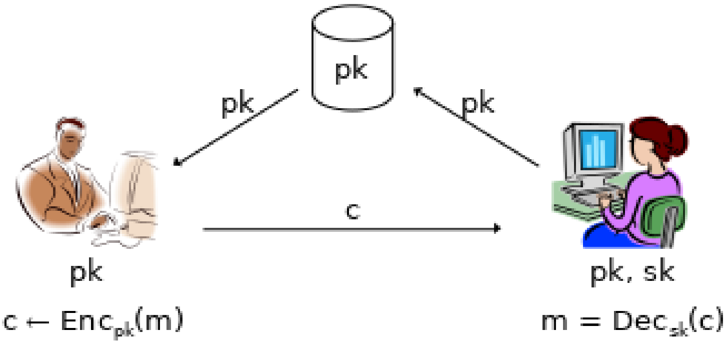
# Cryptography

Lecture 25

# Public-key encryption



# Public-key encryption

- A public-key encryption scheme is composed of three PPT algorithms:
  - Gen: *key-generation algorithm* that on input $1^n$ outputs $pk, sk$
  - Enc: *encryption algorithm* that on input $pk$ and a message $m$ outputs a ciphertext $c$
  - Dec: *decryption algorithm* that on input $sk$ and a ciphertext $c$ outputs a message $m$ or an error $\perp$

$$\forall m, pk, sk \text{ output by Gen, } Dec_{sk}(Encpk(m)) = m$$

# CPA-security

- Fix a public-key encryption scheme $\Pi$ and an adversary $A$
- Define experiment $PubK - CPA_{A,\Pi}(n)$:
  - Run $Gen(1^n)$ to get keys $pk, sk$
  - Give $pk$ to $A$, who outputs $(m_0, m_1)$ of same length
  - Choose uniform $b \in \{0, 1\}$ and compute the ciphertext $c \leftarrow Enc_{pk}(m_b)$; give $c$ to $A$
  - $A$ outputs a guess $b'$ and the experiment evaluates to 1 if $b' = b$

# CPA-security

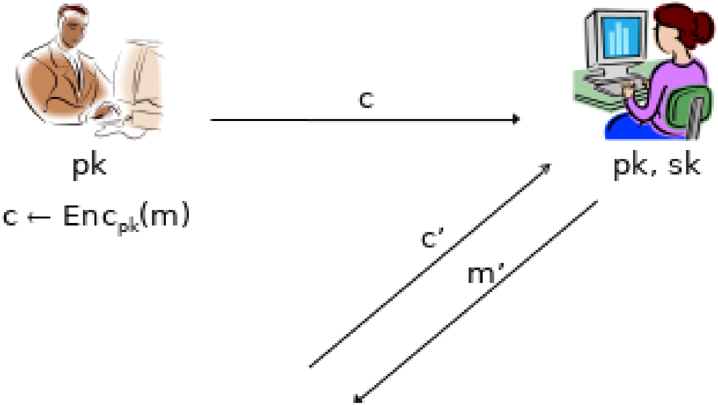
- Public-key encryption scheme $\Pi$ is *CPA-secure* if for all PPT adversaries $A$:

$$Pr[PubK - CPA_{A,\Pi}(n) = 1 \leq \frac{1}{2} + negl(n)$$

# Notes on the definition

- No encryption oracle?!
  - Encryption oracle redundant in public-key setting
- No *perfectly secret* public-key encryption
- No *deterministic* public-key encryption scheme can be CPA-secure
- CPA-security implies security for encrypting multiple messages as in the private-key case

# Chosen-ciphertext attacks



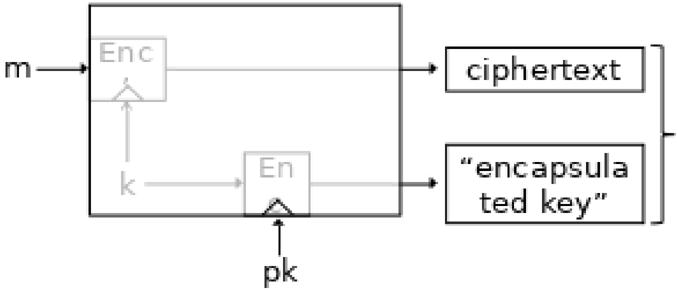
# Chosen-ciphertext attacks

- Chosen-ciphertext attacks are arguable even a greater concern in the public-key setting
  - Attacker might be a legitimate sender
  - Easier for attacker to obtain full decryptions of ciphertexts of its choice
- Related concern: *malleability*
  - i.e. given a ciphertext $c$ that is the encryption of an unknown message $m$, might be possible to produce ciphertext $c'$ that decrypts to a related message $m'$
  - This is also undesirable in the public-key setting

# Chosen-ciphertext attacks

- Can define CCA-security for public-key encryption by analogy to the definition for private-key encryption
  - See book for details

# Hybrid encryption




- (Decryption done in the obvious way)
- The *functionality* of public-key encryption at the (asymptotic) *efficiency* of private-key encryption

