COMS BC3262: Introduction to Cryptography

Lecture 17: El-Gamal and Factoring

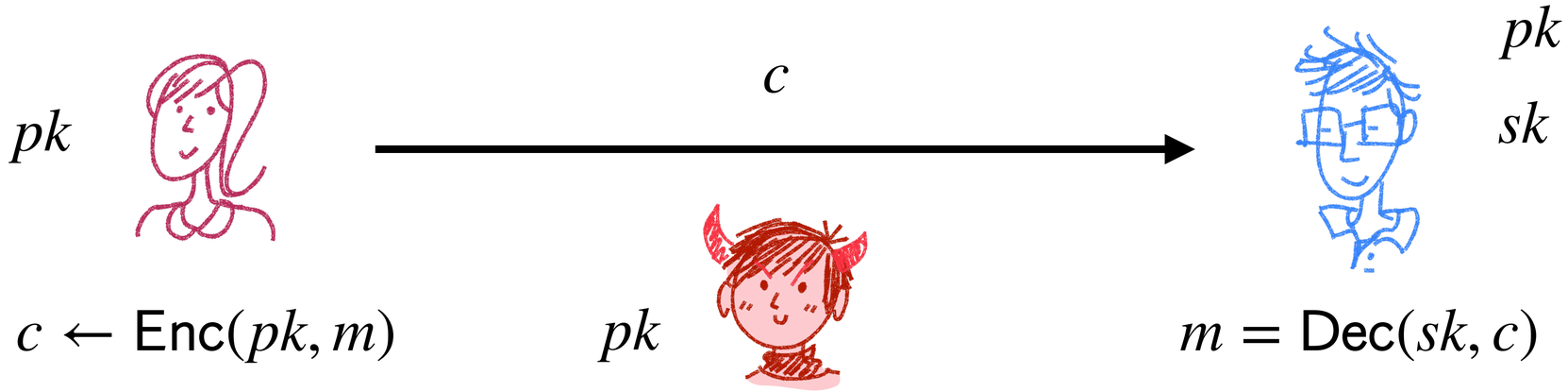
April 1, 2026

Logistics

- Friday is the last day for the alternate oral exam
  - See EdStem for the link to sign up
  - After Friday, midterm grades will be set!
- **Extra credit opportunity**: Attend Olive's talk Thursday, April 16 at 12pm in Milstein 402
  - Olive Franzese-McLaughlin will be giving a talk titled "Cryptographically Verified AI/ML Audits"
- PS 4 is out today and is due in ~2 weeks

Key Exchange vs PKE

Public-Key Encryption (Informal)



1. What is the difference between this and private-key encryption?
2. How did we define security for this setting?

Key Exchange (Informal)

Correctness: At the end of the protocol Alice and Bob derive the same key

Security: An eavesdropper who sees messages exchanged and receives $\tilde{k}$ can't tell if $\tilde{k}$ is a random group element or the derived key k

Key Exchange vs PKE

In general, a public key encryption scheme gives a key exchange protocol

How?

Key Exchange vs PKE

In general, a public key encryption scheme gives a key exchange protocol

How?

$(pk, sk) \leftarrow \mathsf{Gen}(1^n)$

Key Exchange vs PKE

In general, a public key encryption scheme gives a key exchange protocol

How?

$(pk, sk) \leftarrow \mathsf{Gen}(1^n)$

pk

Key Exchange vs PKE

In general, a public key encryption scheme gives a key exchange protocol

How?

$(pk, sk) \leftarrow \mathsf{Gen}(1^n)$

pk

$k \leftarrow \{0,1\}^n$

$c \leftarrow \mathsf{Enc}(pk, k)$

Key Exchange vs PKE

In general, a public key encryption scheme gives a key exchange protocol

How?

$(pk, sk) \leftarrow \mathsf{Gen}(1^n)$

pk

$k \leftarrow \{0,1\}^n$

$c \leftarrow \mathsf{Enc}(pk, k)$

c

Key Exchange vs PKE

In general, a public key encryption scheme gives a key exchange protocol

How?

$(pk, sk) \leftarrow \mathsf{Gen}(1^n)$

pk

$k \leftarrow \{0,1\}^n$

$c \leftarrow \mathsf{Enc}(pk, k)$

c

$k = \mathsf{Dec}(sk, c)$

Key Exchange vs PKE

In general, a public key encryption scheme gives a key exchange protocol

How?

