Public-Key Cryptography

Lecture 7

Public-Key Encryption

Diffie-Hellman Key-Exchange

El Gamal Encryption

PKE scheme

Shared/Symmetric-Key Encryption (a.k.a. private-key encryption)

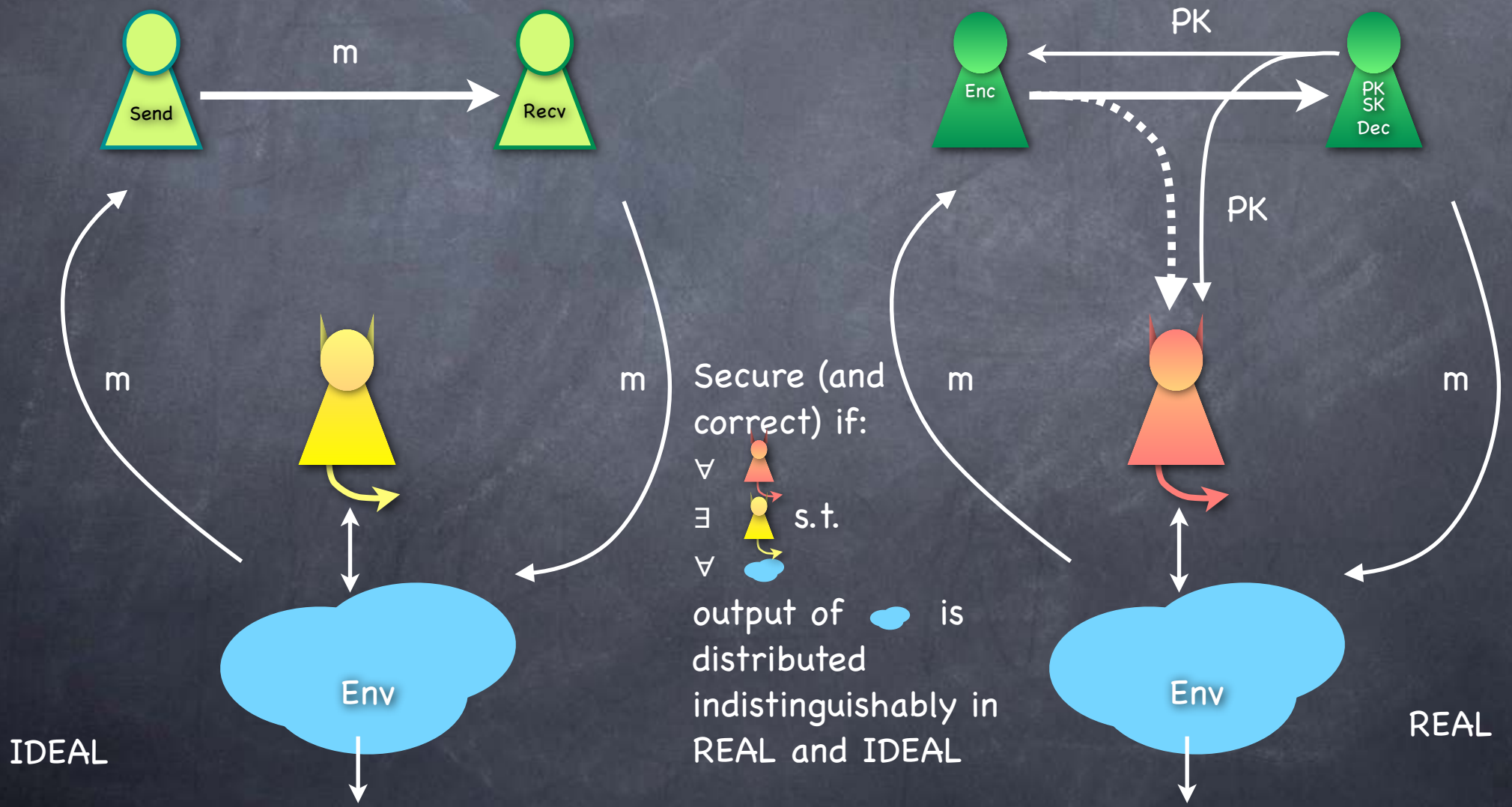
- SKE:
  - Syntax
    - KeyGen outputs $K \leftarrow \mathcal{K}$
    - Enc: $\mathcal{M} \times \mathcal{K} \times \mathcal{R} \rightarrow \mathcal{C}$
    - Dec: $\mathcal{C} \times \mathcal{K} \rightarrow \mathcal{M}$
  - Correctness
    - $\forall K \in$ Range(KeyGen), Dec(Enc(m,K), K) = m
  - Security (SIM/IND-CPA)

