CSCI 1510

- Trapdoor Permutations (continued)
- Post-Quantum PKE from LWE Assumption

ANNOUNCEMENT: Mid-semester survey (for extra credit)

Key Exchange: Security

Def A key exchange protocol Π is secure if

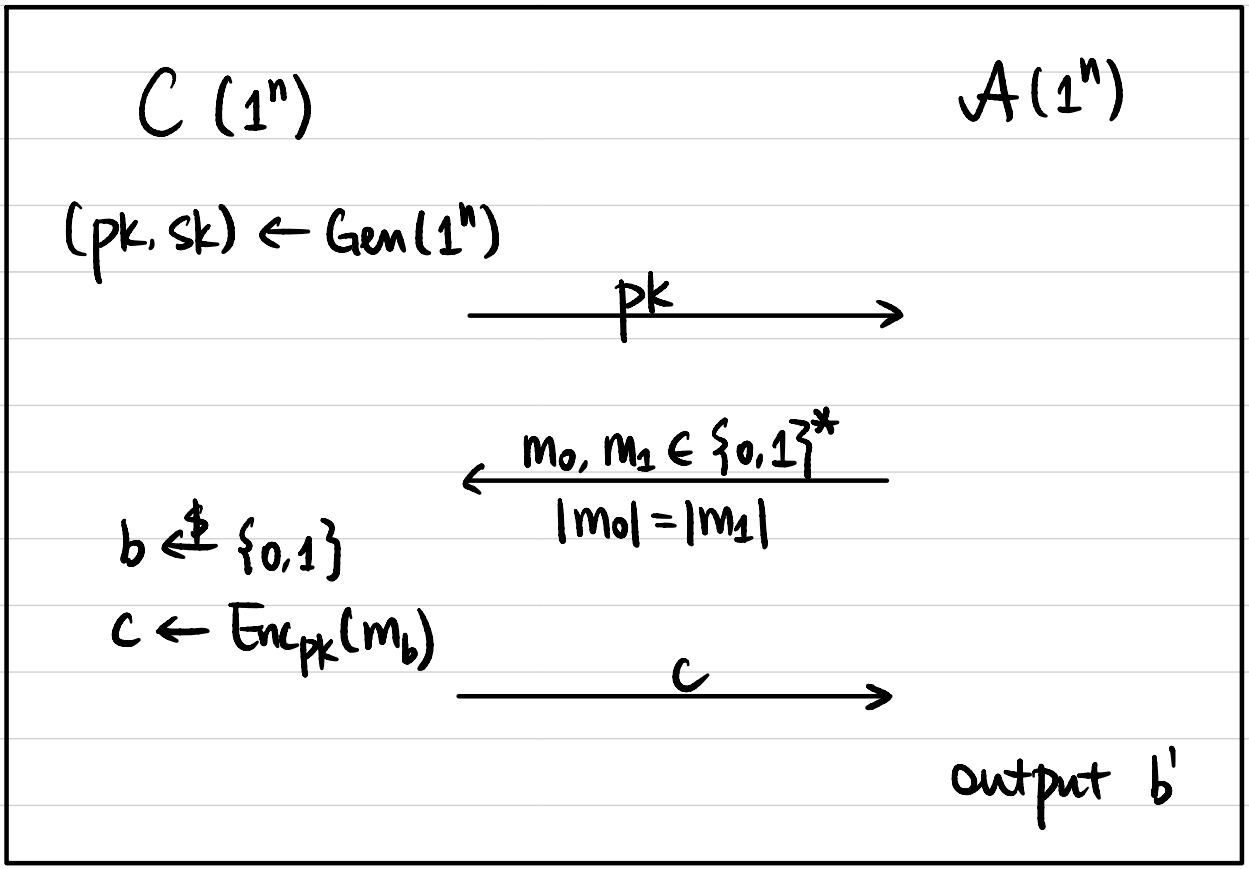
$\forall$ PPT A, $\exists$ negligible function $\varepsilon(\cdot)$ s.t. $\Pr[b = b'] \leq \frac{1}{2} + \varepsilon(n)$.

$C(1^n)$	$A(1^n)$
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'

CPA Security

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

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



CPA-secure PKE $\Rightarrow$ Key Exchange

?