$(pk, sk) \leftarrow \mathsf{Gen}(1^n)$

pk

$k \leftarrow \{0,1\}^n$

$c \leftarrow \mathsf{Enc}(pk, k)$

c

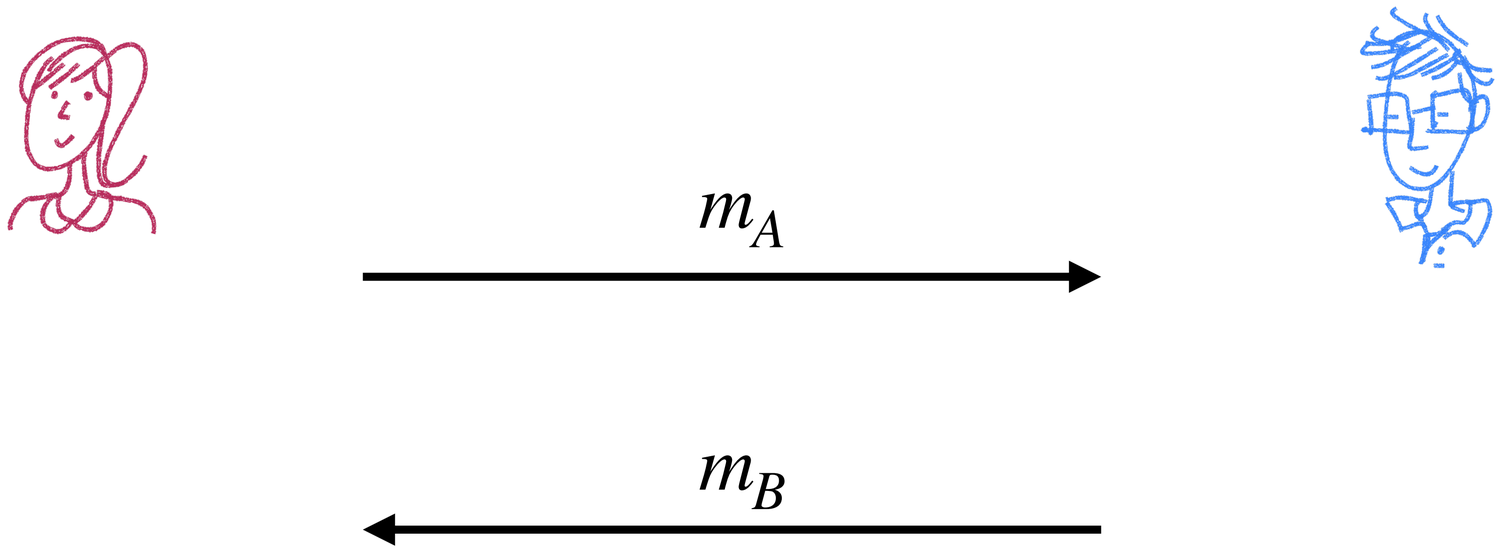
$k = \mathsf{Dec}(sk, c)$

What about the other direction?

Can we build a public key encryption scheme from a key exchange protocol?

Key Exchange vs PKE

Using a 2-round key exchange, we can build a PKE scheme



Key Exchange vs PKE

Using a 2-round key exchange, we can build a PKE scheme

$pk = m_A$

m_B

Alice sets pk as m_A and the sk as the randomness used to generate m_A

Key Exchange vs PKE

Using a 2-round key exchange, we can build a PKE scheme

$pk = m_A$

Alice sets pk as m_A and the sk as the randomness used to generate m_A

m_B

Compute k from m_A, m_B

Use k to hide the message (e.g., OTP). Call this c_2

Key Exchange vs PKE

Using a 2-round key exchange, we can build a PKE scheme

$pk = m_A$

Alice sets pk as m_A and the sk as the randomness used to generate m_A

(m_B, c_2)

Compute k from m_A, m_B

Use k to hide the message (e.g., OTP). Call this c_2

Key Exchange vs PKE

Using a 2-round key exchange, we can build a PKE scheme

$pk = m_A$

Compute k from m_A, m_B

(m_B, c_2)

Derive k from m_B

Reconstruct message from c_2 and k

Use k to hide the message (e.g., OTP). Call this c_2

El-Gamal Encryption Example

El-Gamal Encryption

Let $\mathcal{G}$ be a PPT algorithm that on input 1^n outputs $(\mathbb{G}, q, g)$ where $\mathbb{G}$ is a cyclic group of order q that is generated by g, and q is an n-bit prime

- Gen(1^n): Sample $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ and $x \leftarrow \mathbb{Z}_q$. Set $h = g^x$.

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

- Enc(pk, m): Sample $y \leftarrow \mathbb{Z}_q$ and output $(g^y, h^y \cdot m)$

- Dec$(sk, (c_1, c_2))$: Output $m = c_2 / c_1^x$

El-Gamal Encryption: Example

- Let $q = 5$ and $p = 2q + 1 = 11$
- Let $\mathbb{G}$ be the order-q subgroup of quadratic residues ($\mathbb{G} = \{1,4,9,5,3\}$)
- Let $g = [4 \mod 11]$ (any element other than 1 is a generator)
- Let the secret key be $x = 3 \in \mathbb{Z}_5$
  - Then $g^x = 4^3 = 9 \mod 11$
- Public key is $(\mathbb{G}, q, g, h) = (11,5,4,9)$

El-Gamal Encryption: Example

- Let $q = 5$ and $p = 2q + 1 = 11$
- Let $\mathbb{G}$ be the order-q subgroup of quadratic residues ($\mathbb{G} = \{1,4,9,5,3\}$)
- Let $g = [4 \mod 11]$ (any element other than 1 is a generator)
- Let the secret key be $x = 3 \in \mathbb{Z}_5$. Then $g^x = 4^3 = 9 \mod 11$
- Public key is $(\mathbb{G}, q, g, h) = (11,5,4,9)$

- How could we encrypt $m = 5$?

  - Let's say we sample randomness $y = 2 \in \mathbb{Z}_5$.

