CSCI 1515 Applied Cryptography

This Lecture:

- RSA Assumption/Encryption (continued)
- Diffie-Hellman Assumptions
- ElGamal Encryption
- Diffie-Hellman Key Exchange

Basic Number Theory

$\gcd(a, N) = 1$: a & N are **coprime**

$\Rightarrow \exists b$ s.t. $a \cdot b \equiv 1 \pmod{N}$: a is **invertible modulo** N,
b is its **inverse**, denoted as a^{-1}.

$$\mathbb{Z}_N^* := \{a \mid a \in [1, N-1], \gcd(a, N) = 1\}$$

Euler's phi (totient) function $\phi(N) := |\mathbb{Z}_N^*|$

Ex: $N = p \cdot q$ (p, q are primes) $\phi(N) = (p-1) \cdot (q-1)$.

Euler's Theorem $\forall a, N$ where $\gcd(a, N) = 1$, $a^{\phi(N)} \equiv 1 \pmod{N}$.

Corollary If $d \equiv e^{-1} \bmod \phi(N)$, then $\forall a \in \mathbb{Z}_N^*$, $(a^d)^e \equiv a \bmod N$.

RSA Assumption

Randomly Sample → Miller-Rabin Primality Test

- **Factoring Assumption:**

  Generate two n-bit primes p, q $(p \neq q)$ (How?)

  Compute $N = p \cdot q$

  Given N, it's computationally hard to find p & q (classically)

- **RSA Assumption:**

  Generate two n-bit primes p, q $(p \neq q)$

  Compute $N = p \cdot q$, $\phi(N) = (p-1)(q-1)$

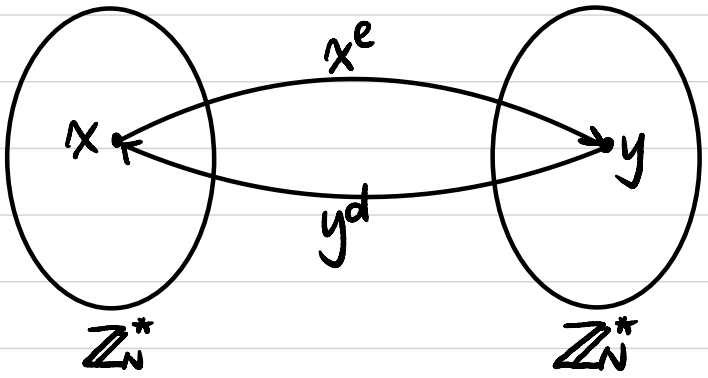
  Choose e s.t. $\gcd(e, \phi(N)) = 1$ (How? ← Small prime)

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

  Given (N, e) & a random $y \xleftarrow{\$} \mathbb{Z}_N^*$, it's computationally hard to find x s.t.

