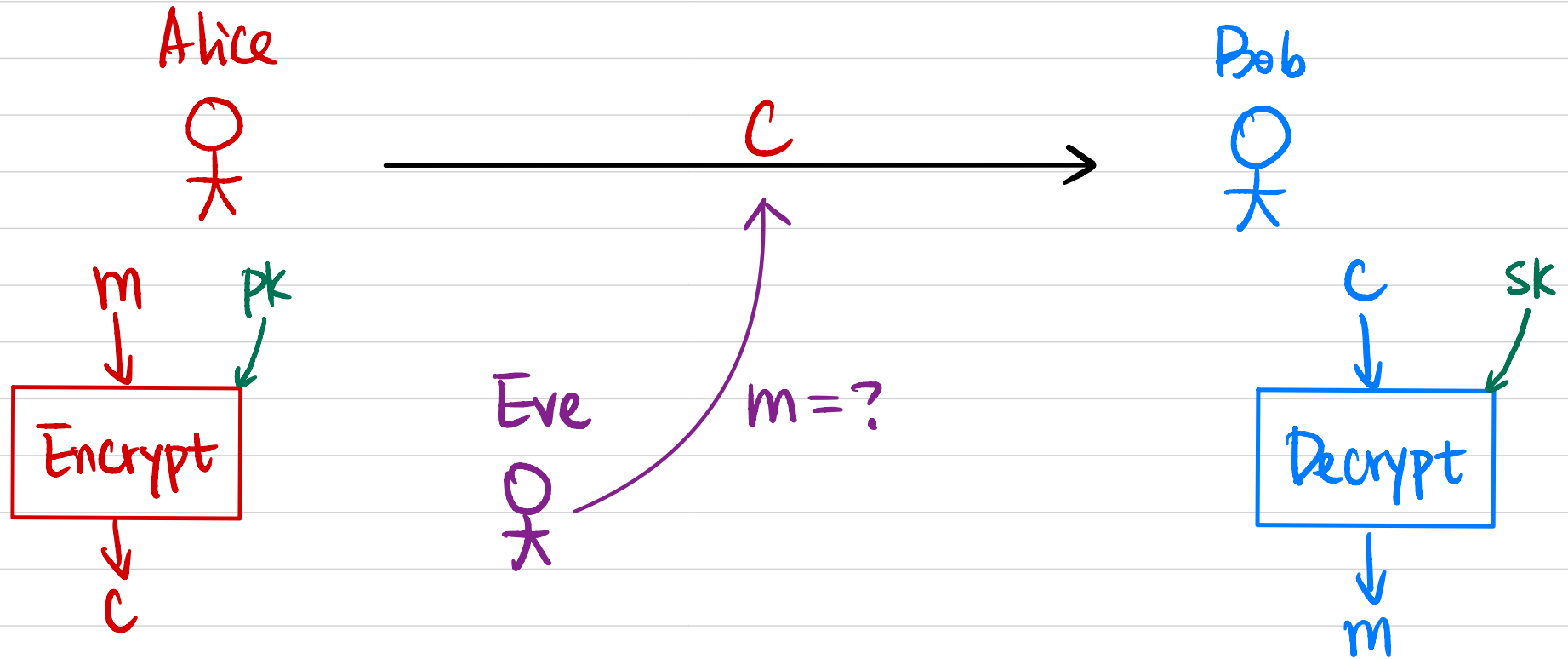
$$\xleftarrow{h_B}$$

$$\Downarrow$$
$$k_A := h_B^x$$

$$\Downarrow$$
$$k_B := h_A^y$$

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  $Gen(1^n)$ ,  $\forall m (\in M_{pk})$ ,