  - Then ciphertext is $c_1 = g^y = 4^2 = 5 \mod 11$ and
    $c_2 = h^y \cdot m = 9^2 \cdot 5 = 4 \cdot 5 = 9 \mod 11$ (i.e., $(c_1, c_2) = (5,9)$)

Security of El-Gamal Encryption

El-Gamal Encryption

Let $\mathcal{G}$ be a PPT algorithm that on input 1^n outputs $(\mathbb{G}, q, g)$ where $\mathbb{G}$ is a cyclic group of order q that is generated by g, and q is an n-bit prime

- Gen(1^n): Sample $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ and $x \leftarrow \mathbb{Z}_q$. Set $h = g^x$.

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

- Enc(pk, m): Sample $y \leftarrow \mathbb{Z}_q$ and output $(g^y, h^y \cdot m)$

- Dec$(sk, (c_1, c_2))$: Output $m = c_2 / c_1^x$

Correctness: $\text{Dec}\left(sk, \text{Enc}(pk, m)\right) = \text{Dec}\left(sk, (g^y, h^y \cdot m)\right) = \dfrac{h^y \cdot m}{(g^y)^x} = \dfrac{(g^x)^y \cdot m}{(g^y)^x}$

Security of El-Gamal Encryption

Theorem: Under the DDH assumption, El-Gamal encryption is CPA-secure

Proof idea:

- DDH assumption says g^{xy} is pseudorandom in $\mathbb{G}$ given (g, g^x, g^y)
- If (g, g^x, g^y, g^{xy}) is computationally indistinguishable from (g, g^x, g^y, g^z), then so is $(g, g^x, g^y, g^{xy} \cdot m)$ from $(g, g^x, g^y, g^z \cdot m)$
- $g^z \cdot m$ is uniformly distributed and independent of m (by the useful lemma we proved a few slides ago)

Security of El-Gamal Encryption

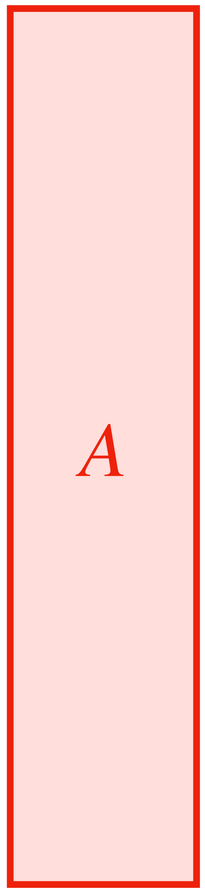
Theorem: Under the DDH assumption, El-Gamal encryption is CPA-secure

Proof:

- Let A be a PPT adversary against the CPA-security of the enc scheme

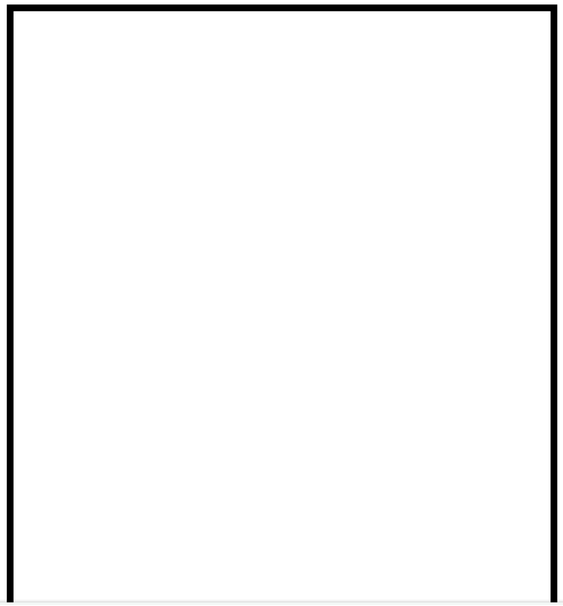
- We will construct a distinguisher D that breaks the DDH assumptions (distinguishes between (g, g^x, g^y, g^{xy}) and (g, g^x, g^y, g^z))

Security of El-Gamal Encryption



Distinguisher D

DDH Challenger

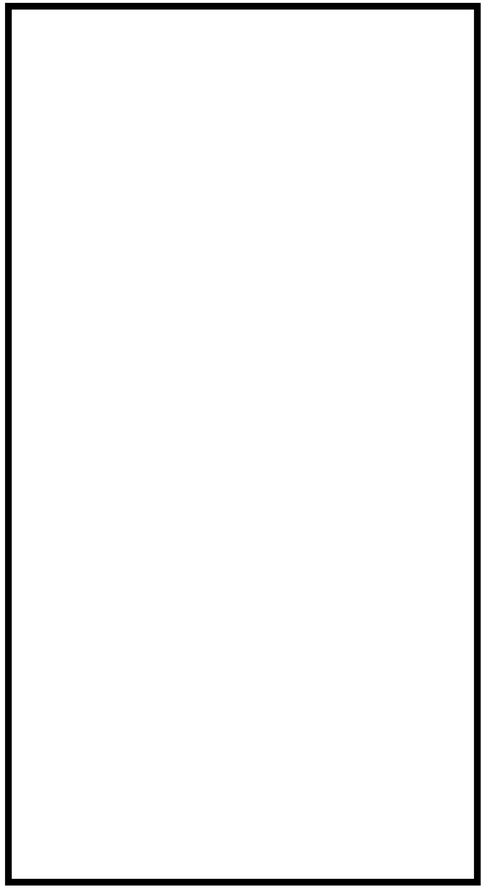


Gen(1^n): Sample $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ and $x \leftarrow \mathbb{Z}_q$. Set $h = g^x$.