  $$x^e \equiv y \pmod N$$

If breaking Factoring → then break RSA



"Plain" RSA Encryption

$\lambda = 128$: best attack takes time 2^{128}

$n = 1024$, key length 2048

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

  Generate two n-bit p, q $(p \neq q)$

  Compute $N = p \cdot q$, $\phi(N) = (p-1)(q-1)$

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

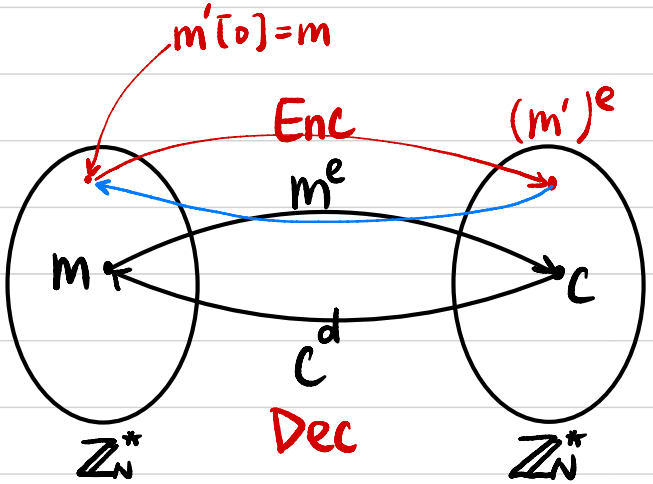
  Compute $d = e^{-1} \bmod \phi(N)$

  $pk = (N, e)$ $\quad sk = d$

  $m \in \{0,1\}$

- $\text{Enc}_{pk}(m)$: $c = m^e \bmod N$

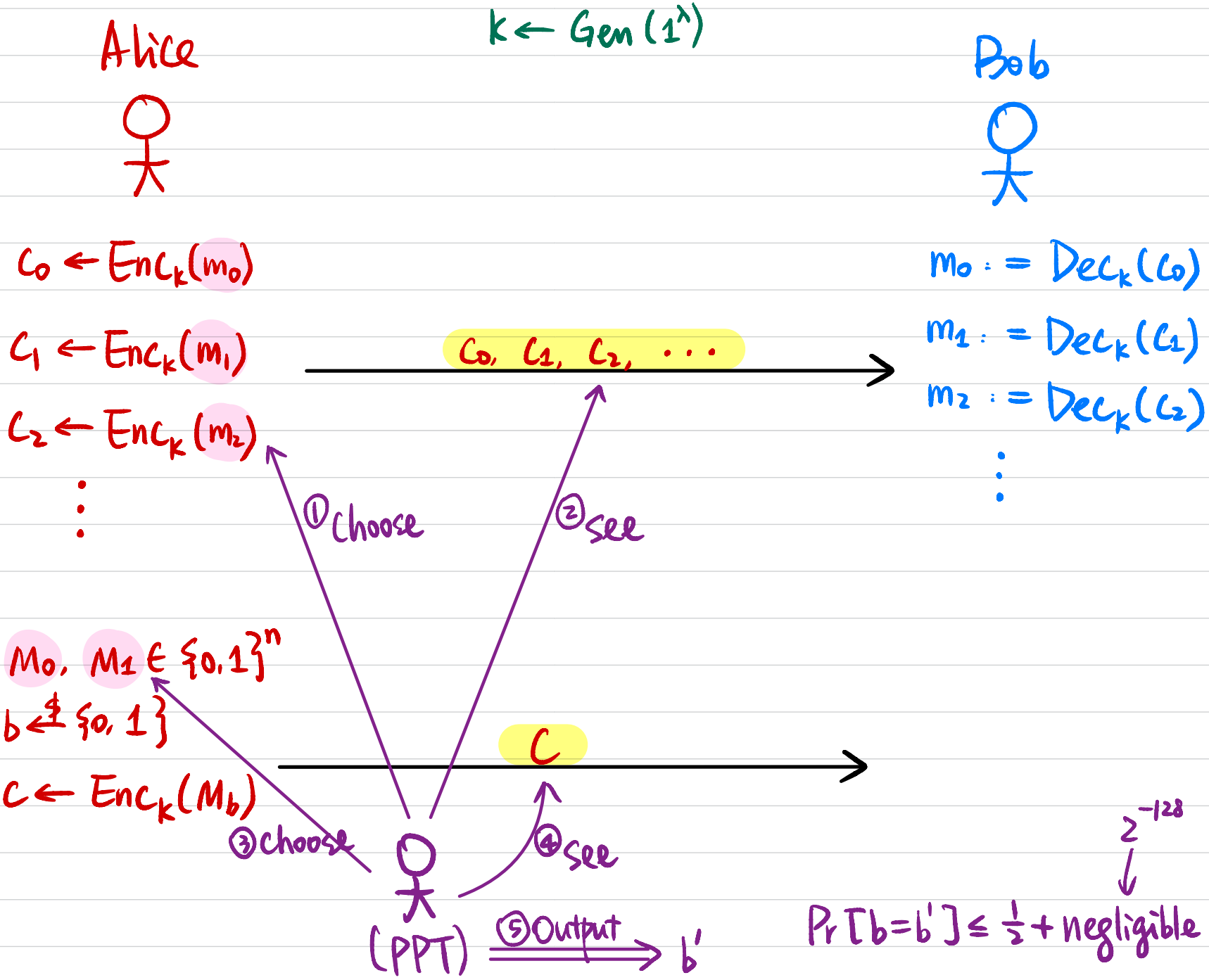
- $\text{Enc}_{sk}(c)$: $m = c^d \bmod N$



Any security issues? CPA secure?