- PKE

a.k.a. asymmetric-key encryption

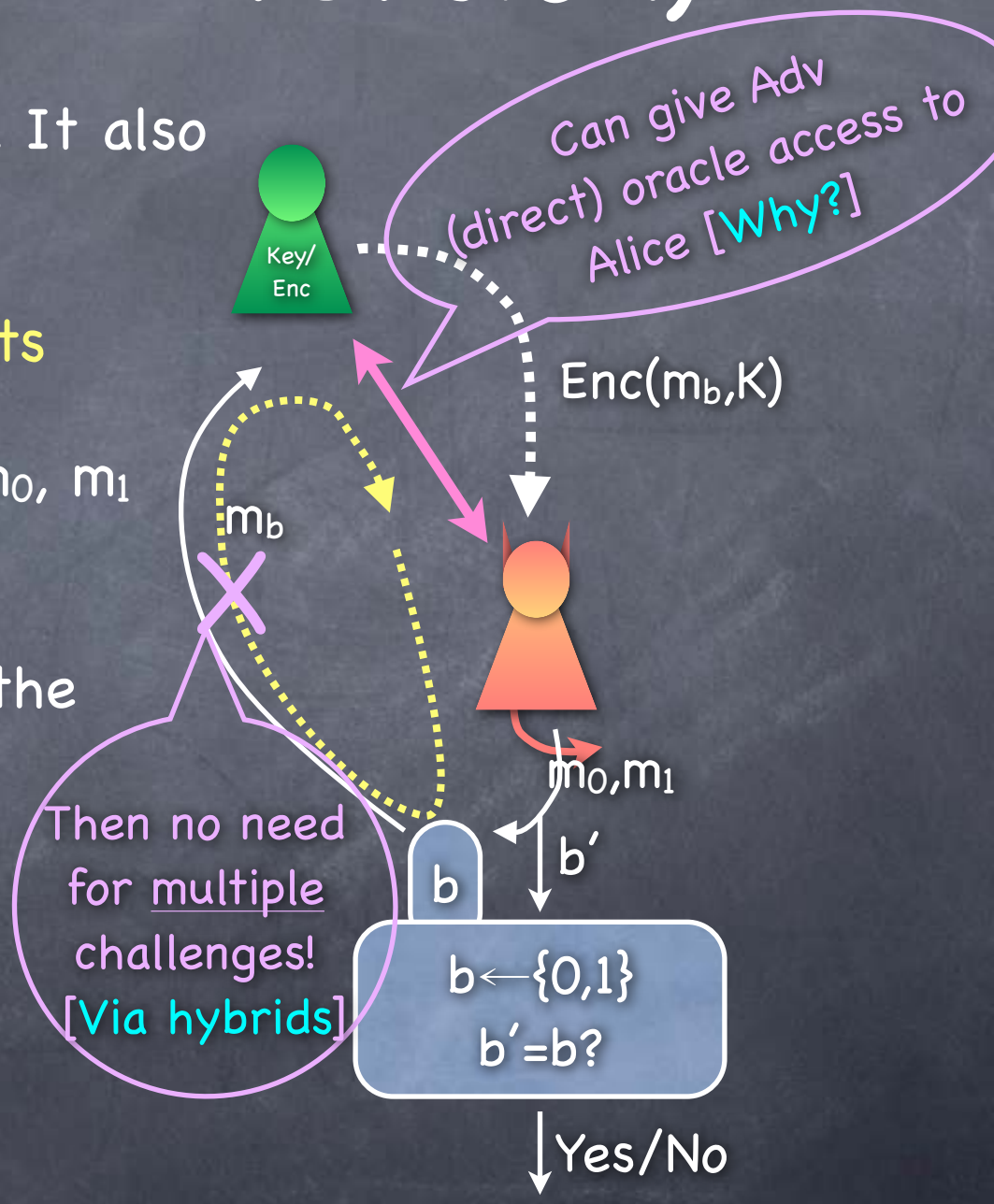
  - Syntax
    - KeyGen outputs $(PK,SK) \leftarrow \mathcal{PK} \times \mathcal{SK}$
    - Enc: $\mathcal{M} \times \mathcal{PK} \times \mathcal{R} \rightarrow \mathcal{C}$
    - Dec: $\mathcal{C} \times \mathcal{SK} \rightarrow \mathcal{M}$
  - Correctness
    - $\forall (PK,SK) \in$ Range(KeyGen), Dec(Enc(m,PK), SK) = m
  - Security (SIM/IND-CPA, PKE version)