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

Enc(pk, m): Sample $y \leftarrow \mathbb{Z}_q$ and output $(g^y, h^y \cdot m)$

Dec$(sk, (c_1, c_2))$: Output $m = c_2 / c_1^x$



Security of El-Gamal Encryption

A

Distinguisher D

DDH Challenger

$(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$
$x, y \leftarrow \mathbb{Z}_q$
$b \leftarrow \{0,1\}$
if $b = 0$:
  $h \leftarrow \mathbb{G}$
else :
  $h = g^{xy}$

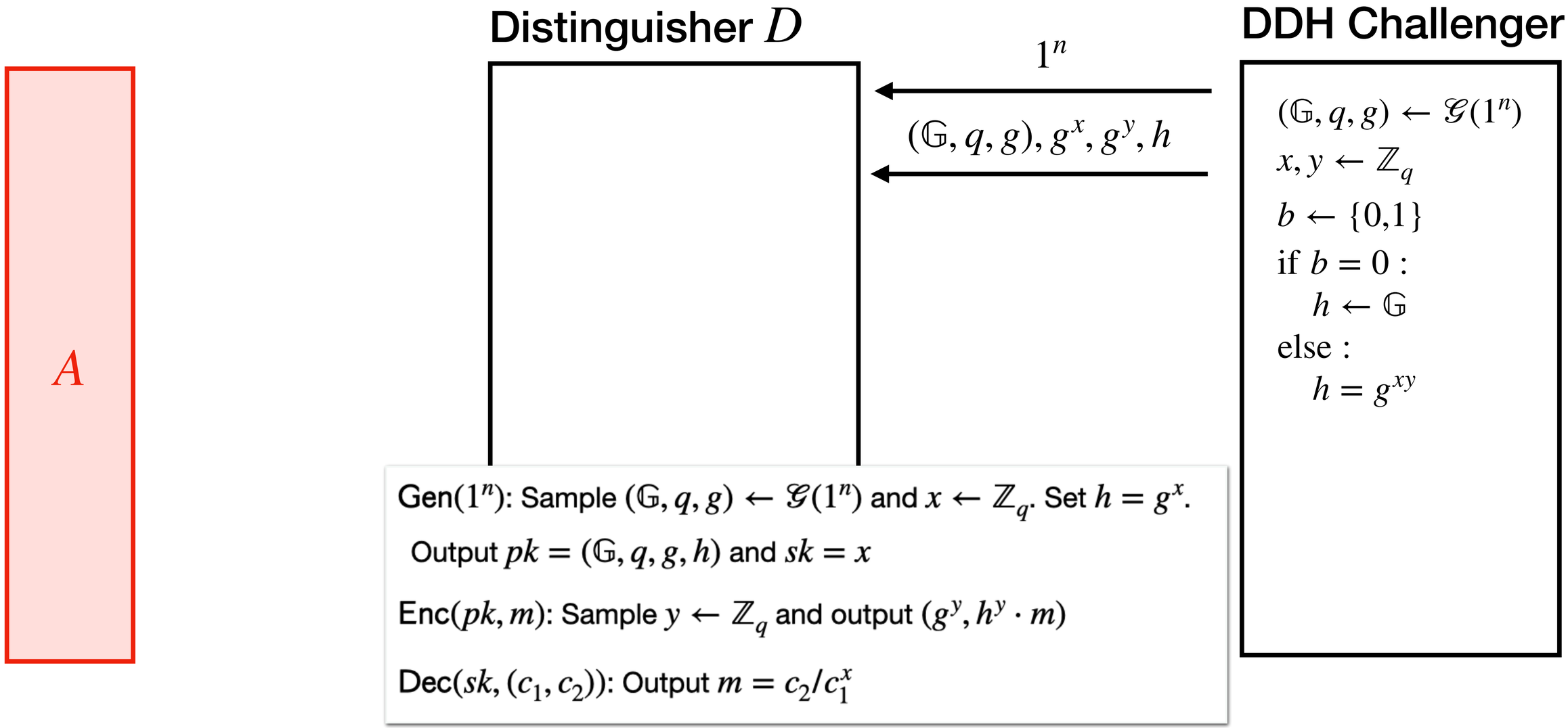
Gen(1^n): Sample $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ and $x \leftarrow \mathbb{Z}_q$. Set $h = g^x$.