PKE from 2-Message Key Exchange

- $Gen(1^n)$:

  Generate Msg_1 of P_1 in key exchange

  $pk := Msg_1$

  $sk :=$ secret state of P_1

- $Enc_{pk}(m)$:

  Generate Msg_2 of P_2 in key exchange

  Derive k in key exchange

  $c := (Msg_2, k \oplus m)$

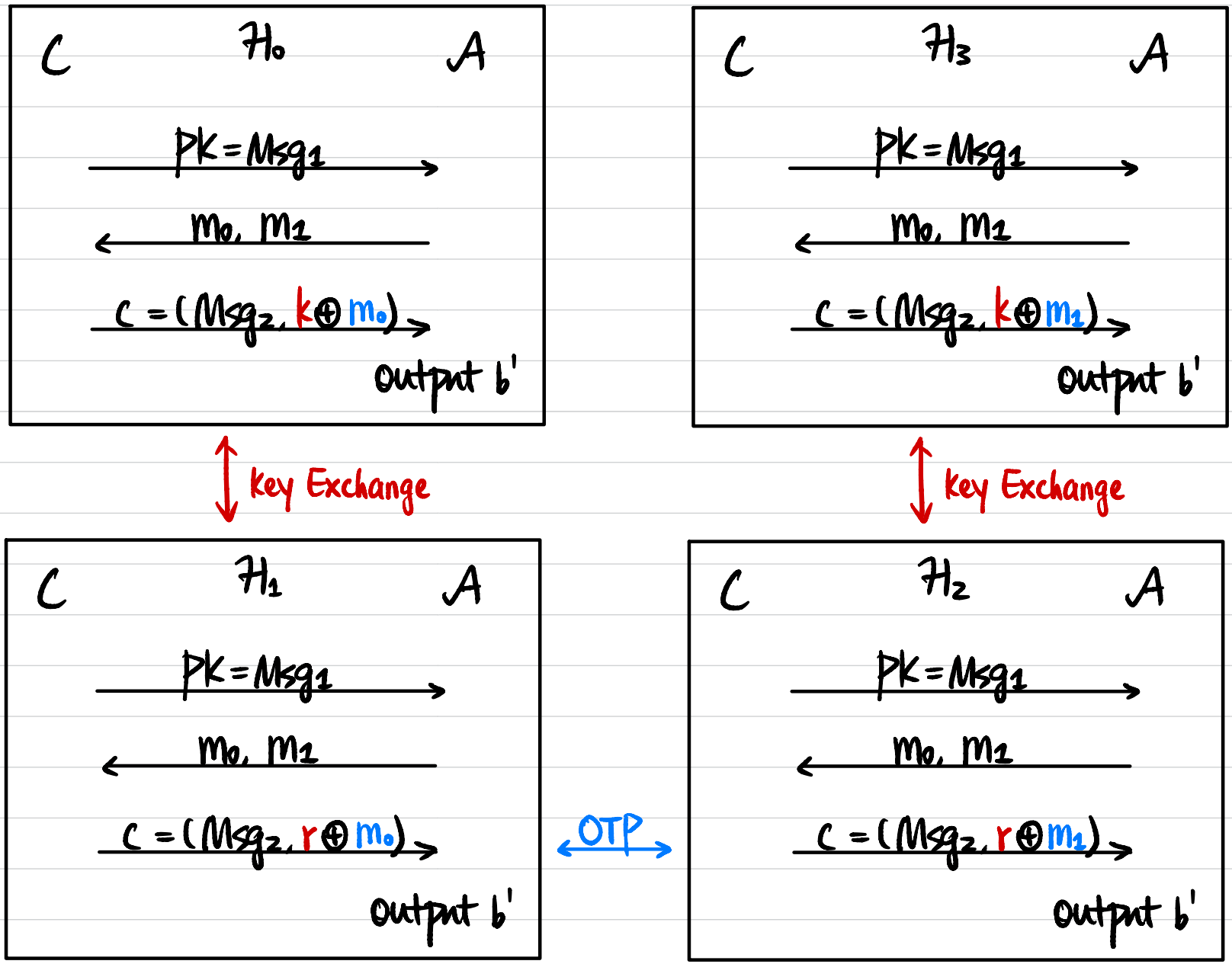
- $Dec_{sk}(c)$: $c = (Msg_2, c')$

  Derive k in key exchange

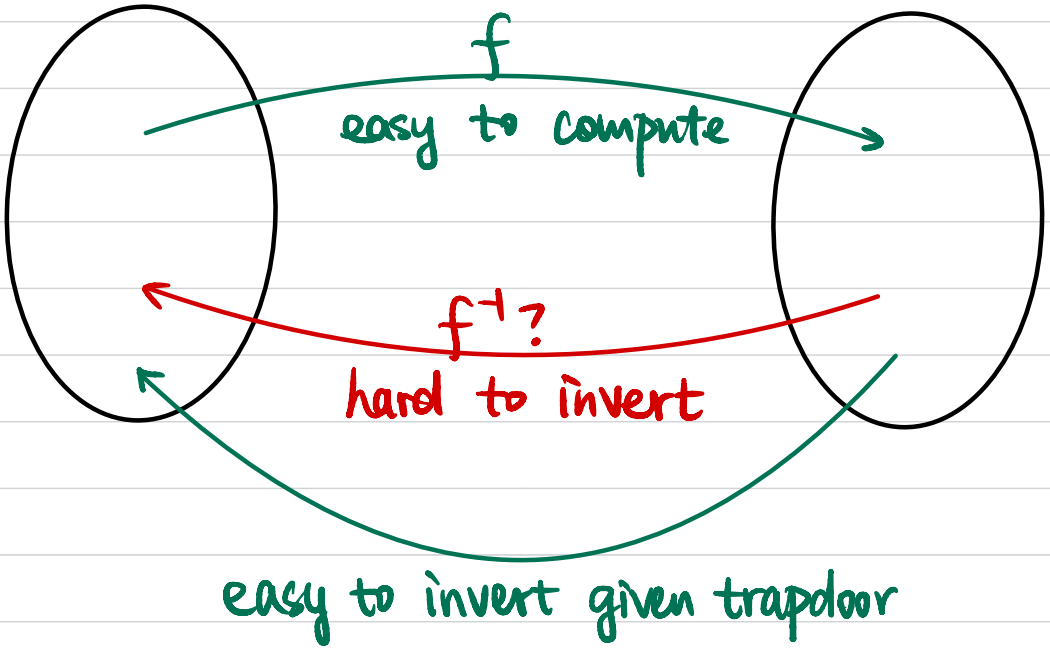
  Output $k \oplus c'$

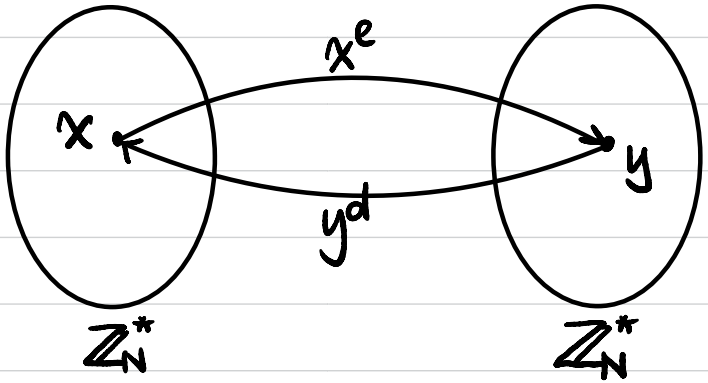
