CIS 5560

Cryptography
Lecture 15

Course website:
pratyushmishra.com/classes/cis-5560-s24/

Slides adapted from Dan Boneh and Vinod Vaikuntanathan

Recap of Last Lecture(s)

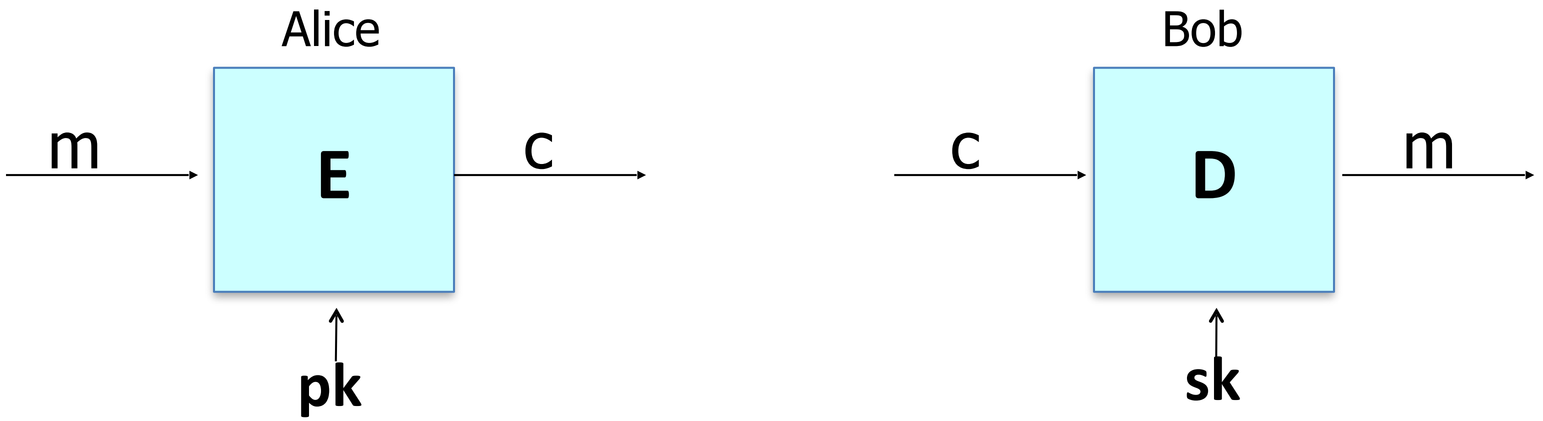
- Number Theory refresher
  - Arithmetic modulo primes
  - Fermat's Little Theorem
  - Cyclic groups
  - Discrete Logarithms
- Key Exchange
  - Merkle puzzles
  - Diffie—Hellman
    - Computational Diffie—Hellman Problem

Today's Lecture

- Public Key Encryption
  - El Gamal Encryption
    - Computational Diffie—Hellman Problem
  - RSA Encryption
    - Arithmetic modulo composites
    - Factoring

