CIS 5560

Cryptography
Lecture 15

Recap of Last Lecture(s)

- Number Theory refresher
  - Arithmetic modulo primes
  - Fermat's Little Theorem
  - Cyclic groups
  - Discrete Logarithms
- Key Exchange
  - Merkle puzzles
  - Diffie—Hellman
    - Computational Diffie—Hellman Problem

The Multiplicative Group $\mathbb{Z}_p^*$

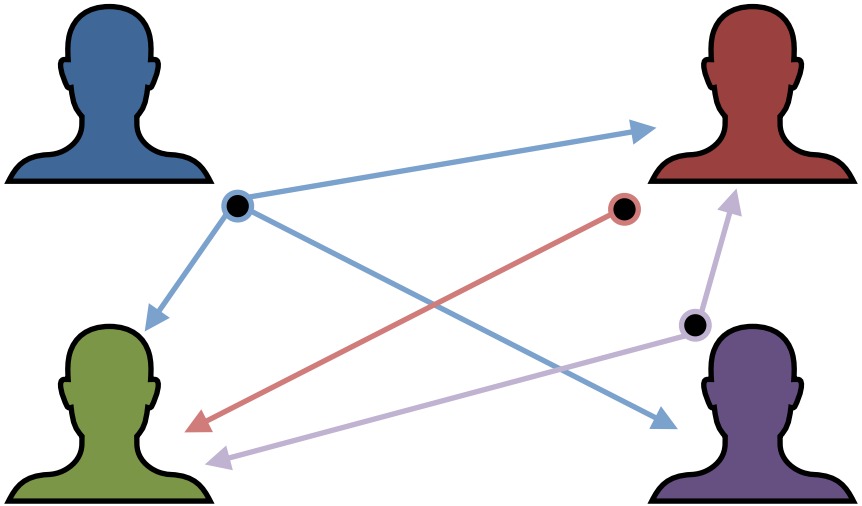
$\mathbb{Z}_p^*$: $(\{1,\ldots,p-1\}$, group operation: $\bullet$ mod $p)$

- Computing the group operation is easy.
- Computing inverses is easy: Extended Euclid.
- Exponentiation (given $g \in \mathbb{Z}_p^*$ and $x \in \mathbb{Z}_{p-1}$, find g^x mod p) is easy: **Repeated Squaring Algorithm.**
- The discrete logarithm problem (given a generator g and $h \in \mathbb{Z}_p^*$, find $x \in \mathbb{Z}_{p-1}$ s.t. $h = g^x$ mod p) is **hard**, to the best of our knowledge!

Key management

Goal: n users want to communicate with each other securely.

Problem: Storing mutual secret keys is onerous!



Total: $O(n)$ keys per user

Key question

Can we generate shared keys without an **online** trusted 3rd party?

Answer: yes!

Starting point of public-key cryptography:

- Merkle (1974), Diffie-Hellman (1976), RSA (1977)
- More recently: ID-based enc. (BF 2001), Functional enc. (BSW 2011)

Today's Lecture

- Diffie—Hellman key exchange
  - Computational Diffie—Hellman Problem
- Public Key Encryption
  - El Gamal Encryption
    - Decisional Diffie—Hellman Problem.

The Diffie—Hellman protocol

Fix a DL-hard group $\mathbb{G}$ of prime order p. Fix generator g of $\mathbb{G}$

Alice

Bob

choose random a in $\{1,\ldots,p-1\}$

choose random b in $\{1,\ldots,p-1\}$

"Alice", $A := g^a$

"Bob", $B := g^b$

$$B^a = g^{ba} = k_{AB} = g^{ab} = g^{ab} = A^b$$

Security

Eavesdropper sees: $\mathbb{G}, g, A = g^a, B = g^b$

Can she compute g^{ab} ??

Formally, define $\mathrm{DH}_g(g^a, g^b) = g^{ab}$

How hard is the DH function mod p?

New Assumption: Computational Diffie—Hellman

Let $\mathbb{G}$ be a finite cyclic group and g a generator of $\mathbb{G}$

Def: We say that **CDH is hard in** $\mathbb{G}$ if for all efficient algorithms A:

$$\Pr_{h \leftarrow G, a, b \leftarrow \mathbb{Z}_p}[A(\mathbb{G}, p, g, g^a, g^b) = g^{ab}] < \mathsf{negl}(p)$$

Example candidates:

(1) $\mathbb{Z}_p^*$ for large p,

(2) Elliptic curve groups mod p

Active Security against Malicious Attackers?

What can an attacker sitting between Alice and Bob do?

Active Security against Malicious Attackers?

What can an attacker sitting between Alice and Bob do?