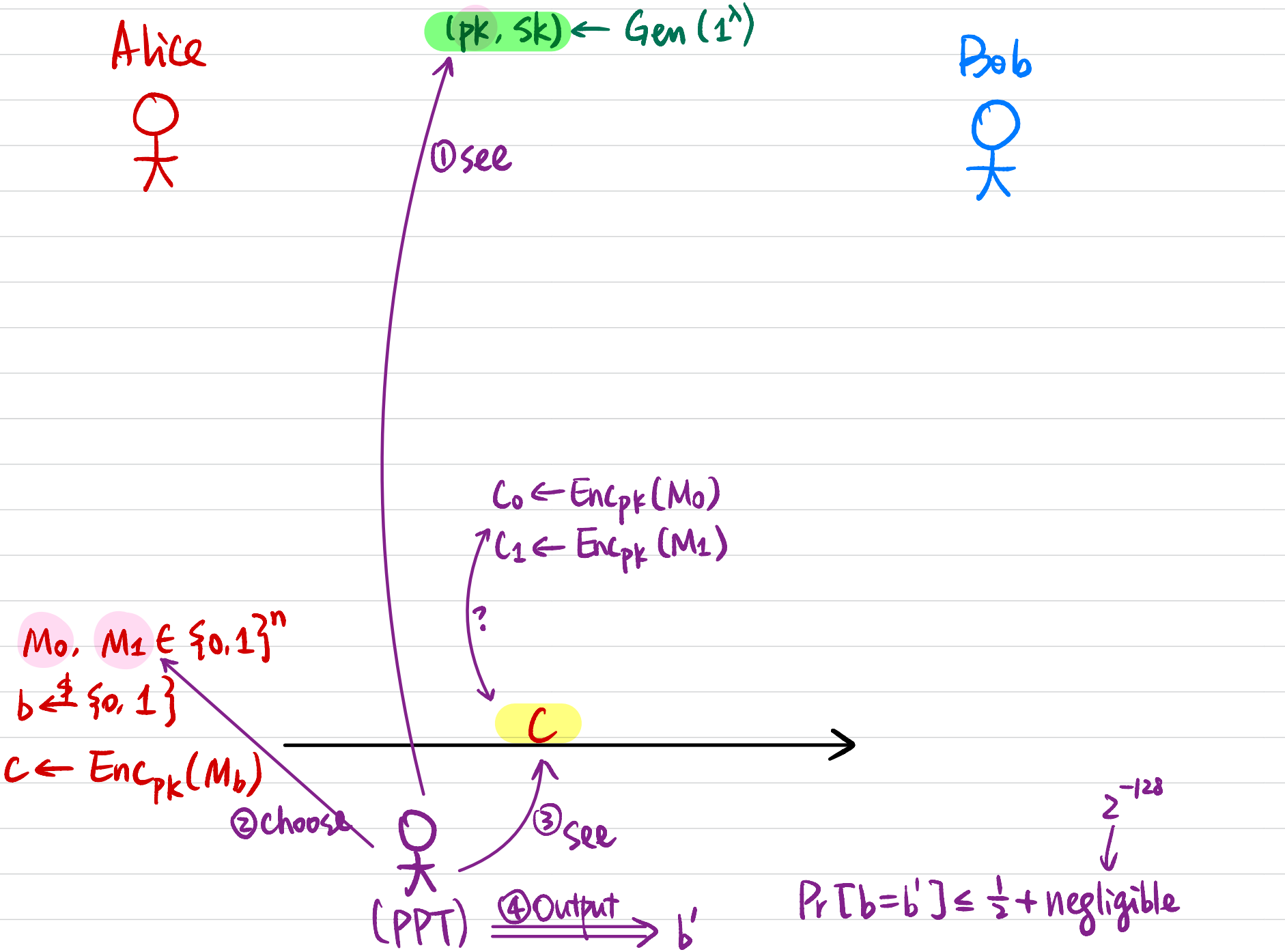
Computational Security

Chosen-Plaintext Attack (CPA) Security



Computational Security

Chosen-Plaintext Attack (CPA) Security



Basic Group Theory

Def A group is a set G along with a binary operation $\circ$ with properties:

① Closure: $\forall g, h \in G,\ g \circ h \in G$

② Existence of an identity: $\exists e \in G$ s.t. $\forall g \in G,\ e \circ g = g \circ e = g$.

③ Existence of inverse: $\forall g \in G,\ \exists h \in G$ s.t. $g \circ h = h \circ g = e$

Inverse of g denoted as g^{-1}.

④ Associativity: $\forall g_1, g_2, g_3 \in G,\ (g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$

We say a group is abelian if it satisfies:

⑤ Commutativity: $\forall g, h \in G,\ g \circ h = h \circ g$

Exercises: Is this a group?

- $(\mathbb{Z}, +)$ Yes. $e = 0$
- $(\mathbb{Z}, \cdot)$ No
- $(G = \{0, 1, \cdots, N-1\},\ + \bmod N)$ Yes, $e = 0$
- $(\mathbb{Z}_N^*,\ \cdot \bmod N)$ Yes, $e = 1$

Basic Group Theory

Group Exponentiation

For a group $(G, \cdot)$. $g^m := \overbrace{g \cdot g \cdots g}^{m}$ $\quad g^0 := e$ $\quad g^{-m} := (g^{-1})^m$

$g^{m_1} \cdot g^{m_2} = g^{m_1+m_2}$ $\quad (g^{m_1})^{m_2} = g^{m_1 \cdot m_2}$ $\quad g^m \cdot h^m = (g \cdot h)^m$ $\quad g^{-m} = (g^m)^{-1}$

Def For a **finite group**, we use $|G|$ to denote its **order** (# of elements)

Let G be a finite group of order m.