Public key encryption

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



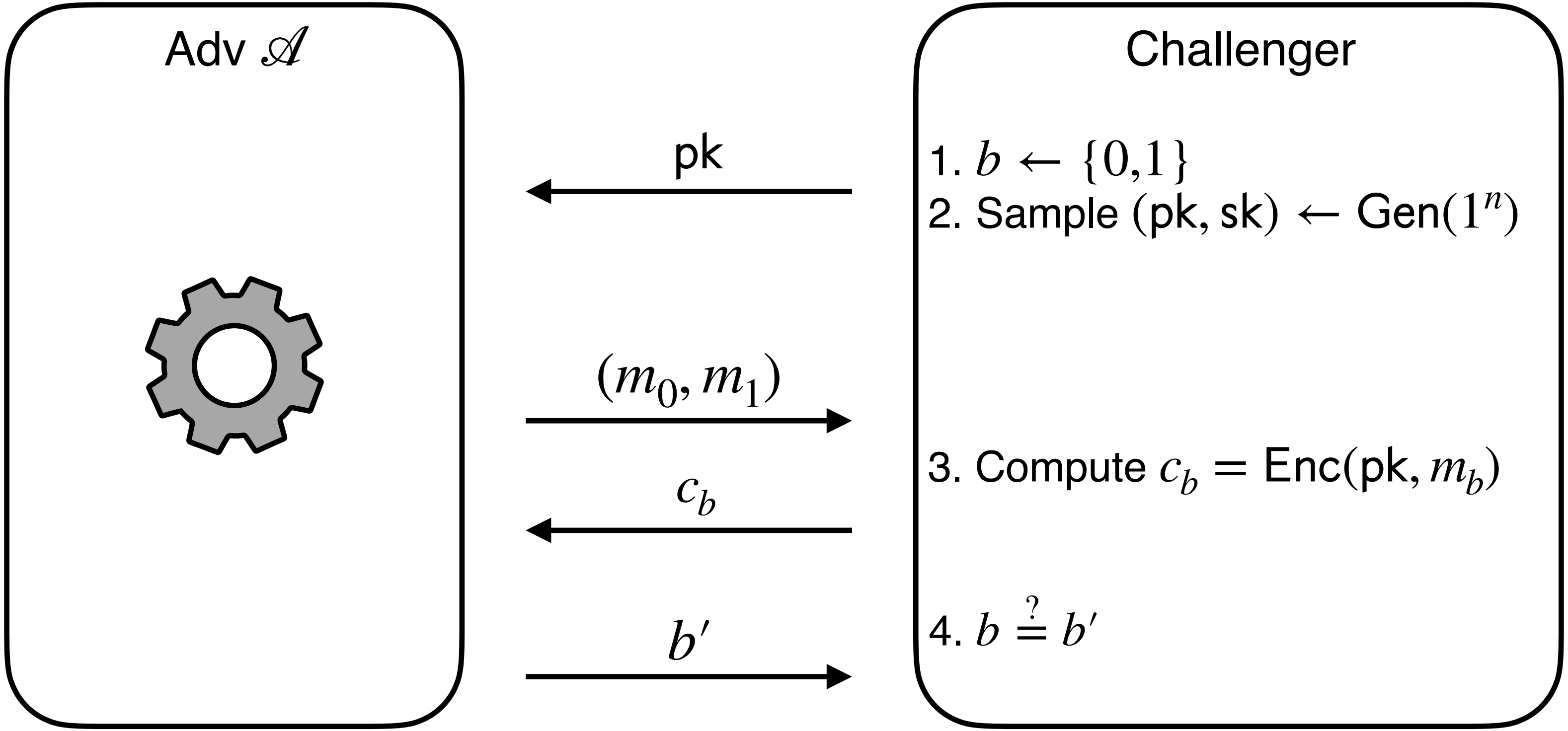
Public key encryption

<u>Def</u>: a public-key encryption system is a triple of algs. (G, E, D)

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

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

Security: IND-CPA for PKE



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

Security: IND-CPA for PKE

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

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

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:

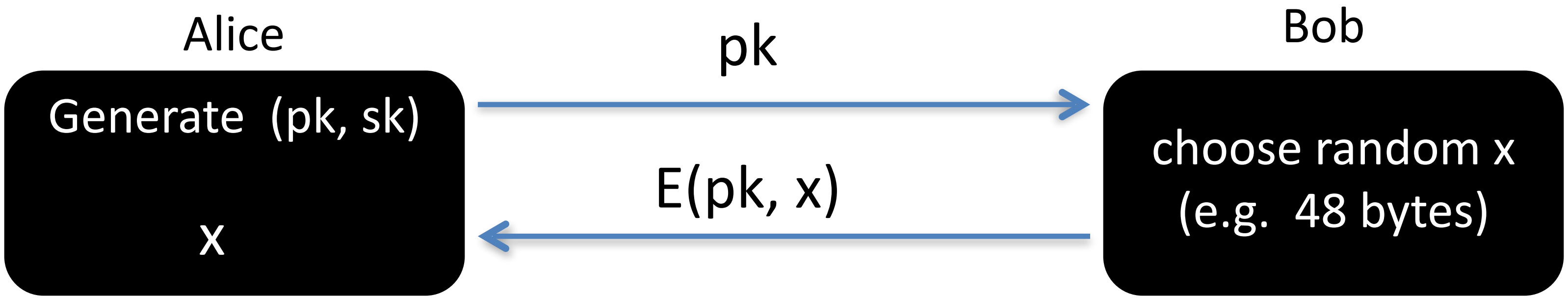
- One-time security $\Rightarrow$ many-time security (CPA)

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

- Public key encryption **must** be randomized

Applications

Session setup (for now, only eavesdropping security)