Alice

$a \leftarrow \mathbb{Z}_p$

Mal

$a', b' \leftarrow \mathbb{Z}_p$

Bob

$b \leftarrow \mathbb{Z}_p$

"Alice", $A := g^a$

"Bob", $B' := g^{b'}$

"Alice", $A' := g^{a'}$

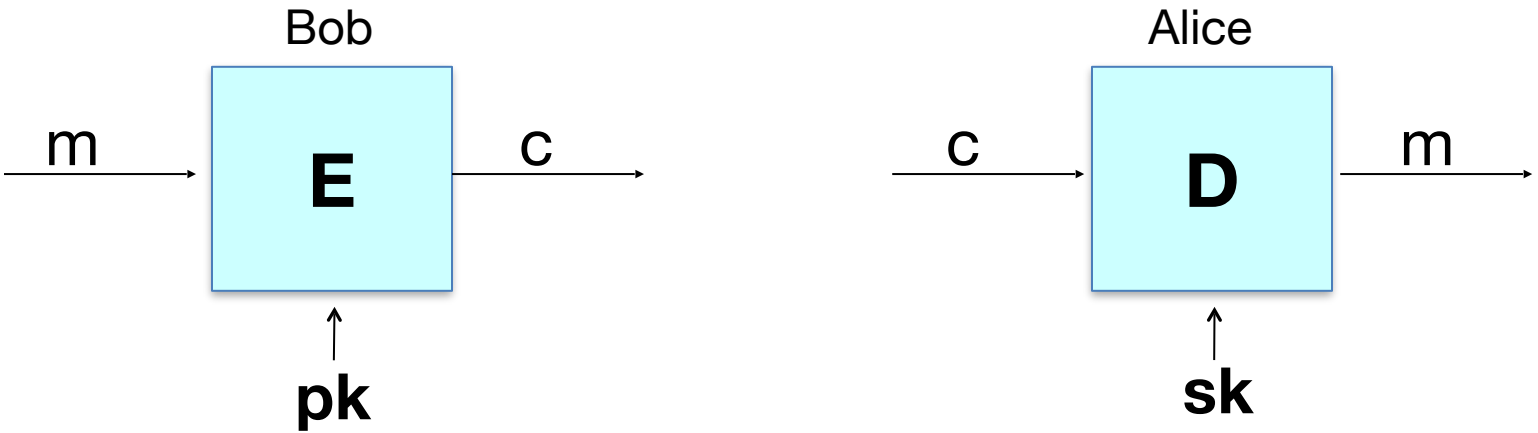
"Bob", $B := g^b$

$k_{AB'} = g^{ab'}$

$k_{A'B} = g^{a'b}$

Public key encryption

Alice: generates (PK, SK) and gives PK to Bob



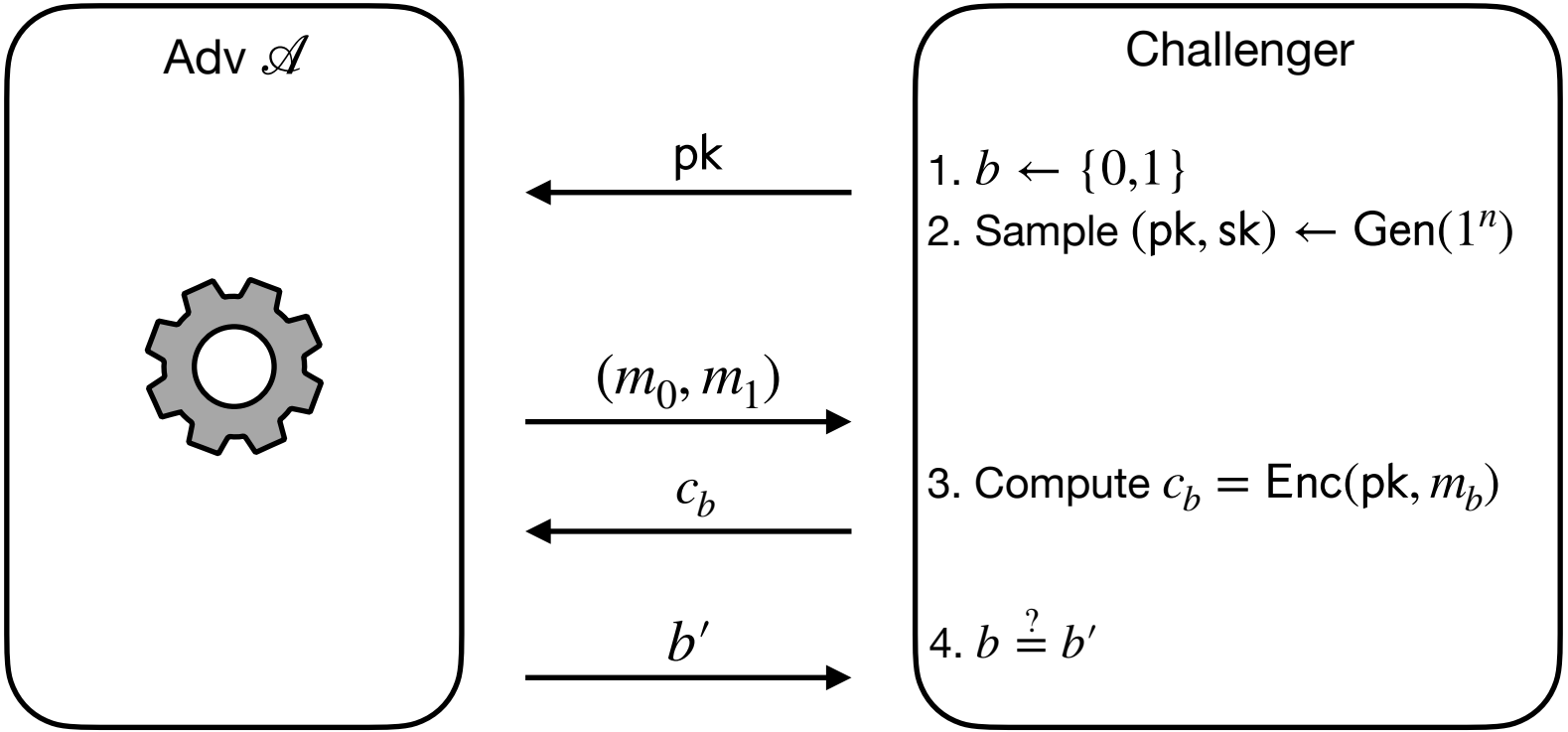
Public key encryption

Def: a public-key encryption system is a triple of algorithms (Gen, Enc, Dec)

- Gen: randomized algorithm that outputs a key pair (pk, sk)
- Enc(pk, m): randomized; takes $m \in \mathcal{M}$ and outputs $c \in \mathcal{C}$
- Dec(sk, c): deterministic; takes $c \in \mathcal{C}$ and outputs $m \in \mathcal{M} \cup \{ \perp \}$