$\forall g \in G$, $\langle g \rangle := \{g^0, g^1, \cdots, g^{m-1}\}$ $(g^m = e)$ $\quad \underbrace{e, g, g^2, \cdots}_{|\langle g \rangle|}, e, g, g^2, \cdots, \cdots g^m = e$

G is a **cyclic group** if $\exists g \in G$ s.t. $\langle g \rangle = G$. g is a **generator** of G.

Thm $(\mathbb{Z}_p^*, \cdot \bmod p)$ for a prime p is a cyclic group of order $p-1$.

$p = 7$, $\langle 3 \rangle = \{1, 3, 2, 6, 4, 5\}$ $\quad \langle 2 \rangle = \{1, 2, 4\}$

Thm If G has prime order, then G is cyclic and every element except the identity is a generator.

Prime-Order Subgroups of $\mathbb{Z}_p^*$

Def For a group $(G, \cdot)$, $H \subseteq G$ is a **subgroup** of G if $(H, \cdot)$ forms a group.

A prime p is a **safe prime** if $p = 2q + 1$ and q is a prime.

$\mathbb{Z}_p^*$ is a cyclic group of order $p - 1 = 2q$.

Define $H := \{ x^2 \bmod p \mid x \in \mathbb{Z}_p^* \}$

Thm H is a subgroup of $\mathbb{Z}_p^*$ of order q.

$p = 7$, $H = \{1, 2, 4\} = \langle 2 \rangle = \langle 4 \rangle$

Diffie-Hellman Assumptions

Integer group key 2048-bit
Elliptic Curve group key 256-bit

$(G, q, g) \leftarrow G(1^\lambda)$ — $O(\lambda)$-bit integer

Cyclic group G of order q with generator g

- **Discrete Logarithm (DLOG) Assumption:**

  $x \xleftarrow{\$} \mathbb{Z}_q$, compute $h = g^x$ $\quad g^x \overset{?}{\Rightarrow} x$

  Given (G, q, g, h), it's computationally hard to find x (classically).

If breaking DLOG ⇓ Break CDH ⇓ Break DDH

- **Computational Diffie-Hellman (CDH) Assumption:**

  $x, y \xleftarrow{\$} \mathbb{Z}_q$, compute $h_1 = g^x$, $h_2 = g^y$ $\quad (g^x, g^y) \overset{?}{\Rightarrow} g^{xy}$

  Given (G, q, g, h_1, h_2), it's computationally hard to find g^{xy}.

- **Decisional Diffie-Hellman (DDH) Assumption:**

  $x, y, z \xleftarrow{\$} \mathbb{Z}_q$, compute $h_1 = g^x$, $h_2 = g^y$

  Given (G, q, g, h_1, h_2), it's computationally hard to distinguish

  $(g^x, g^y, g^{xy}) \overset{c}{\simeq} (g^x, g^y, g^z)$ between g^{xy} and g^z.

ElGamal Encryption

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

  $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^\lambda)$ ← can be re-used

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

  $pk = (\mathbb{G}, q, g, h)$ $\quad sk = x$

- $\text{Enc}_{pk}(m)$: $m \in \mathbb{G}$

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

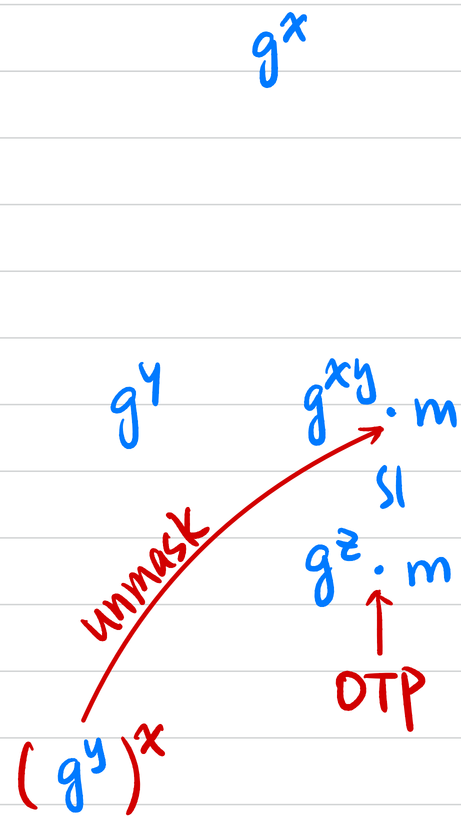
  $c = \langle g^y, h^y \cdot m \rangle$ → $(g^x)^y$