SIM-CPA (PKE Version)



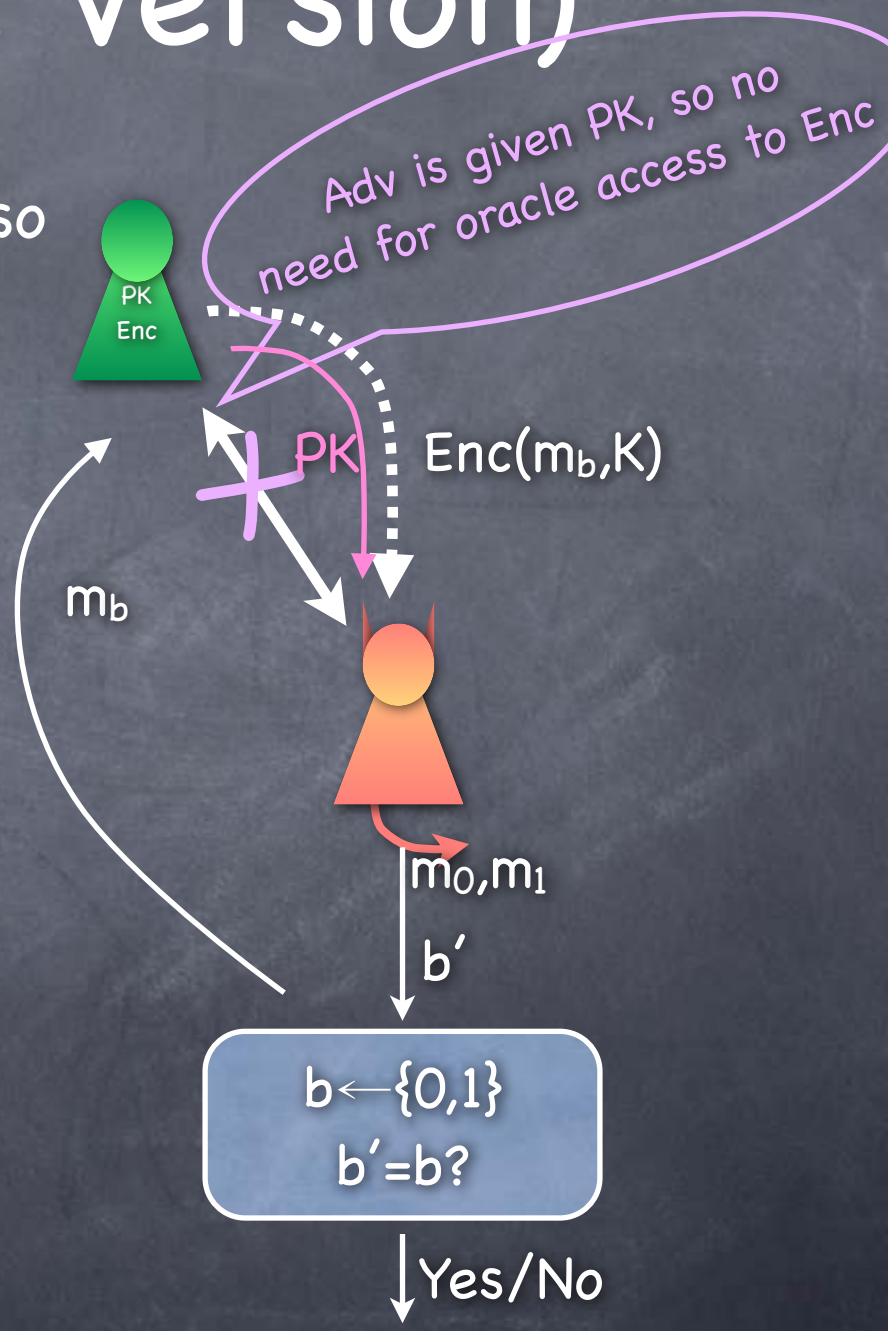
IND-CPA (SKE version)