# Security of hybrid encryption

- Let $\Pi$ be the public-key component, and $\Pi'$ the private-key component; let $\Pi_{hy}$ denote their combination
- If $\Pi$ is a CPA-secure public-key scheme, and $\Pi'$ is a CPA-seucre private-key scheme, then $\Pi_{hy}$ is a CPA-secure public-key scheme
  - Similarly for CCA-security

# KEM/DEM paradigm

- For hybrid encryption, something *weaker* than public key encryption would suffice
- Sufficient to have an "encapsulation algorithm" that takes a public key and outputs a ciphertext/key pair $(c, k)$
  - Correctness: $k$ is recoverable from $c$ given $sk$
  - Security: $k$ is indistinguishable from uniform given $pk$ and $c$
- This can lead to more-efficient constructions

# Dlog-based PKE

# Diffie-Hellman key exchange

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

$x \leftarrow \mathbb{Z}_q$

$h_1 = g^x$

$k = (h_2)^x$

$m = c_2/k$

$G, q, g, h_1 \longrightarrow$

$\longleftarrow h_2$

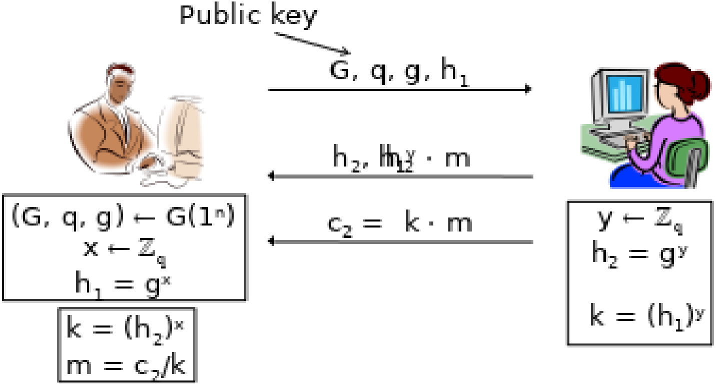
$\longleftarrow c_2 = k \cdot m$

$y \leftarrow \mathbb{Z}_q$

$h_2 = g^y$

$k = (h_1)^y$

# El Gamal encryption



# El Gamal encryption

- $Gen(1^n)$
  - Run $G(1^n)$ to obtain $G, q, g$. Choose uniform $x \in \mathbb{Z}_q$. The public key is $(G, q, g, g^x)$ and the private key is $x$
- $Enc_{pk}(m)$, where $pk = (G, q, g, h)$ and $m \in G$
  - Choose uniform $y \in \mathbb{Z}_q$. The ciphertext is $(g^y, h^y \cdot m)$
- $Dec_{sk}(c_1, c_2)$
  - Output $c_2/c_1^x$

# Security?

- If the DDH assumption is hard for $G$, then the El Gamal encryption scheme is CPA-secure
  - Follows from security of Diffie-Hellman key exchange, or can be proved directly
- Discrete-logarithm assumption alone is not enough here

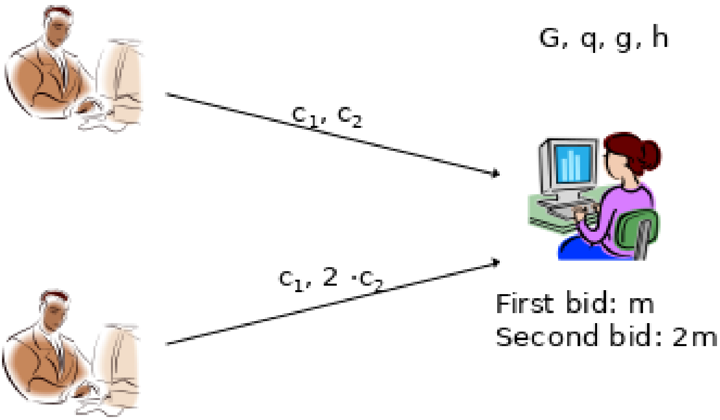
# In practice...

- Parameters $g, q, g$ are standardized and shared
- Inconvenient to treat message as group element
  - Use *key derivation* to derive a key $k$ instead, and use $k$ to encrypt the message
  - i.e. ciphertext is $(g^y, Enc'_k(m))$ where $k = H(h^y)$
  - Can be analyzed using KEM/DEM paradigm

# Chosen-ciphertext attacks?

- El Gamal encryption is *not secure* against chosen-ciphertext attacks
  - Follows from the fact that it is *malleable*
- Given ciphertext $c_1, c_2$, transform it to obtain the ciphertext $c_1, c_2' = c_1, \alpha \cdot c_2$ for arbitrary $\alpha$
  - Since $c_1, c_2 = g^y, h^y \cdot m$, we have $c_1, c_2' = g^y, h^y \cdot (\alpha m)$
  - i.e. encryption of $m$ becomes an encryption of $\alpha m$

# Attack!
# (Assume $2 \in G \subset \mathbb{Z}_p^*$)



# Chosen-ciphertext security

- Use key derivation coupled with CCA-secure private-key encryption scheme
  - i.e. ciphertext is $(g^y, Enc'_k(m))$ where $k = H(h^y)$ and $Enc'$ is a CCA-seucre scheme
- Can be proved CCA-seucre under appropriate assumptions, if $H$ is modeled as a random oracle
- DHIES / ECIES