Thm If the key exchange protocol is secure, then this encryption scheme is CPA-secure.

Proof Sketch



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: $Inv(i, t, f_i(x)) = x$ $\quad (i,t) \leftarrow Gen(1^n)$, $x \in D_i$

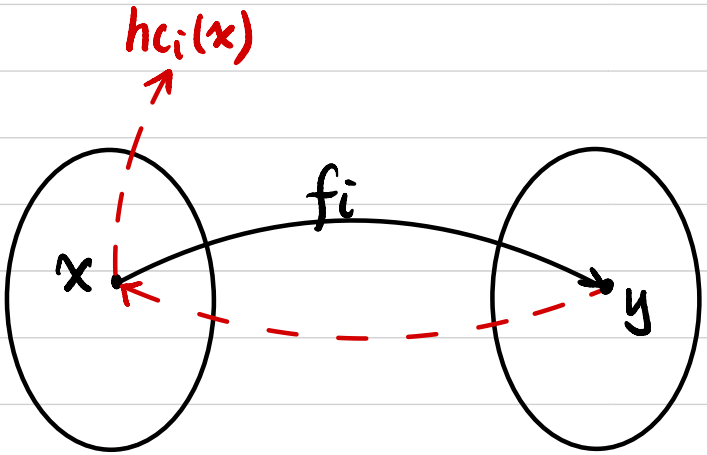
Example: RSA trapdoor permutation

Hard-Core Predicate

Def Let $\Pi = (F, Gen, Inv)$ 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 Π if
$\forall$ PPT A, $\exists$ negligible function $\varepsilon(\cdot)$ s.t.