- Experiment picks a random bit b. It also runs KeyGen to get a key K
  - For as long as Adversary wants
    - Adv sends two messages m_0, m_1 to the experiment
    - Expt returns Enc(m_b,K) to the adversary
  - Adversary returns a guess b′
  - Experiment outputs 1 iff b′=b
- IND-CPA secure if for all PPT adversaries $Pr[b'=b] - 1/2 \leq \nu(k)$



IND-CPA (~~SKE~~ PKE version)

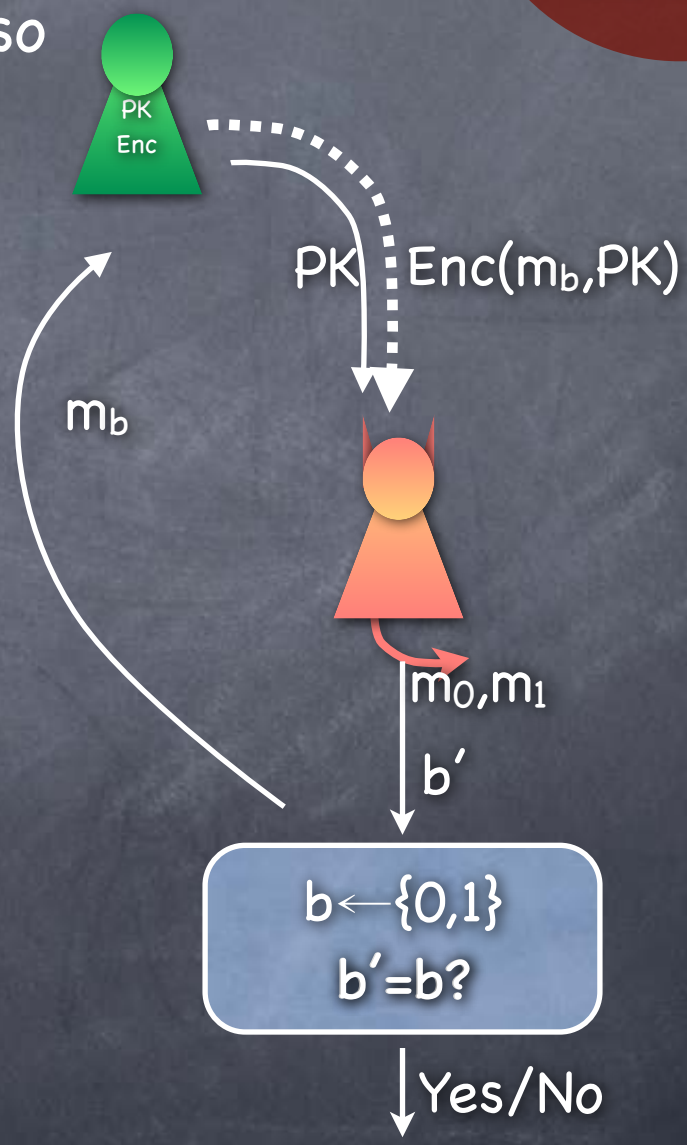
- Experiment picks a random bit b. It also runs KeyGen to get a key (PK,SK). Adv given PK
  - Adv sends two messages m_0, m_1 to the experiment
  - Expt returns Enc(m_b,K) to the adversary
  - Adversary returns a guess b′
  - Experiment outputs 1 iff b′=b
- IND-CPA secure if for all PPT adversaries $Pr[b'=b] - 1/2 \leq \nu(k)$



IND-CPA (PKE version)

IND-CPA + ~correctness equivalent to SIM-CPA

- Experiment picks a random bit b. It also runs KeyGen to get a key (PK,SK). Adv given PK
  - Adv sends two messages m_0, m_1 to the experiment
  - Expt returns Enc(m_b,K) to the adversary
  - Adversary returns a guess b′
  - Experiment outputs 1 iff b′=b
- IND-CPA secure if for all PPT adversaries $Pr[b'=b] - 1/2 \leq \nu(k)$