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

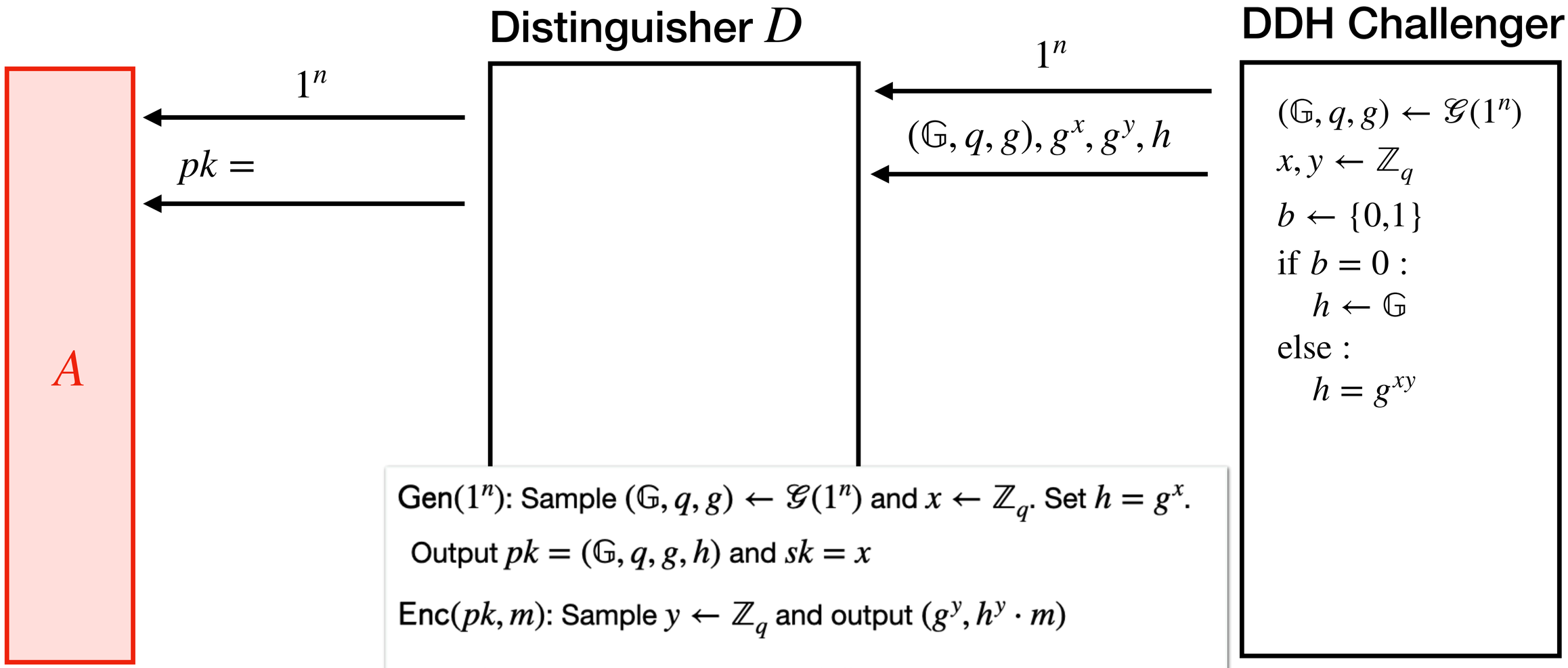
Enc(pk, m): Sample $y \leftarrow \mathbb{Z}_q$ and output $(g^y, h^y \cdot m)$