Correctness: $\forall$(pk, sk) output by Gen, $\forall m \in \mathcal{M}$, Dec(sk, Enc(pk, m)) $= m$

Security: IND-CPA for PKE



$$\Pr[b = b'] = 1/2 + \mathsf{negl}(n)$$

Security: IND-CPA for PKE

For all PPT adversaries $\mathcal{A}$, the following holds:

$$\Pr\left[b = \mathcal{A}(\mathsf{Enc}(\mathsf{pk}, m_b)) \;\middle|\; \begin{array}{c} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n) \\ \text{Sample } b \leftarrow \{0,1\} \\ (m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pk}) \end{array} \right] \leq \mathsf{negl}(n)$$

How does it relate to symmetric-key IND-CPA?

Recall: for symmetric ciphers we had two security notions:

- One-time security and many-time security (CPA)
- We showed that one-time security does not imply many-time security

For public key encryption:

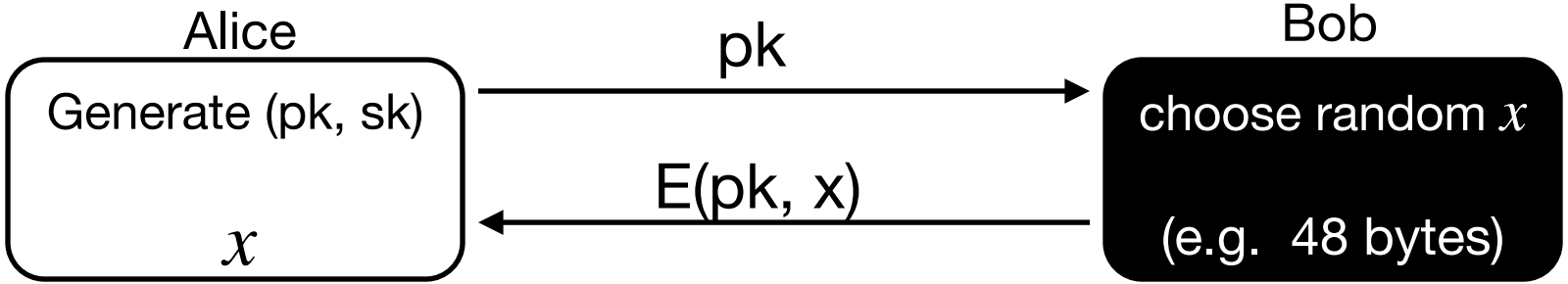
- One-time security $\Rightarrow$ many-time security (CPA)

  (follows from the fact that attacker can encrypt by themselves)

- Public key encryption **must** be randomized
  - Q: why not stateful?

Applications

Session setup (for now, only eavesdropping security)

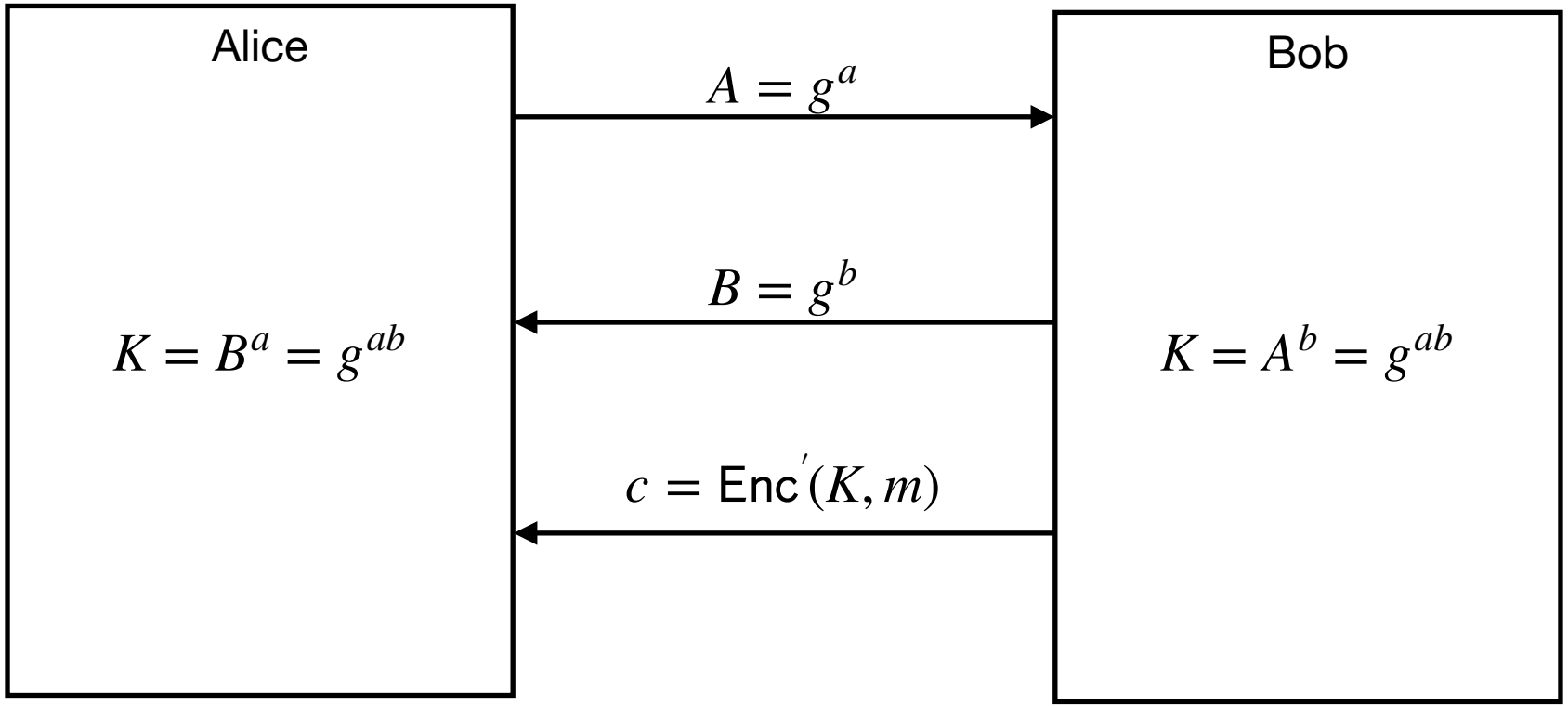


Non-interactive applications: (e.g. Email)