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

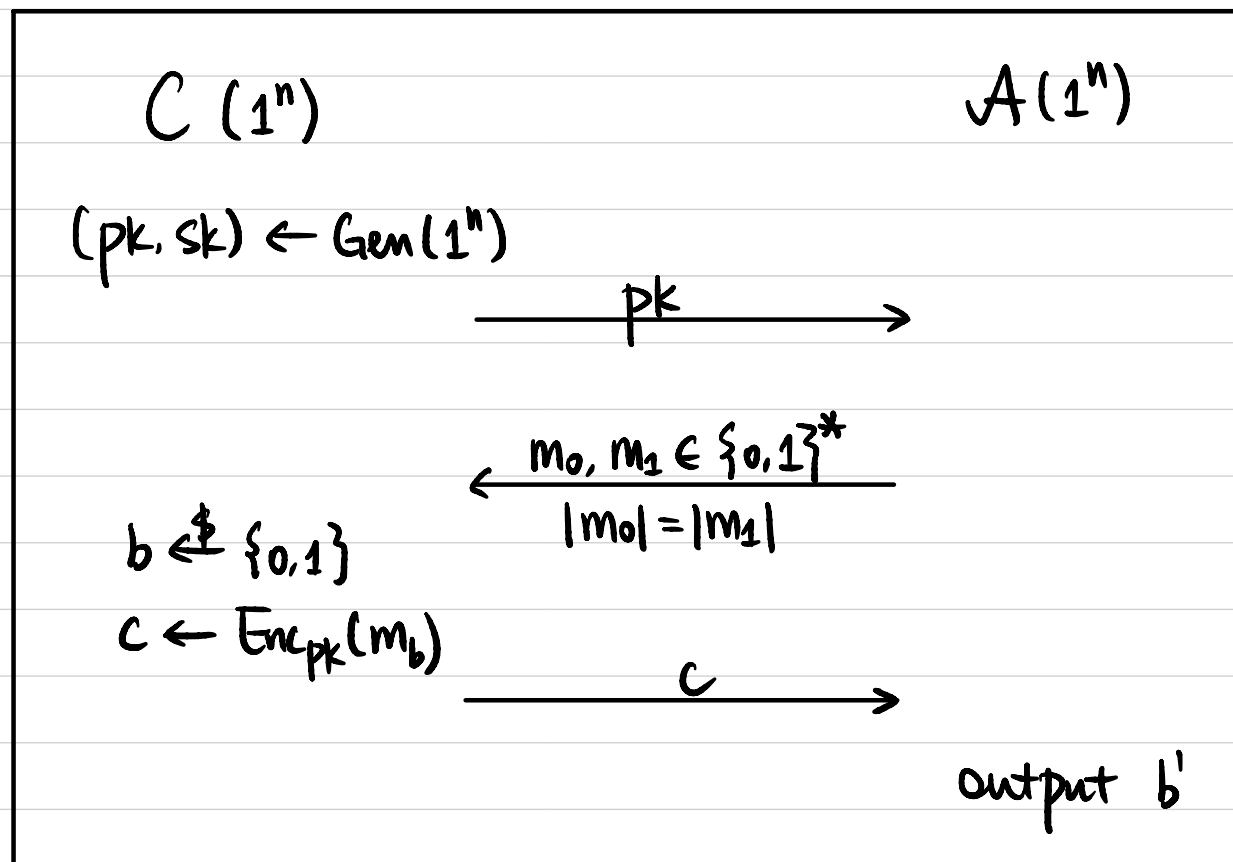
- **Security:** Semantic / CPA / CCA?