Dec$(sk, (c_1, c_2))$: Output $m = c_2 / c_1^x$

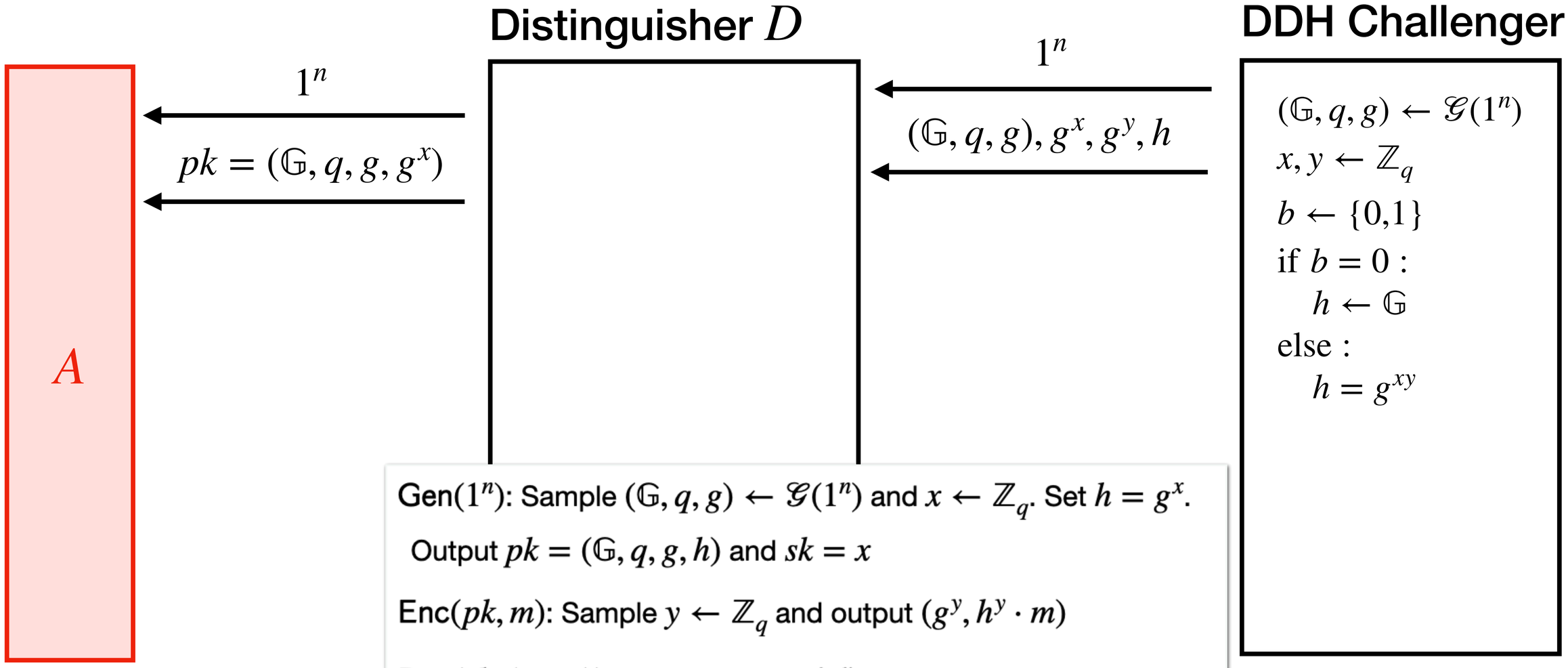
Security of El-Gamal Encryption



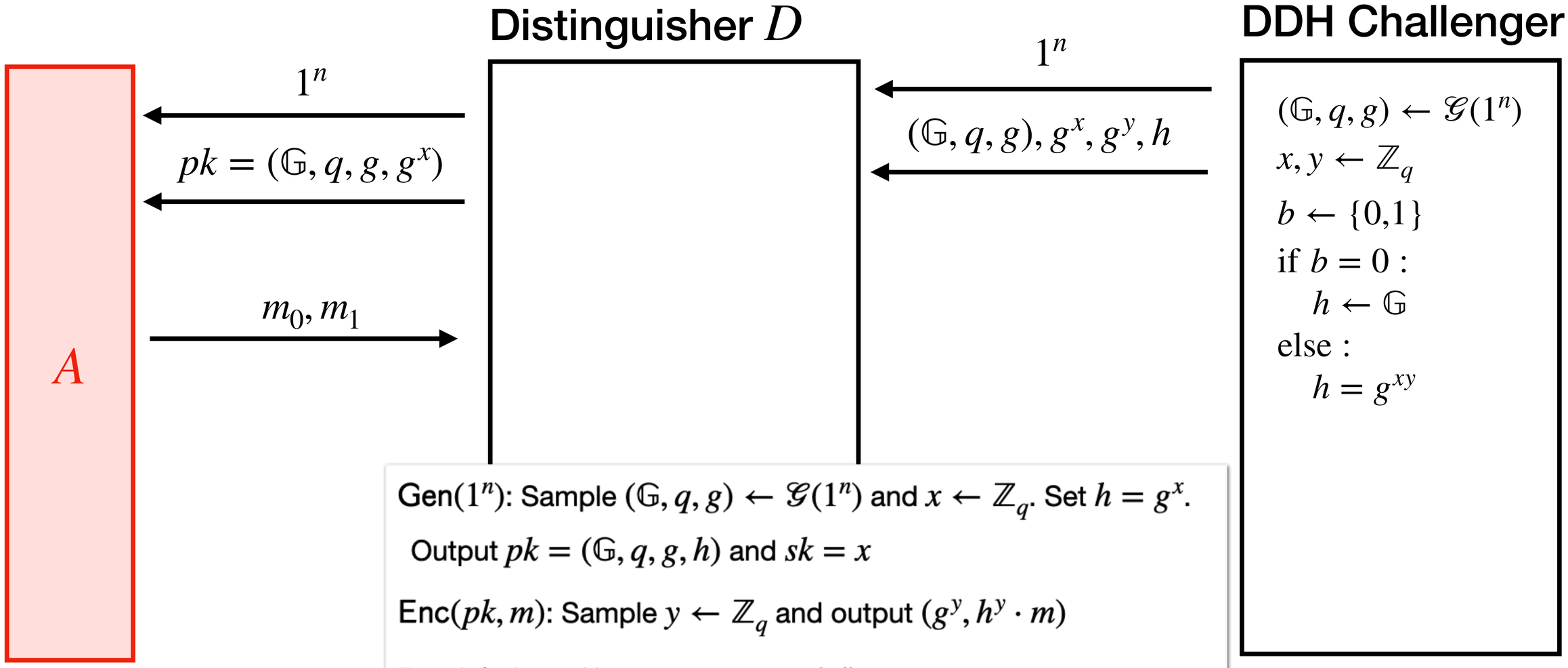
Security of El-Gamal Encryption



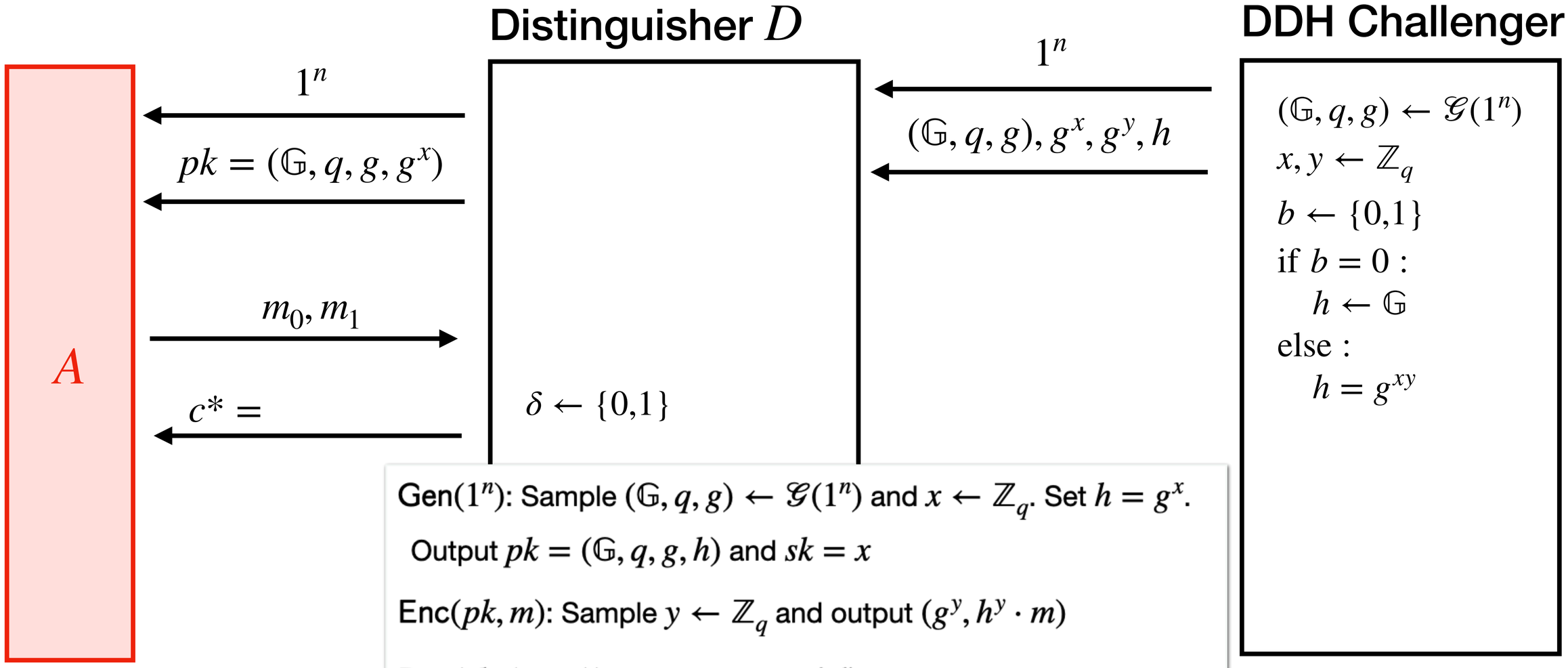