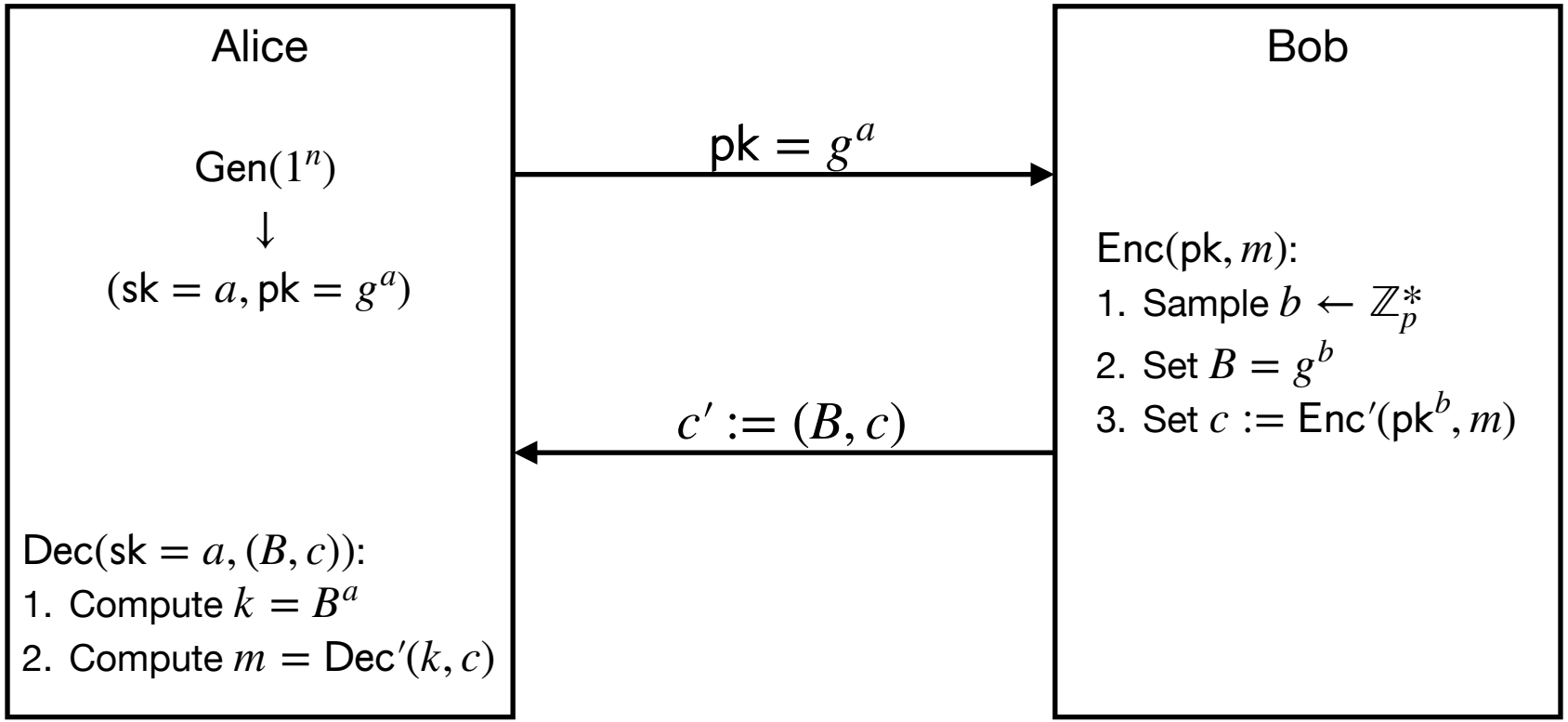
- Bob sends email to Alice encrypted using pk_{alice}
- Note: Bob needs pk_{alice} (public key management)

Constructions of PKE

Recall: DH Key Exchange



Convert DH → PKE



The Elgamal system (an abstract view)

- $\mathbb{G}$: finite cyclic group of prime order p with generator g
- $(\mathsf{Enc}', \mathsf{Dec}')$: symmetric-key encryption with keyspace $\mathscr{K} = \mathbb{G}$

Gen(1^n):
1. Sample $a \leftarrow \mathbb{Z}_p^*$
2. Output $(\mathsf{sk} = a, \mathsf{pk} = g^a)$

Enc(pk, m):
1. Sample $b \leftarrow \mathbb{Z}_p^*$
2. Set $B = g^b$
3. Set $c := \mathsf{Enc}'(\mathsf{pk}^b, m)$
4. Output $c' = (B, c)$

Dec(sk $= a$, (B, c)):
1. Compute $k = B^a$
2. Output $m = \mathsf{Dec}'(k, c)$

What choice of $(\mathsf{Enc}', \mathsf{Dec}')$?

How to prove security?

Q1: Choice of $(\mathsf{Enc}', \mathsf{Dec}')$: OTP?