# Semantic Security

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

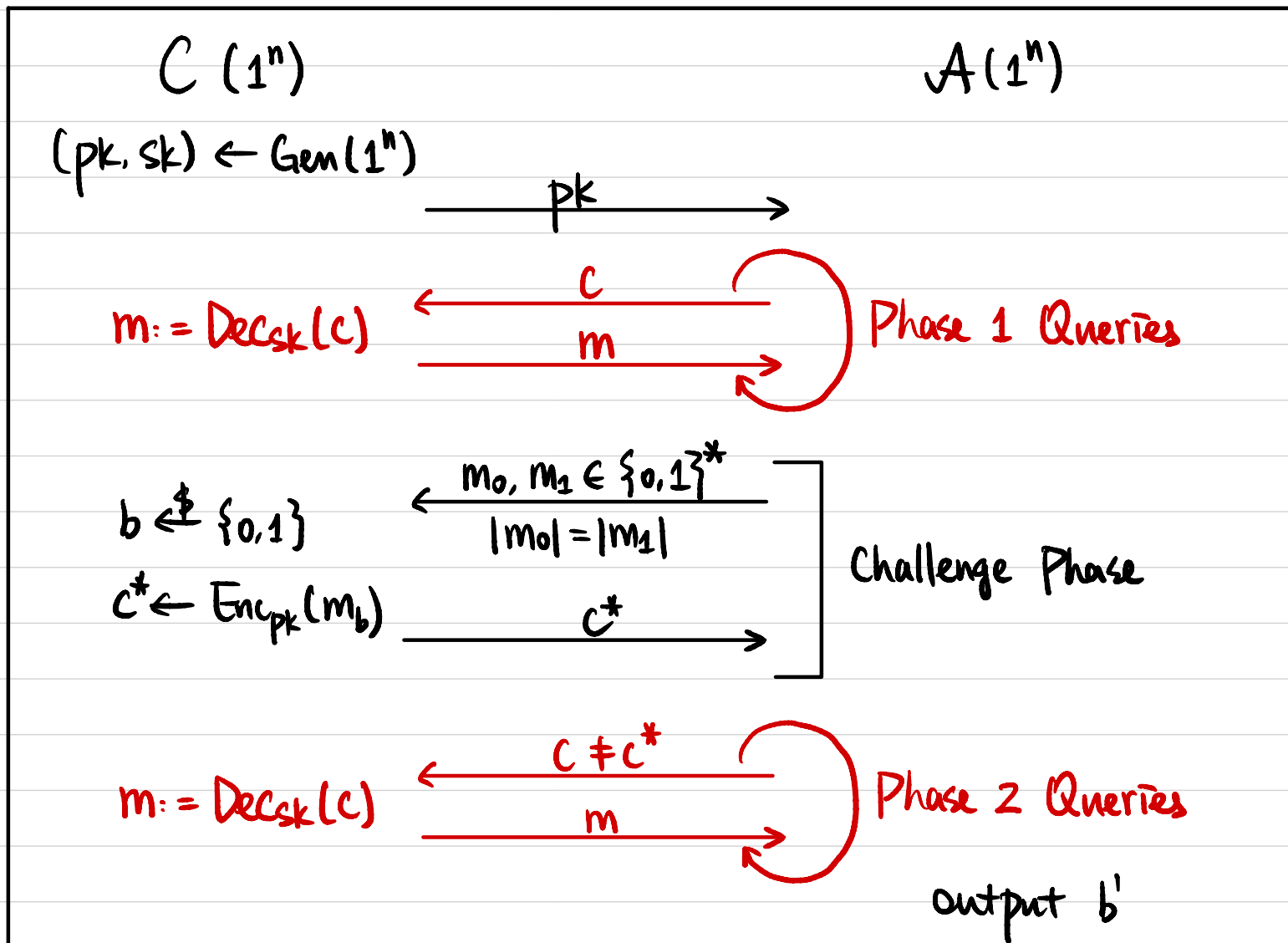
$$\Pr[b = b'] \leq \frac{1}{2} + \epsilon(n)$$



**CPA Security?**

# Chosen Ciphertext Attack (CCA) Security

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



# El Gamal Encryption

•  $\text{Gen}(1^n)$ :

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

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

$$\text{PK} := (G, q, g, h)$$

$$\text{SK} := x$$

•  $\text{Enc}_{\text{PK}}(m)$ :  $m \in G$

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

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

•  $\text{Dec}_{\text{SK}}(c)$ :  $c = (c_1, c_2)$

?

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

## RSA-based Encryption

### Plain RSA Encryption:

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

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

$$\text{pk} := (N, e)$$

$$\text{sk} := (N, d)$$

- $\text{Enc}_{\text{pk}}(m)$ :  $m \in \mathbb{Z}_N^*$

$$c := m^e \bmod N$$

- $\text{Dec}_{\text{sk}}(c)$ : ?

Is it CPA-secure?