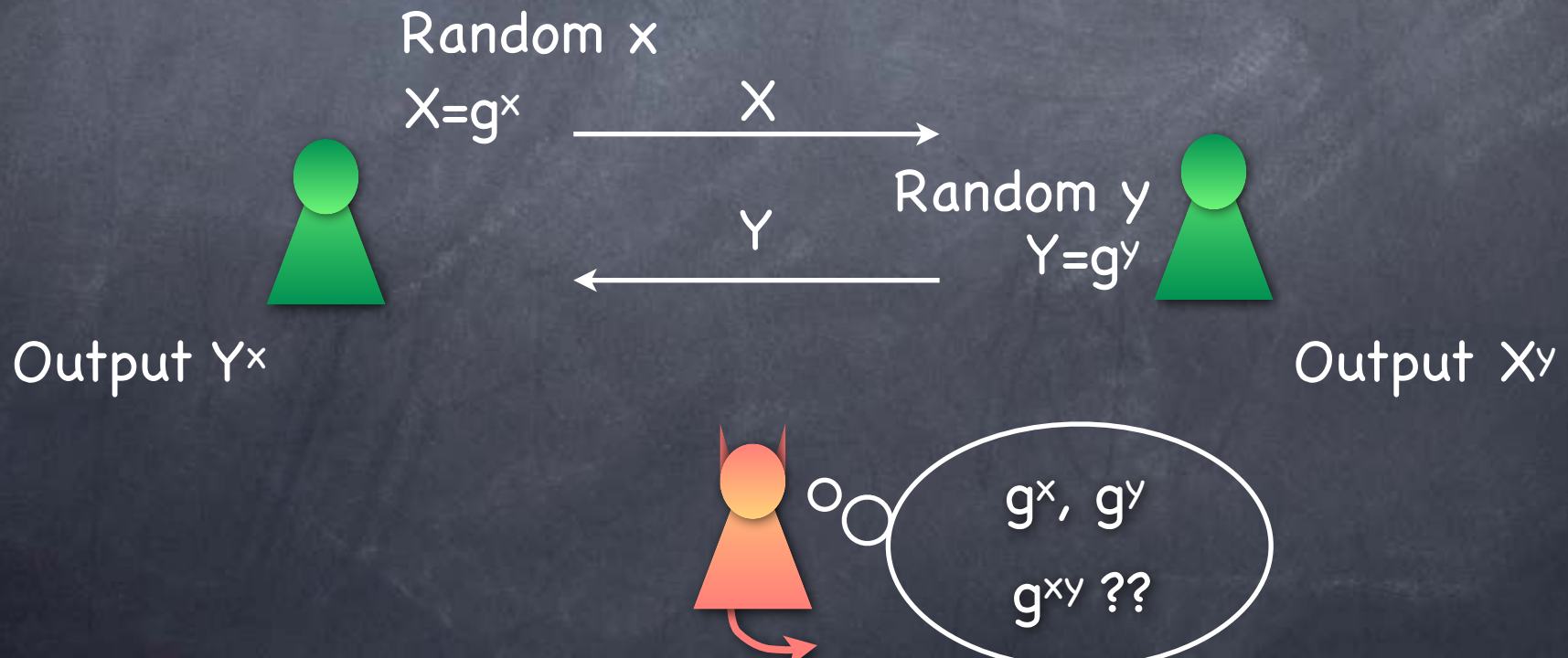
Perfect Secrecy?

- No perfectly secret and correct PKE (even for one-time encryption)
  - Public-key and ciphertext (the total shared information between Alice and Bob at the end) should together have entire information about the message
    - Intuition: If Eve thinks Bob could decrypt it as two messages based on different SKs, Alice should be concerned too
    - i.e., Alice conveys same information to Bob and Eve
- PKE only with computational security

Unless assumptions of imperfect eavesdropping

Diffie-Hellman Key-exchange

- A candidate for how Alice and Bob could generate a shared key, which is "hidden" from Eve



Why DH-Key-exchange could be secure

- Given g^x, g^y for random x, y, g^{xy} should be "hidden"
  - i.e., could still be used as a pseudorandom element
  - i.e., $(g^x, g^y, g^{xy}) \approx (g^x, g^y, R)$
- Is that reasonable to expect?
  - Depends on the "group"

Groups, by examples

- A group $(G, *)$ specified by a set G (for us finite, unless otherwise specified) and a "group operation" $*$ that is associative, has an identity, is invertible, and (for us) commutative