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

  $c = \langle c_1, c_2 \rangle$

  $m = c_2 \cdot (c_1^{sk})^{-1}$ → $[(g^y)^x]^{-1}$

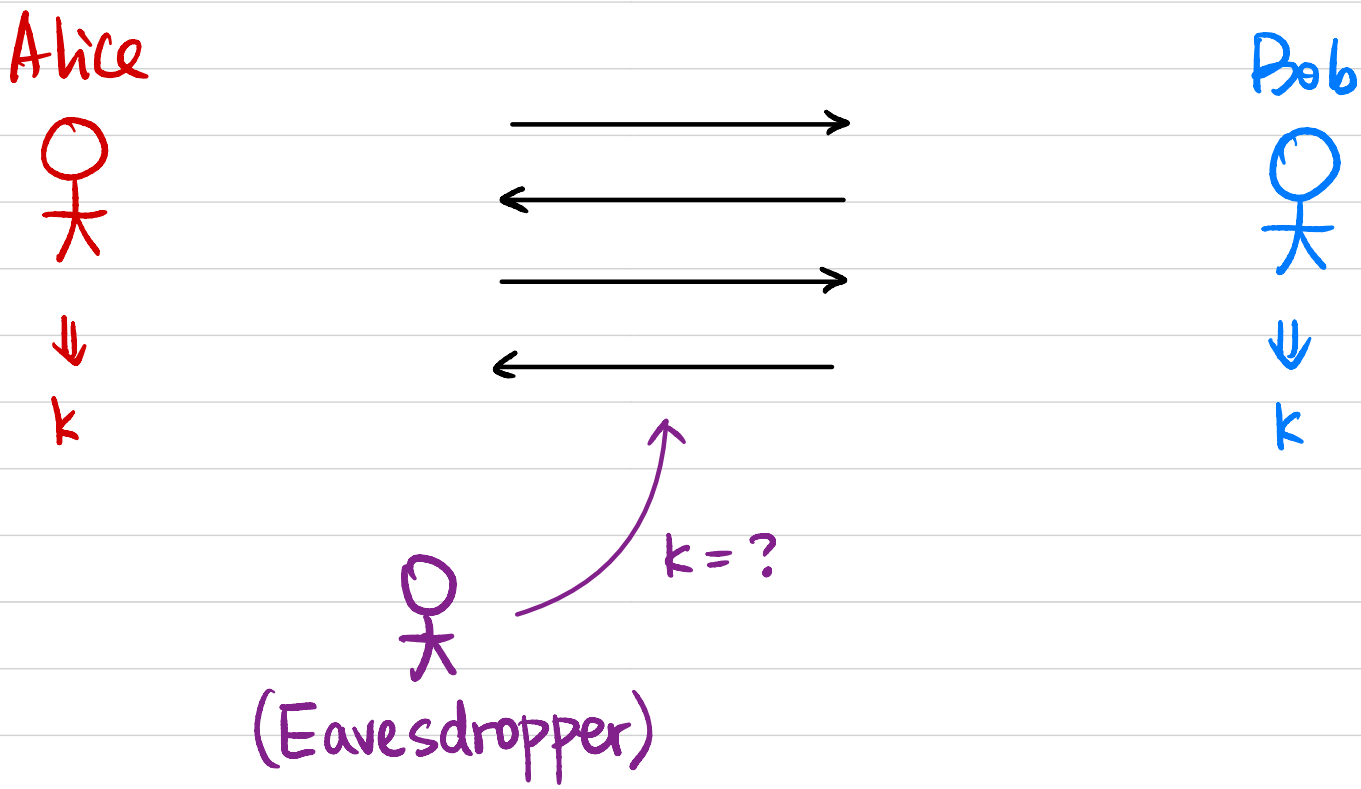
  c_2 → $g^{xy} \cdot m$



Correctness?

CPA Security?

Secure Key Exchange



Thm (Informal): It's impossible to construct secure key exchang from SKE in a black-box way.

Diffie-Hellman Key Exchange

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

$x \xleftarrow{\$} \mathbb{Z}_q$, compute $h_A = g^x$

(G, q, g, h_A) →

$y \xleftarrow{\$} \mathbb{Z}_q$, compute $h_B = g^y$

← h_B

$k = h_B^x = (g^y)^x$

$k = h_A^y = (g^x)^y$

$k = ?$

Correctness?

Security?

What happens in practice