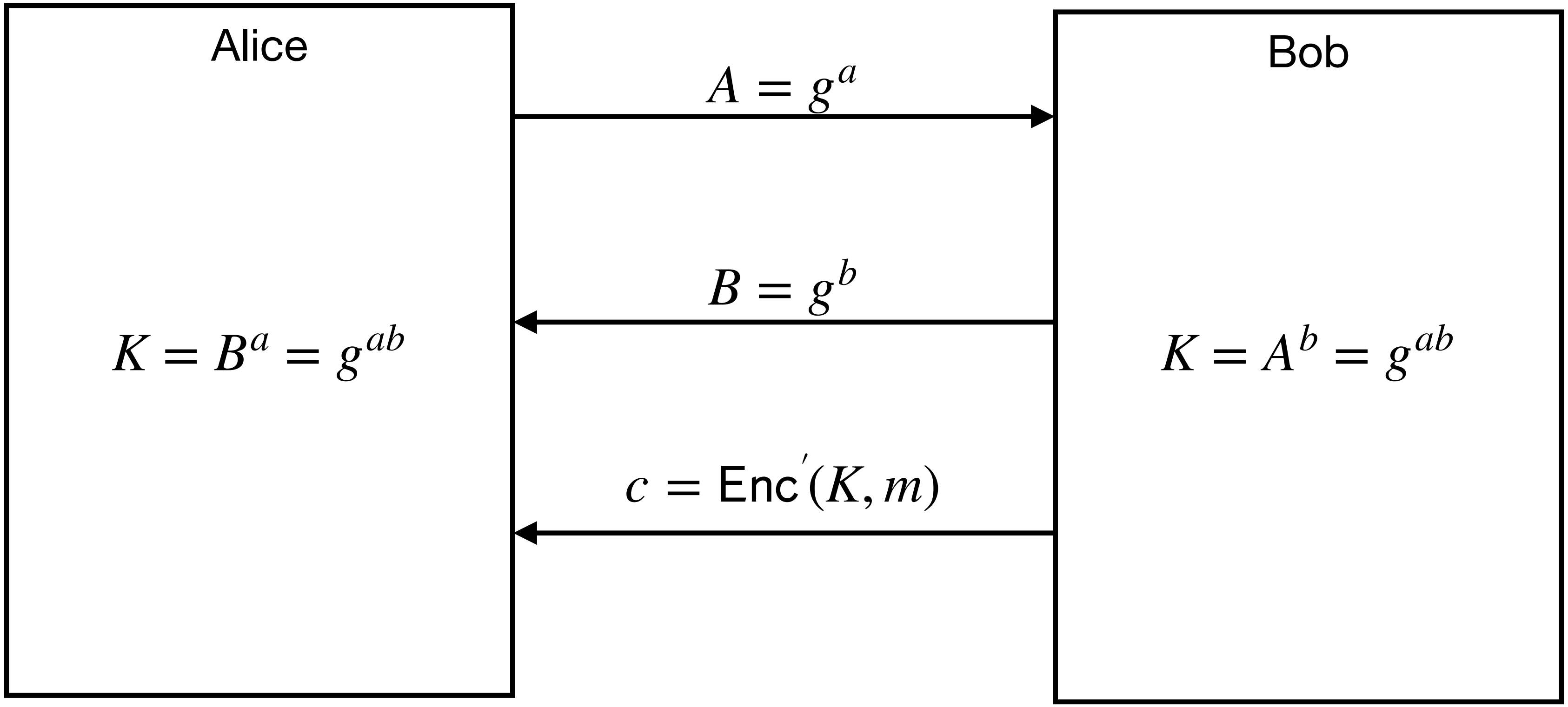
Non-interactive applications: (e.g. Email)

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

Constructions of PKE:
Elgamal Encryption

Review of cyclic groups
(On board)

Recall: DH Key Exchange



Convert DH → PKE

Alice

$\mathsf{Gen}(1^n)$

$\downarrow$

$(\mathsf{sk} = a, \mathsf{pk} = g^a)$

$\xrightarrow{\mathsf{pk} = g^a}$

Bob

$\mathsf{Enc}(\mathsf{pk}, m)$:

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

$\xleftarrow{c' := (B, c)}$

$\mathsf{Dec}(\mathsf{sk} = a, (B, c))$:

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

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 $\mathscr{K} = \mathbb{G}$

<u>Gen(1^n):</u>

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

<u>Enc(pk, m):</u>

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

<u>Dec(sk $= a, (B, c)$):</u>

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 $(\text{Enc}^{'}, \text{Dec}^{'})$: 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*
  - $\text{Gen}^{'}(1^n)$: Sample $r \leftarrow \mathbb{Z}_p$, and output g^r
  - $\text{Enc}^{'}(k = g^r, m \in \mathbb{G})$: Output $c = k \cdot m \in \mathbb{G}$
  - $\text{Dec}^{'}(k = g^r, c \in \mathbb{G})$: Output $m = k^{-1} \cdot c \in \mathbb{G}$

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

Security: Goal: $\forall m, m' \in \mathbb{G}, c \in \mathbb{G}, \Pr_{k \leftarrow \mathbb{G}}[\text{Enc}(k, m) = c] = \Pr_{k \leftarrow \mathbb{G}}[\text{Enc}(k, m') = c]$