Abelian

- Examples: $\mathbb{Z}$ = (integers, +) (this is an infinite group),
  $\mathbb{Z}_N$ = (integers modulo N, + mod N),
  G^n = (Cartesian product of a group G, coordinate-wise operation)

Direct Product

- Order of a group G: $|G|$ = number of elements in G
- For any $a \in G$, $a^{|G|} = a * a * \ldots * a$ ($|G|$ times) = identity

Lagrange's theorem

- Finite Cyclic group (in multiplicative notation): there is one element g such that $G = \{g^0, g^1, g^2, \ldots g^{|G|-1}\}$
  - Prototype: $\mathbb{Z}_N$ (additive group), with g=1
    - or any d s.t. gcd(d,N) = 1

Computing on a Group

- We need groups with efficient algorithms to work on them
  - An ensemble of groups, indexed by security parameter
  - Group generation: Given a security parameter, output a group G and a generator for it, g
  - Elements of G should have (about) k-bit representation
    - Note: |G| can be exponentially large in k
  - G has polynomial time algorithms for adding, inverting and randomly sampling a group element

Discrete Log Assumption

Repeated squaring

- Discrete Log (w.r.t g) in a (multiplicative) cyclic group G generated by g: $DL_g(X)$:= unique x such that $X = g^x$ ($x \in \{0,1,\ldots,|G|-1\}$)
- In a (computationally efficient) group, given integer x and the standard representation of a group element g, can efficiently find the standard representation of $X=g^x$ (How?)
  - But given X and g, may not be easy to find x (depending on G)
  - DLA: Every PPT Adv has negligible success probability in the DL Expt: (G,g)←GroupGen; X←G; Adv(G,g,X)→z; $g^z=X$?
- If DLA broken, then Diffie-Hellman key-exchange broken
  - Eve gets x, y from g^x, g^y (sometimes) and can compute g^{xy} herself
    - A "key-recovery" attack
  - Note: could potentially break pseudorandomness without breaking DLA too

OWF: Raise(x;G,g) = $(g^x;G,g)$

Decisional Diffie-Hellman (DDH) Assumption

- $\{(g^x, g^y, g^{xy})\}_{(G,g)\leftarrow\textbf{GroupGen}; x,y\leftarrow[|G|]} \approx \{(g^x, g^y, g^r)\}_{(G,g)\leftarrow\textbf{GroupGen}; x,y,r\leftarrow[|G|]}$
- At least as strong as DLA
  - If DDH assumption holds, then DLA holds [Why?]
- But possible that DLA holds and DDH assumption doesn't
  - e.g.: DLA is widely believed to hold in $\mathbb{Z}_p^*$ (p prime), but DDH assumption doesn't hold there!

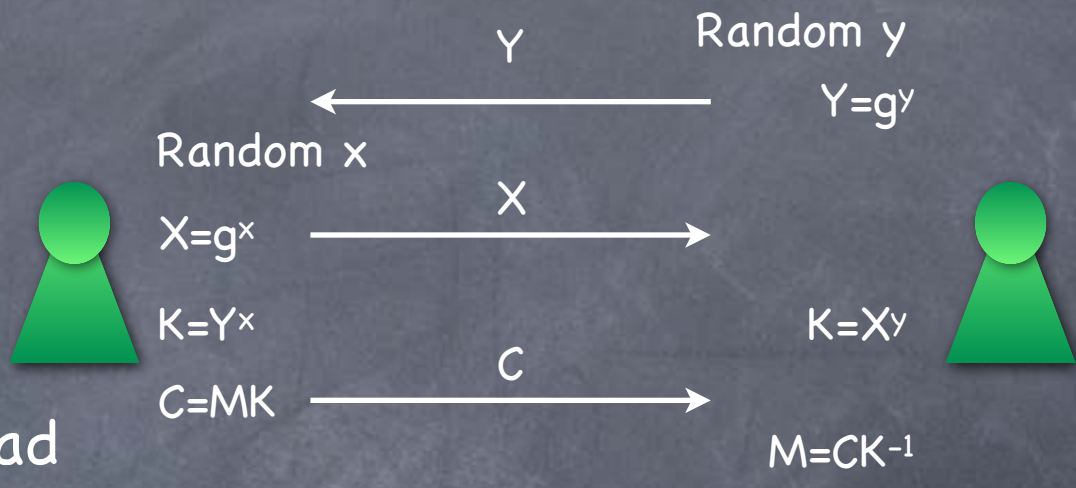
Group elements are non-zero elements mod p and group operation is multiplication mod p