# RSA-based Encryption

## Padding RSA Encryption:

•  $\text{Gen}(1^n)$ :

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

$$\text{pk} := (N, e)$$

$$\text{sk} := (N, d)$$

•  $\text{Enc}_{\text{pk}}(m)$ :  $m \in \{0, 1\}$  *least significant bit*

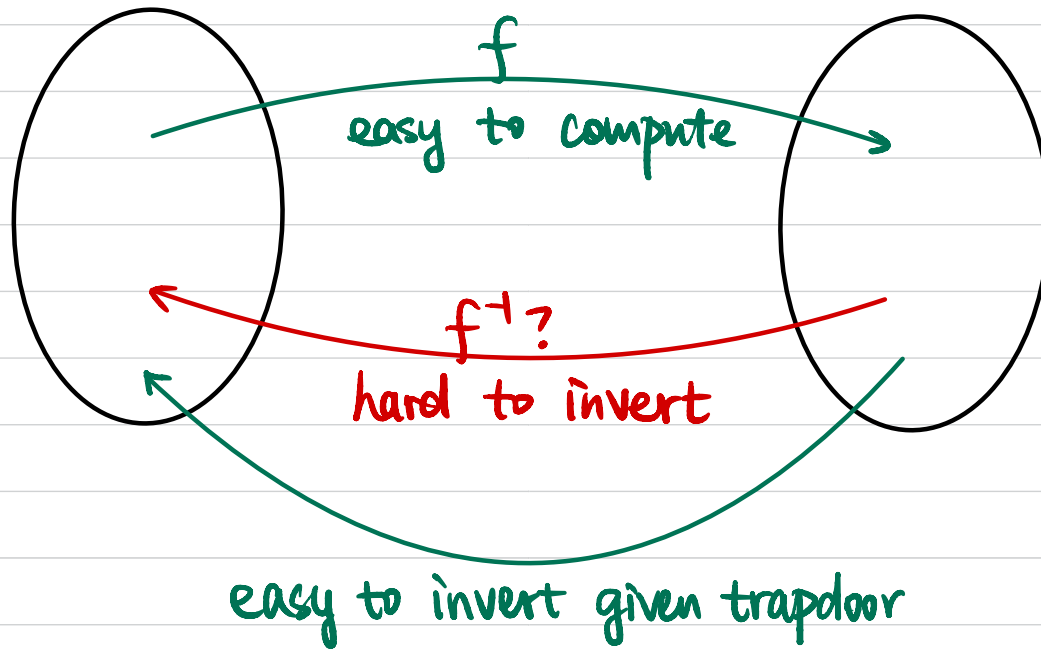
$$\hat{m} \leftarrow \mathbb{Z}_N^* \text{ s.t. } \text{lsb}(\hat{m}) = m$$

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

•  $\text{Dec}_{\text{sk}}(c)$ : ?

Thm If the RSA problem is hard relative to  $\text{GenRSA}$ , then this encryption scheme is CPA-secure.

# Trapdoor Permutation



## Trapdoor Permutation

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

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

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

③ easy to sample an input:  $x \leftarrow \text{Sample}(i \in I)$ .  $x$  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  $f_i$ :  $\forall \text{PPT } A, \exists$  negligible function  $\epsilon(\cdot)$  s.t.

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

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

Example: RSA trapdoor permutation

## Hard-Core Predicate

**Def** Let  $\Pi = (F, \text{Gen}, \text{Inv})$  be a trapdoor permutation,  
Let  $hc$  be a deterministic poly-time algorithm that, on input  $i$  &  $x \in D_i$ ,  
Outputs a single bit  $hc_i(x)$ .

$hc$  is a hard-core predicate of  $\Pi$  if

$\forall$  PPT  $A$ ,  $\exists$  negligible function  $\epsilon(\cdot)$  s.t.

$$\Pr_{\substack{(i,t) \leftarrow \text{Gen}(1^n) \\ x \leftarrow D_i}} [A(i, f_i(x)) = hc_i(x)] \leq \frac{1}{2} + \epsilon(n)$$

**Thm** Assume trapdoor permutation exists.

Then there exists a trapdoor permutation  $\Pi$  with a hard-core predicate  $hc$  of  $\Pi$ .



## PKE from TDP

•  $\text{Gen}(1^n)$ :

$$(i, t) \leftarrow \text{Gen}(1^n)$$

$$pk := i$$

$$sk := t$$

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

$$r \leftarrow D_i \text{ st. } hc_i(r) = m$$

$$c := f_i(r)$$

•  $\text{Dec}_{sk}(c)$ : ?

Thm If  $\pi = (F, \text{Gen}, \text{Inv})$  be a trapdoor permutation with a hard-core predicate  $hc$ , then this encryption scheme is CPA-secure.