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

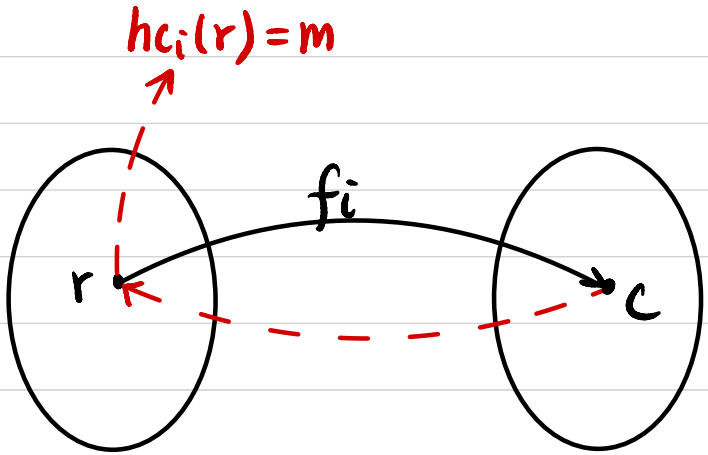


Thm Assume trapdoor permutation exists.
Then there exists a trapdoor permutation Π with a hard-core predicate hc of Π.

PKE from TDP

- $\text{Gen}(1^n)$:
  $(i, t) \leftarrow \text{Gen}(1^n)$
  $pk := i$
  $sk := t$

- $\text{Enc}_{pk}(m)$: $m \in \{0,1\}$
  $r \leftarrow D_i$ s.t. $hc_i(r) = m$
  $c := f_i(r)$