- DH Key exchange is secure (against an eavesdropper) iff the DDH assumption holds in the group used

Security definition here is simply that (transcript, generated key) ≈ (transcript, random key)

El Gamal Encryption

- Based on DH key-exchange
  - Alice, Bob generate a key using DH key-exchange
  - Then use it as a one-time pad for messages in the group
- Bob's "message" in the key-exchange is his PK
- Alice's message in the key-exchange and the ciphertext of the one-time pad together form a single ciphertext



KeyGen: PK=(G,g,Y), SK=(G,g,y)

$Enc_{(G,g,Y)}(M) = (X=g^x, C=MY^x)$

$Dec_{(G,g,y)}(X,C) = CX^{-y}$

- KeyGen uses GroupGen to get (G,g)
- x, y uniform from $\mathbb{Z}_{|G|}$
- Message encoded into group element, and decoded

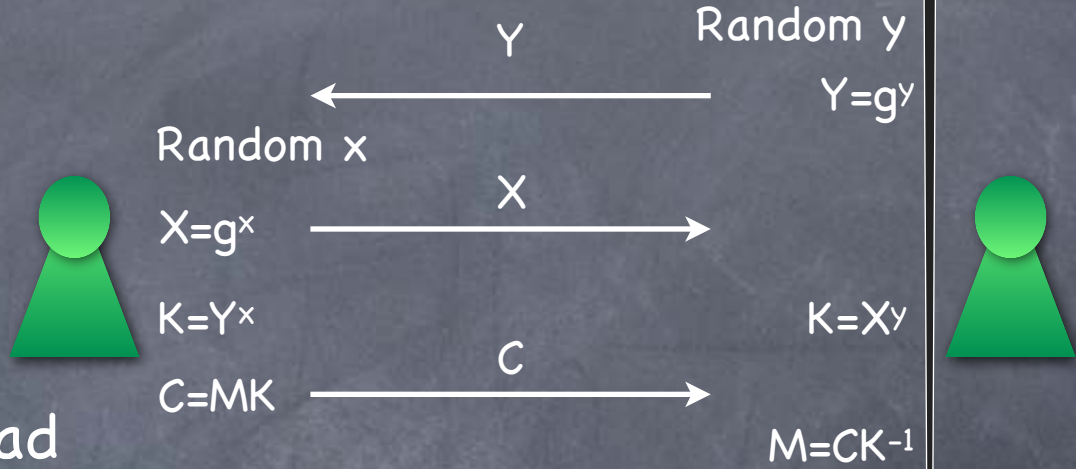
Security of El Gamal

- El Gamal is IND-CPA secure if DDH holds (for the collection of groups used)
  - Construct a DDH adversary A* given an IND-CPA adversary A
  - A*(G,g; g^x,g^y,g^z) (where (G,g) ← GroupGen, x,y random and z=xy or random) plays the IND-CPA experiment with A:
    - But sets PK=(G,g,g^y) and Enc(M_b)=(g^x,$M_b g^z$)
    - Outputs 1 if experiment outputs 1 (i.e. if b=b')
  - When z=random, A* outputs 1 with probability = 1/2
  - When z=xy, exactly IND-CPA experiment: A* outputs 1 with probability = 1/2 + advantage of A.

Alternately, convert the key K into a pseudorandom bit string using a "Key Derivation Function"

El Gamal Encryption

- Based on DH key-exchange
  - Alice, Bob generate a key using DH key-exchange
  - Then use it as a one-time pad for messages in the group
- Bob's "message" in the key-exchange is his PK
- Alice's message in the key-exchange and the ciphertext of the one-time pad together form a single ciphertext



KeyGen: PK=(G,g,Y), SK=(G,g,y)

$Enc_{(G,g,Y)}(M) = (X=g^x, C=MY^x)$

$Dec_{(G,g,y)}(X,C) = CX^{-y}$

- KeyGen uses GroupGen to get (G,g)
- x, y uniform from $\mathbb{Z}_{|G|}$
- Message encoded into group element, and decoded