- $\mathbb{G}$: finite cyclic group of prime order p with generator g
- Key idea: One-Time Pad works not just with $\{0,1\}^n$ and XOR, but with *any group*
  - $\mathsf{Gen}'(1^n)$: Sample $r \leftarrow \mathbb{Z}_p$, and output g^r
  - $\mathsf{Enc}'(k = g^r, m \in \mathbb{G})$: Output $c = k \cdot m \in \mathbb{G}$
  - $\mathsf{Dec}'(k = g^r, c \in \mathbb{G})$: Output $m = k^{-1} \cdot c \in \mathbb{G}$

Correctness: $\mathsf{Dec}'(k, \mathsf{Enc}'(k, m)) = k \cdot m \cdot k^{-1} = m$

Security: Goal: $\forall m, m' \in \mathbb{G}, c \in \mathbb{G}$,

$$\Pr_{k \leftarrow \mathbb{G}}[\mathsf{Enc}(k, m) = c] = \Pr_{k \leftarrow \mathbb{G}}[\mathsf{Enc}(k, m') = c]$$

Exercise: prove this (try to adapt proof from Lecture 1)

The Elgamal system (a concrete view)

- $\mathbb{G}$: finite cyclic group of prime order p with generator g
- $(\mathsf{Enc}', \mathsf{Dec}')$: symmetric-key encryption with keyspace $\mathscr{K} = \mathbb{G}$

Gen(1^n):

1. Sample $a \leftarrow \mathbb{Z}_p^*$
2. Output $(\mathsf{sk} = a, \mathsf{pk} = g^a)$

Enc(pk, m):

1. Sample $b \leftarrow \mathbb{Z}_p^*$
2. Set $B = g^b$
3. Set $c := \mathsf{Enc}'(\mathsf{pk}^b, m)$
4. Output $c' = (B, c)$

Dec(sk $= a$, (B, c)):

1. Compute $k = B^a$
2. Output $m = \mathsf{Dec}'(k, c)$

What choice of $(\mathsf{Enc}', \mathsf{Dec}')$?

How to prove security?

The Elgamal system (a concrete view)

- $\mathbb{G}$: finite cyclic group of prime order p with generator g
- $(\mathsf{Enc}', \mathsf{Dec}')$: symmetric-key encryption with keyspace $\mathcal{K} = \mathbb{G}$

Gen(1^n):

1. Sample $a \leftarrow \mathbb{Z}_p^*$
2. Output $(\mathsf{sk} = a, \mathsf{pk} = g^a)$

Enc(pk, m):

1. Sample $b \leftarrow \mathbb{Z}_p^*$
2. Set $B = g^b$
3. Set $c := m \cdot \mathsf{pk}^b = mg^{ab}$
4. Output $c' = (B, c)$

Dec(sk $= a$, (B, c)):

1. Compute $k = B^a$
2. Output $m = k^{-1}c$
   $= cg^{-ab}$
   $= mg^{ab}g^{-ab}$

What choice of $(\mathsf{Enc}', \mathsf{Dec}')$?

How to prove security?

Problem:
OTP uses random group element

But we only have g^{ab}!

Is this a problem? Isn't g^{ab} also random?

Problem: adversary *also* sees g^a and g^b!

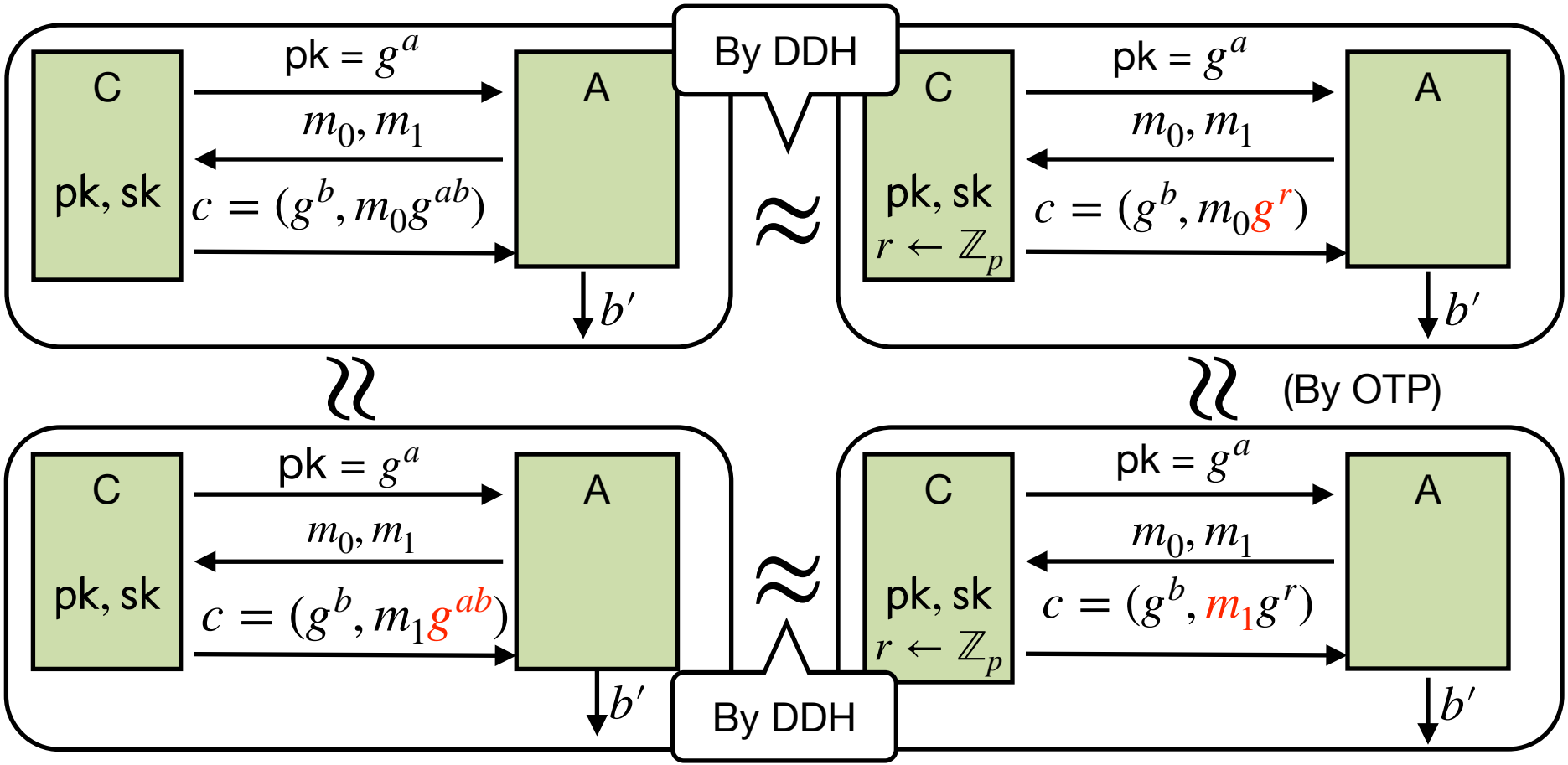
New assumption: Decisional Diffie—Hellman