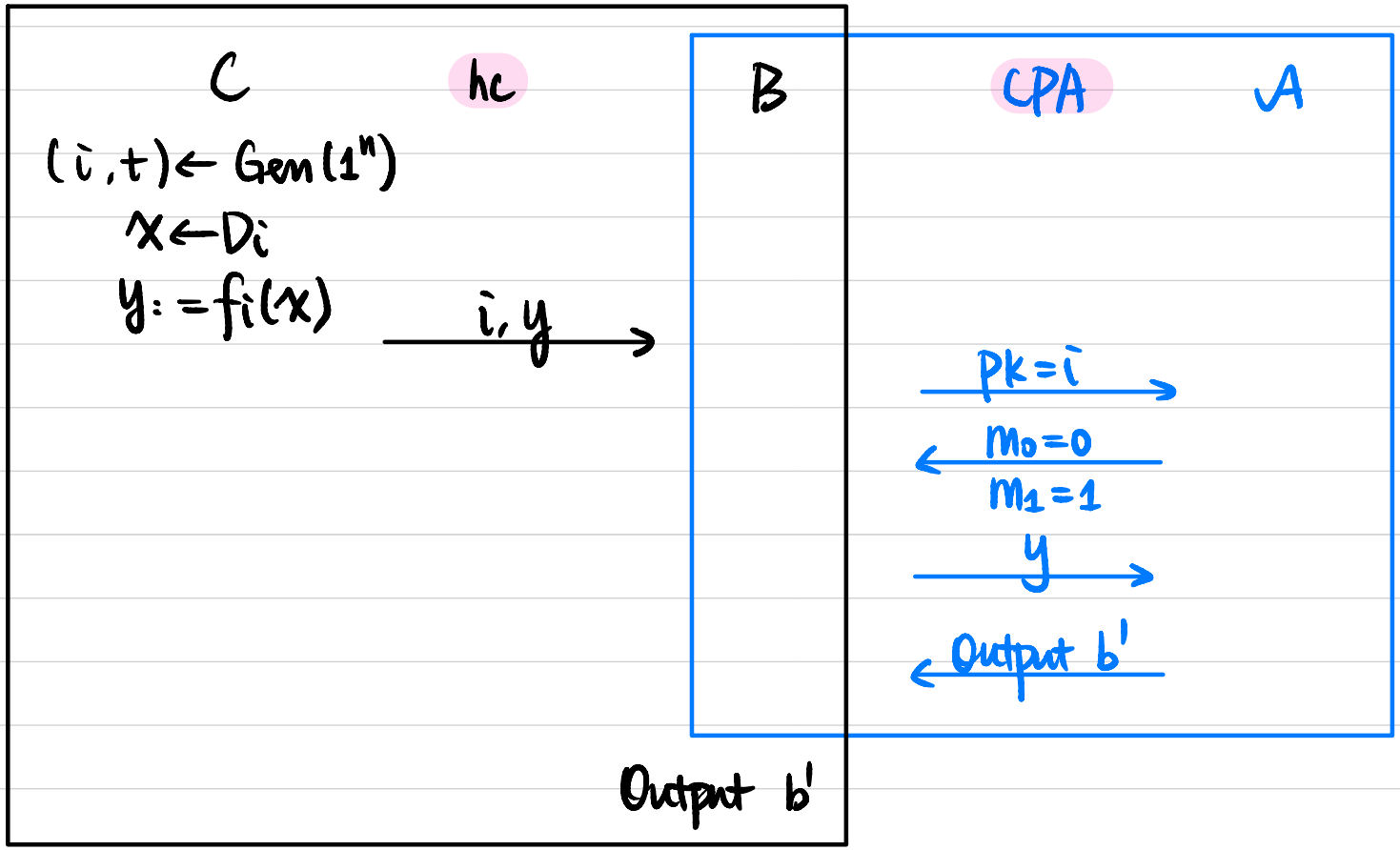


- $\text{Dec}_{sk}(c)$: $hc_i(\text{Inv}(i, t, c))$

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.

Proof Assume $\exists$ PPT $\mathcal{A}$ that breaks CPA security.

We construct PPT B to break the security of hc.



$P_0 + P_1 = 1$

$|P_0 - P_1| \le negl(n)$

$\Rightarrow P_0 \ge \frac{1}{2} - negl(n)$

$P_1 \ge \frac{1}{2} - negl(n)$

$\frac{1}{2} \cdot (q_0 + q_1)$

$= Pr[\mathcal{A} \text{ wins CPA game}]$

$\ge \frac{1}{2} + non\text{-}negl(n)$.

$$Pr[h_i(x) = b' \mid x \leftarrow D_i] = \underset{P_0}{Pr[h_i(x)=0 \mid x \leftarrow D_i]} \cdot \underset{q_0}{Pr[\mathcal{A} \text{ guesses correctly} \mid y \text{ encrypts } 0]}$$

$$+ \underset{P_1}{Pr[h_i(x)=1 \mid x \leftarrow D_i]} \cdot \underset{q_1}{Pr[\mathcal{A} \text{ guesses correctly} \mid y \text{ encrypts } 1]}$$

$$\ge (\tfrac{1}{2} - negl(n)) \cdot q_0 + (\tfrac{1}{2} - negl(n)) \cdot q_1$$

$$\ge \tfrac{1}{2} \cdot (q_0 + q_1) - negl(n) \ge \tfrac{1}{2} + non\text{-}negl(n) - negl(n)$$

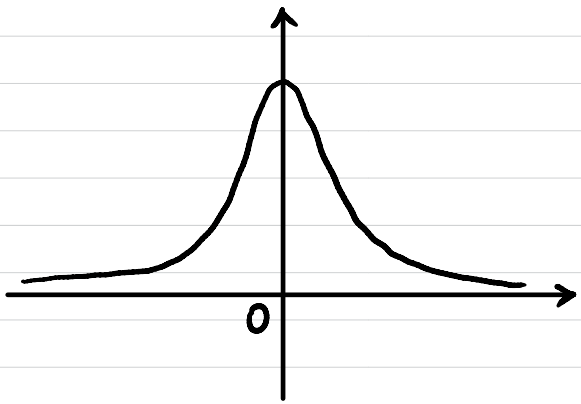
Post-Quantum Assumption: Learning With Errors (LWE)

n: security parameter

$q \sim 2^{n^\epsilon}$

$m = \Omega(n \log q)$

χ: distribution over $\mathbb{Z}_q$
(concentrated on "small integers")

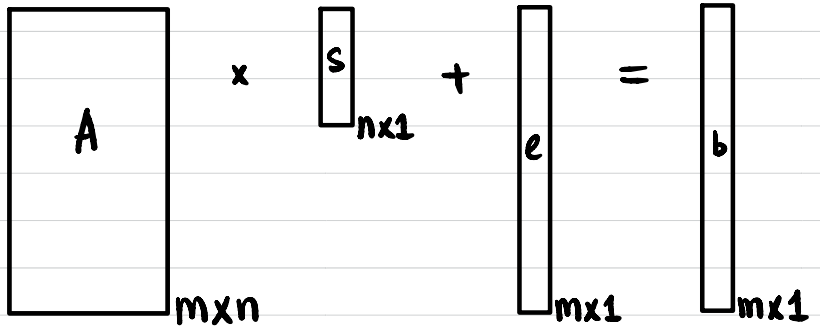


$$\Pr[|e| > \alpha \cdot q \mid e \leftarrow \chi] \leq \text{negl}(n)$$

$\alpha \ll 1$

Def We say the decisional $\text{LWE}_{n,m,q,\chi}$ problem is (quantum) hard if $\forall$ (quantum) PPT $\mathcal{A}$, $\exists$ negligible function $\epsilon(\cdot)$ s.t.