Security of El-Gamal Encryption



Security of El-Gamal Encryption



Security of El-Gamal Encryption



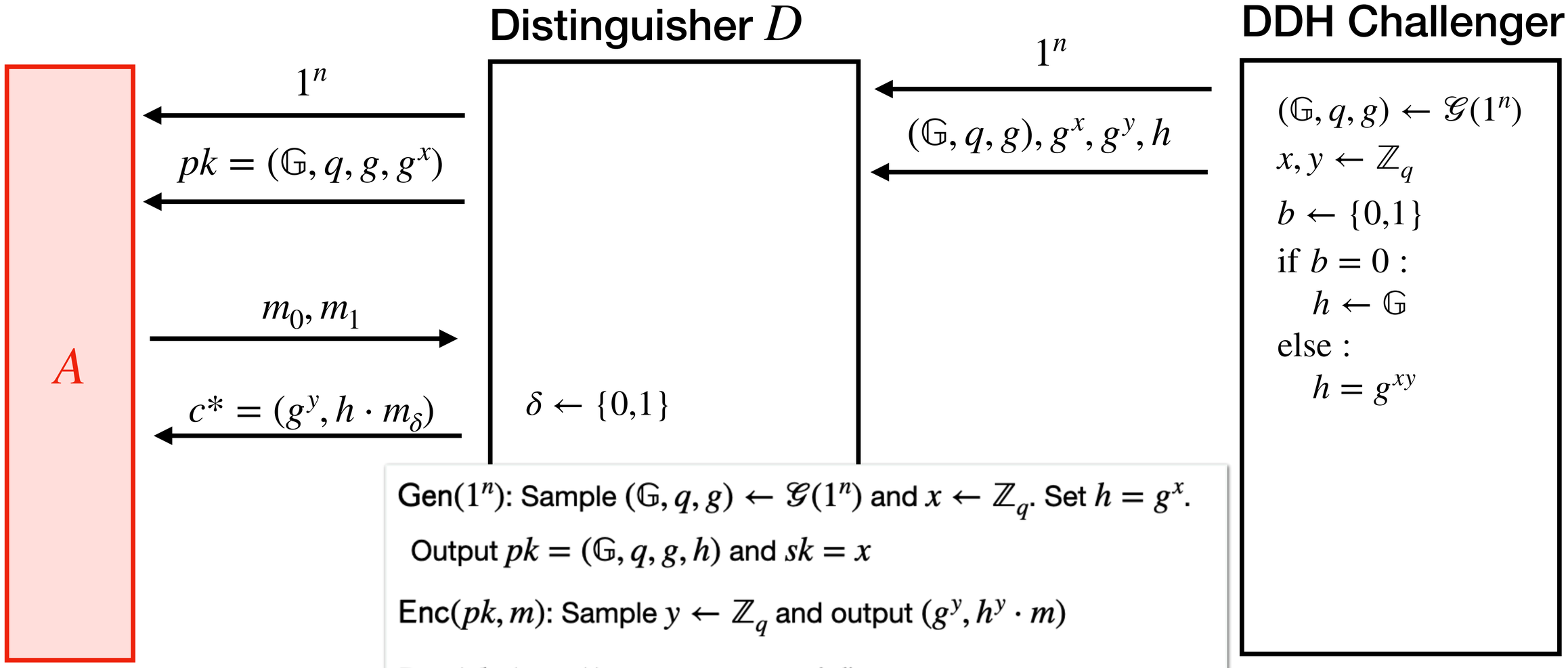
Gen(1^n): Sample $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ and $x \leftarrow \mathbb{Z}_q$. Set $h = g^x$.