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 $\mathcal{K} = \mathbb{G}$

<u>Gen(1^n):</u>

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

<u>Enc(pk, m):</u>

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

<u>Dec(sk $= a, (B, c)$):</u>

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 $\mathcal{K} = \mathbb{G}$

$\underline{\mathsf{Gen}(1^n)}$:	$\underline{\mathsf{Enc}(\mathsf{pk}, m)}$:	$\underline{\mathsf{Dec}(\mathsf{sk} = a, (B, c))}$:
1. Sample $a \leftarrow \mathbb{Z}_p^*$	1. Sample $b \leftarrow \mathbb{Z}_p^*$	1. Compute $k = B^a$
2. Output $(\mathsf{sk} = a, \mathsf{pk} = g^a)$	2. Set $B = g^b$	2. Output $m = k^{-1}c$
	3. Set $c := m \cdot \mathsf{pk}^b = mg^{ab}$	$= cg^{-ab}$
	4. Output $c' = (B, c)$	$= mg^{ab}g^{-ab}$

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!

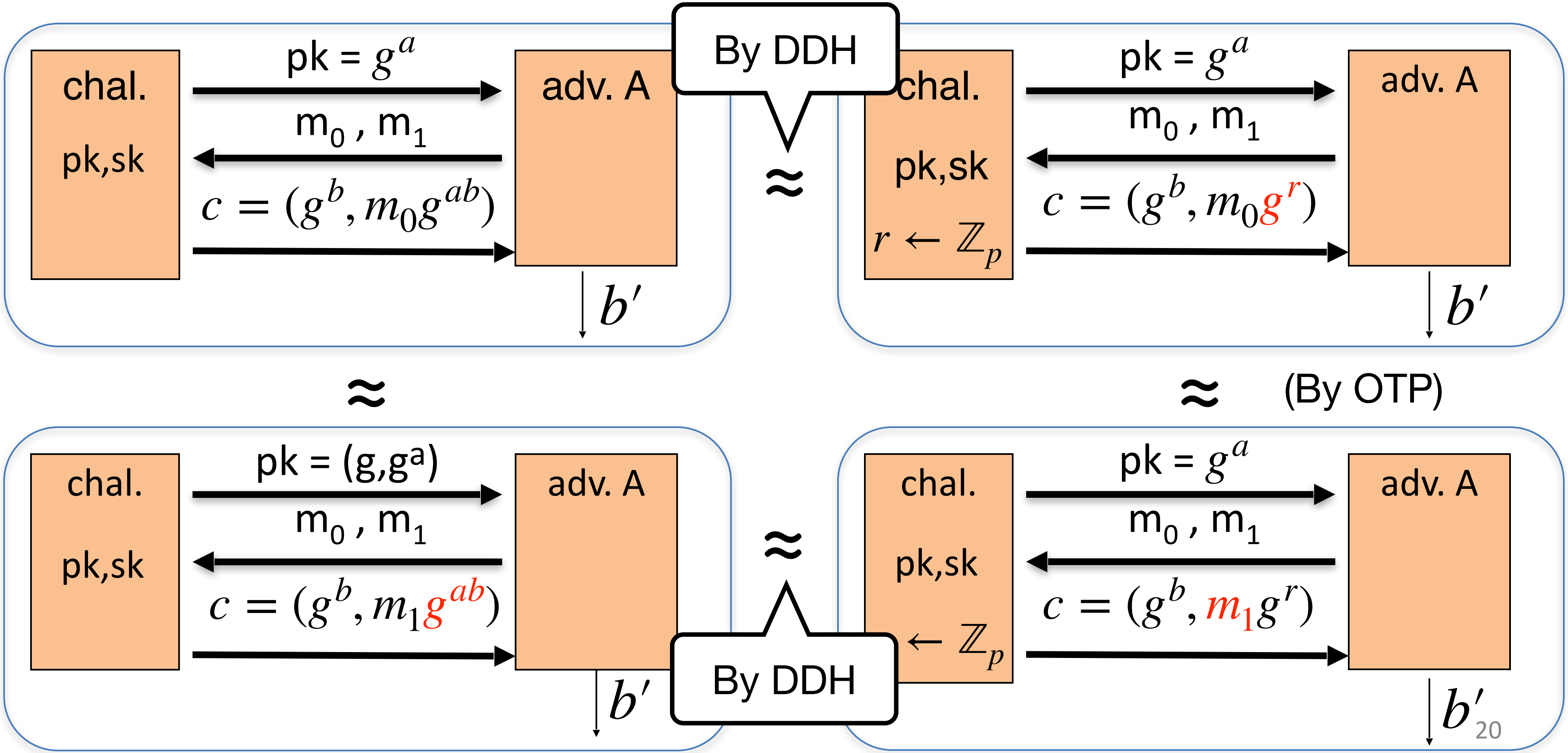
New assumption: Decisional Diffie—Hellman

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 $H : \mathbb{G} \to \mathscr{K}$

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

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

<u>Dec(sk $= a, (B, c)$):</u>
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:

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

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

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

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

$H: G \longrightarrow K$ 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