$$\left| \Pr\left[\begin{matrix} A \xleftarrow{\$} \mathbb{Z}_q^{m \times n} \\ s \xleftarrow{\$} \mathbb{Z}_q^n \\ e \leftarrow \chi^m \end{matrix} : \mathcal{A}(A, [As + e \bmod q]) = 1 \right] - \Pr\left[\begin{matrix} A \xleftarrow{\$} \mathbb{Z}_q^{m \times n} \\ b' \xleftarrow{\$} \mathbb{Z}_q^m \end{matrix} : \mathcal{A}(A, b') = 1 \right] \right| \leq \epsilon(n).$$



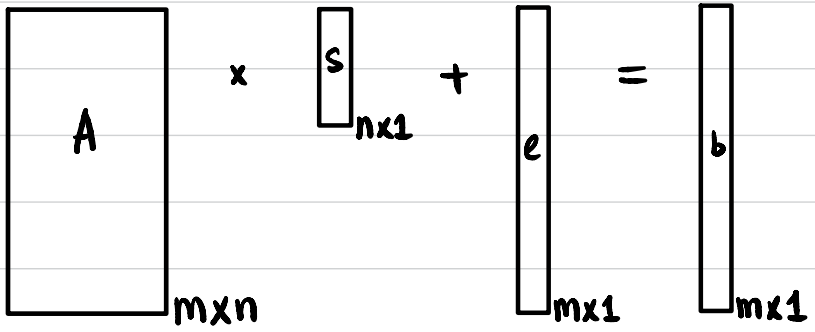
Post-Quantum PKE: Regev Encryption

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

  $A \xleftarrow{\$} \mathbb{Z}_q^{m \times n} \quad s \xleftarrow{\$} \mathbb{Z}_q^n \quad e \leftarrow \chi^m$

  $pk = (A, b = As + e \bmod q)$

  $sk = s$

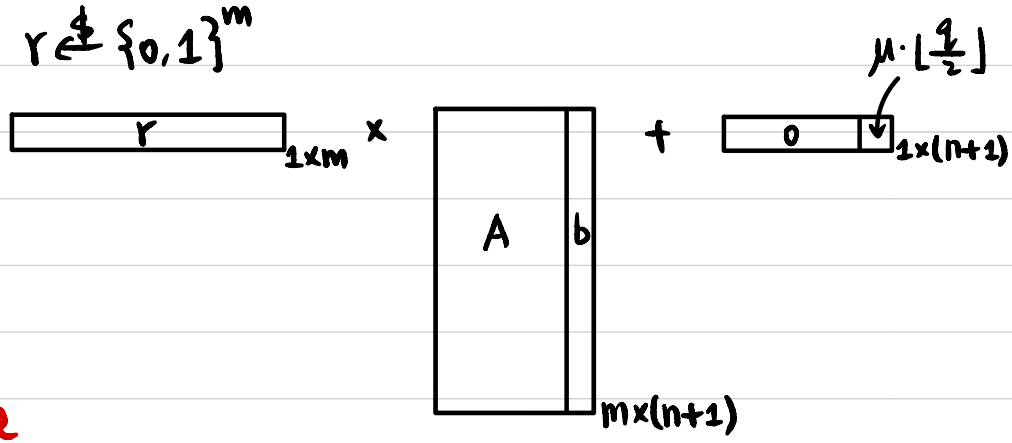


- $\mathrm{Enc}_{pk}(\mu)$: $\mu \in \{0,1\}$

  sample a random $S \subseteq [m]$

  $c = \left(\sum_{i \in S} A_i \,,\, \left(\sum_{i \in S} b_i \right) + \mu \cdot \lfloor \frac{q}{2} \rfloor \right)$

  A_i: i-th row of A



- $\mathrm{Dec}_{sk}(c)$: $c = [c_1 \mid c_2]$

  $c_2 - \langle c_1, s \rangle = \mu \cdot \lfloor \frac{q}{2} \rfloor + \sum_{i \in S} e_i$ (small noise)

Thm If $\mathrm{LWE}_{n,m,q,\chi}$ is (quantum) hard, then Regev encryption is (post-quantum) CPA-secure.

Proof Sketch