Roughly, (g^a, g^b, g^{ab}) is indistinguishable from (g^a, g^b, g^r)

Formally, the following two distributions are computationally indistinguishable:

$\{(g^a, g^b, g^{ab})\}_{a,b \leftarrow \mathbb{Z}_p}$ and $\{(g^a, g^b, g^r)\}_{a,b,r \leftarrow \mathbb{Z}_p}$

Elgamal is semantically secure under DDH



The Elgamal system (a modern view)

- $\mathbb{G}$: finite cyclic group of prime order p with generator g
- $(\mathsf{Enc}', \mathsf{Dec}')$: what about arbitrary keyspace $\mathscr{K}$?
- New ingredient: "Random"-ish hash function $H : \mathbb{G} \to \mathscr{K}$

Gen(1^n):
1. Sample $a \leftarrow \mathbb{Z}_p^*$
2. Output $(\mathsf{sk} = a, \mathsf{pk} = g^a)$

Enc(pk, m):
1. Sample $b \leftarrow \mathbb{Z}_p^*$
2. Set $k := H(g^{ab})$
3. Set $c \leftarrow \mathsf{Enc}(k, m)$
4. Output $c' = (g^b, c)$

Dec(sk $= a$, (B, c)):
1. Compute $k = H(B^a)$
2. Output $m = \mathsf{Dec}'(k, c)$

New assumption: Hash-DDH

Roughly, $(g^a, g^b, H(g^{ab}))$ is indistinguishable from (g^a, g^b, R)

Formally, the following two distributions are computationally indistinguishable:

$\{(g^a, g^b, H(g^{ab}))\}_{a,b \leftarrow \mathbb{Z}_p}$ and $\{(g^a, g^b, R)\}_{a,b \leftarrow \mathbb{Z}_p, R \leftarrow \mathcal{K}}$

Q: If DDH is hard, is H-DDH hard?

Q: If H-DDH is hard, is DDH hard?

Suppose $K = \{0,1\}^{128}$ and

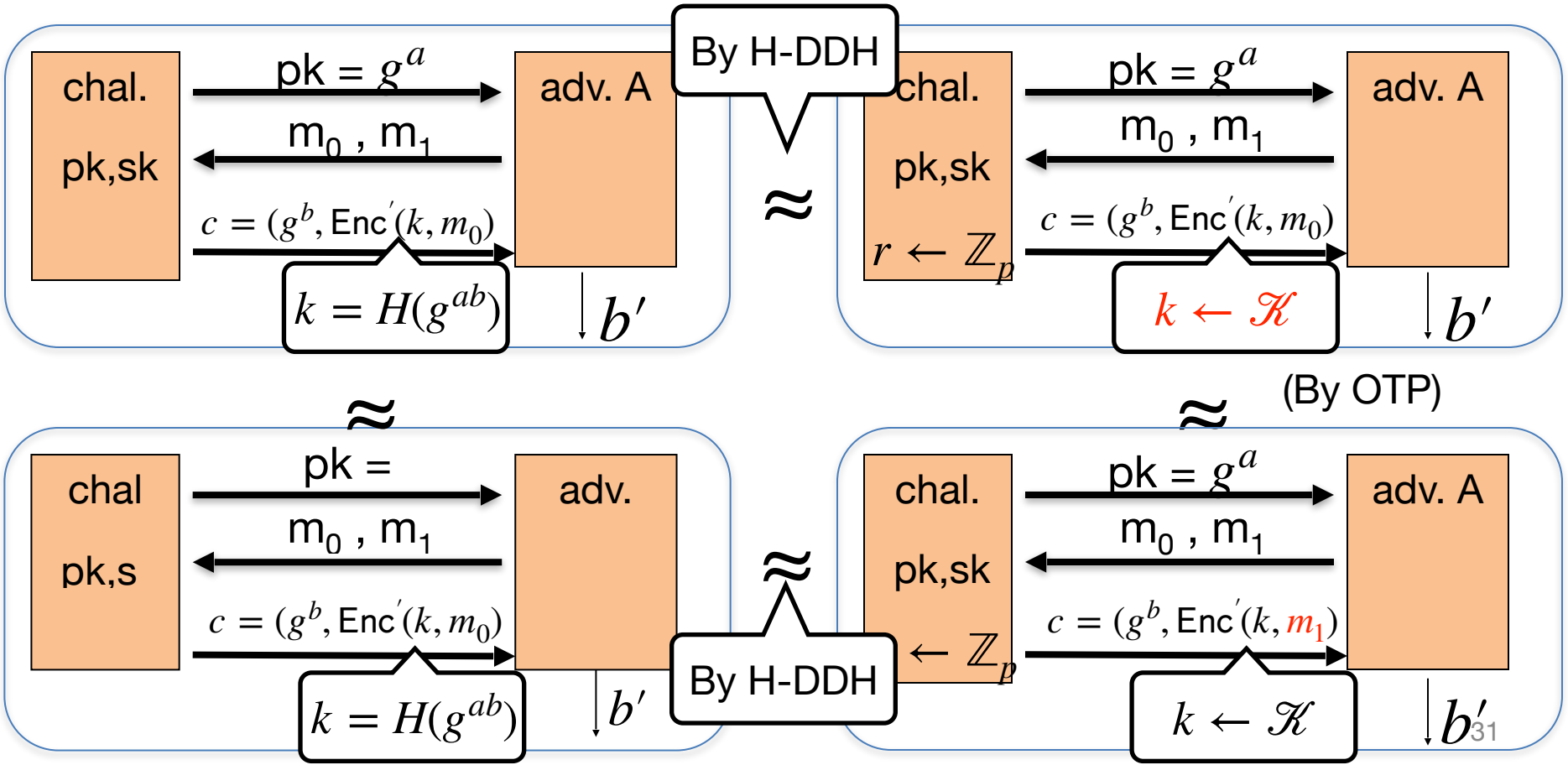
$H: G \rightarrow K$ only outputs strings in K that begin with 0

(i.e. for all y: msb(H(y))=0)

Can Hash-DH hold for (G, H) ?

- Yes, for some groups G
- No, Hash-DH is easy to break in this case
- Yes, Hash-DH is always true for such H

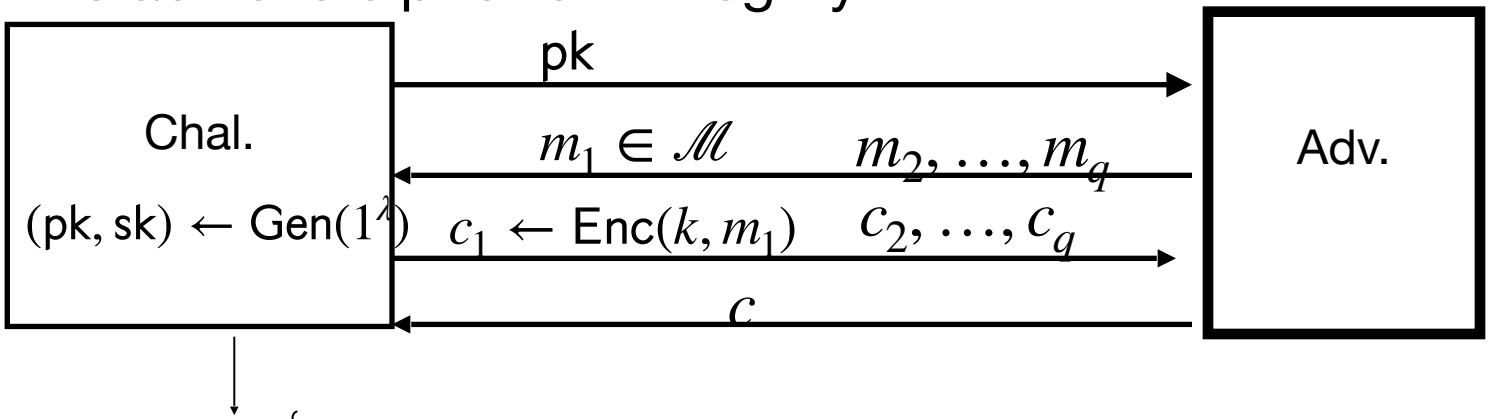
Elgamal is semantically secure under H-DDH



What about active attacks?

What about security against active attacks?

Can we achieve ciphertext integrity?



Def: (Gen, Enc, Dec) has **ciphertext integrity** if for all PPT A:

$$\mathsf{Adv}_{\mathsf{CI}}[A] = \Pr[b = 1] = \mathsf{negl}(\lambda)$$

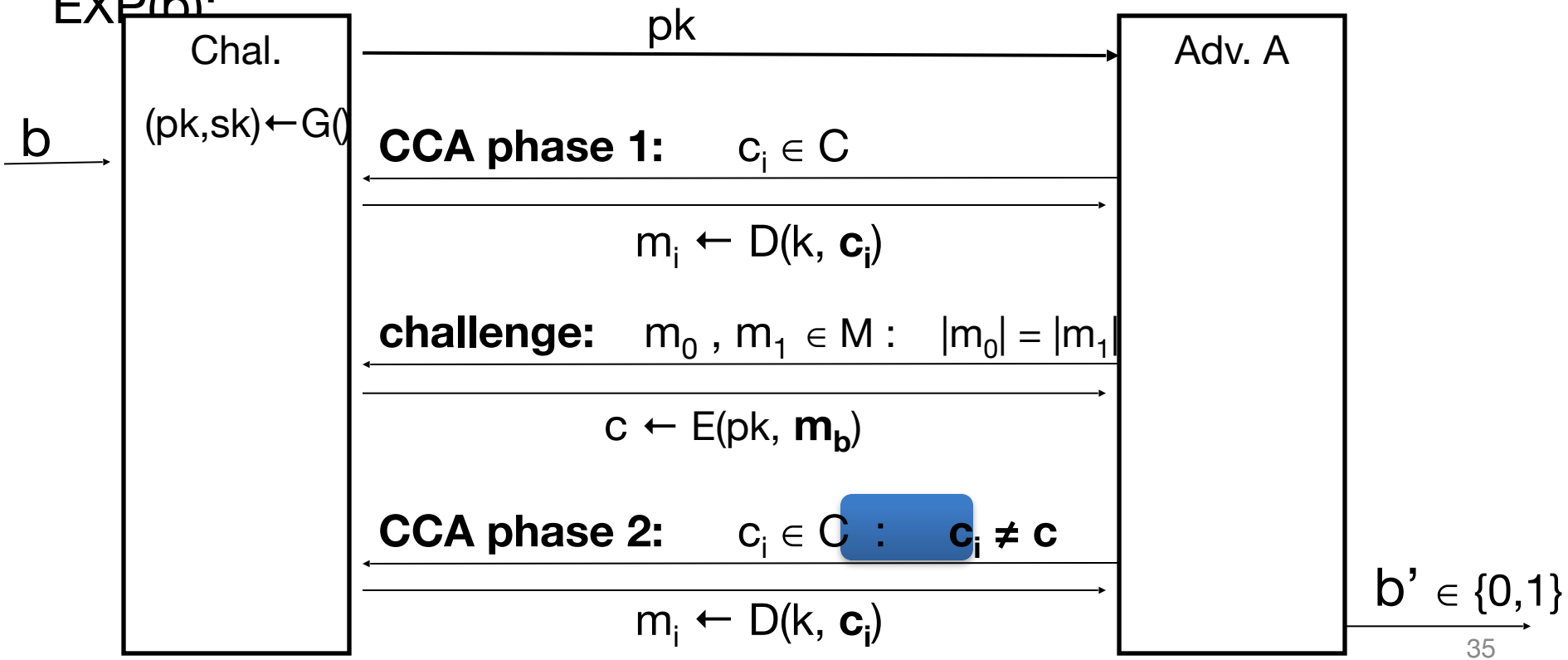
Problem

In public-key settings:

- Attacker **can** *always* create new ciphertexts using pk !!
- So instead: we directly require chosen ciphertext security

(pub-key) Chosen Ciphertext Security: definition

E = (G,E,D) public-key enc. over (M,C). For b=0,1 define EXP(b):



Chosen ciphertext security: definition

Def: E is CCA secure (a.k.a IND-CCA) if for all efficient A:

$$\mathrm{Adv}_{CCA}[A,E] = \left|\Pr[EXP(0)=1] - \Pr[EXP(1)=1]\right| \text{ is negligible.}$$

b

ElGamal chosen ciphertext security?

Security Theorem:

If **IDH** holds in the group G, **(E_s, D_s)** provides auth. enc.

and **H:** $G^2 \longrightarrow K$ is a "random oracle"

then **ElGamal** is CCA^{ro} secure.

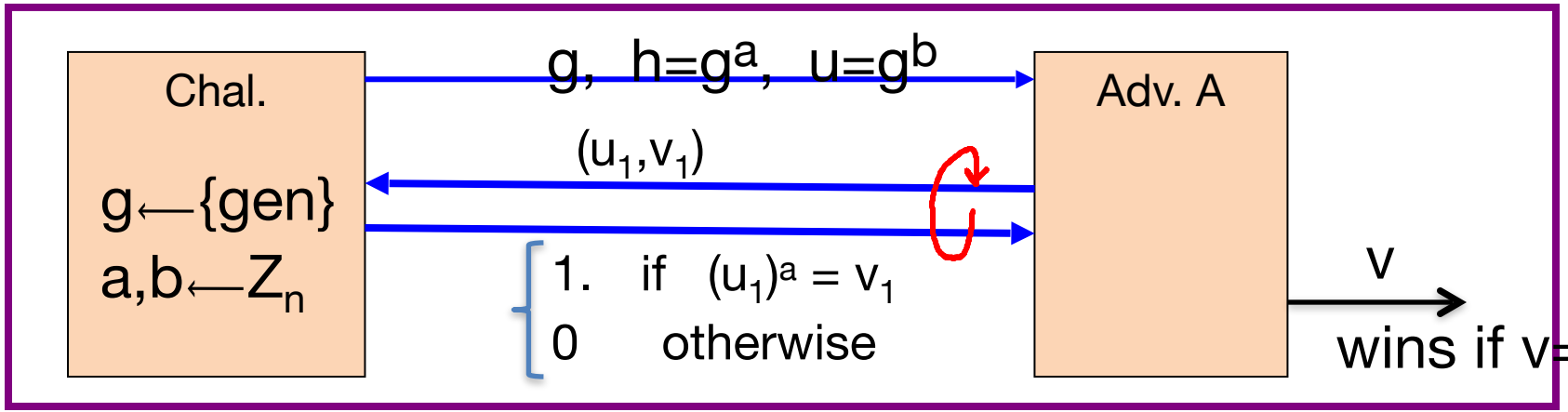
Questions: (1) can we prove CCA security based on CDH?

(2) can we prove CCA security without random oracles?

ElGamal chosen ciphertext security?

To prove chosen ciphertext security need stronger assumption

Interactive Diffie-Hellman (IDH) in group G:



IDH holds in G if: **∀efficient A: Pr[A outputs g^{ab}] < negligible**

Decisional Diffie-Hellman Assumption

Decisional Diffie-Hellman Assumption (DDHA):

Hard to distinguish between g^{xy} and a uniformly random group element, given g, g^x and g^y

That is, the following two distributions are computationally indistinguishable:

$$(g, g^x, g^y, g^{xy}) \approx (g, g^x, g^y, u)$$

DH/El Gamal is IND-secure under the DDH assumption on the given group.