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

Enc(pk, m): Sample $y \leftarrow \mathbb{Z}_q$ and output $(g^y, h^y \cdot m)$

Dec($sk, (c_1, c_2)$): Output $m = c_2 / c_1^x$

Security of El-Gamal Encryption



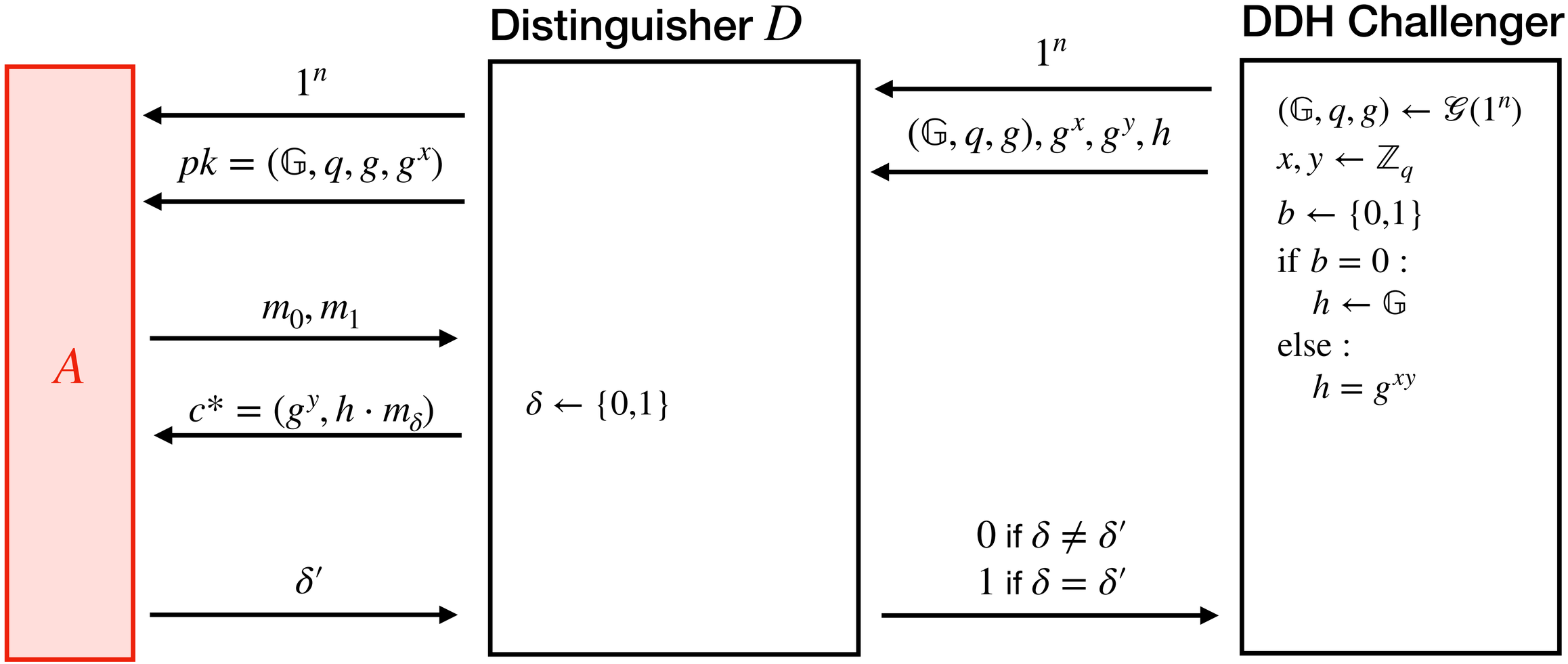
Gen(1^n): Sample $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ and $x \leftarrow \mathbb{Z}_q$. Set $h = g^x$.

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

Enc(pk, m): Sample $y \leftarrow \mathbb{Z}_q$ and output $(g^y, h^y \cdot m)$

Dec($sk, (c_1, c_2)$): Output $m = c_2 / c_1^x$

Security of El-Gamal Encryption

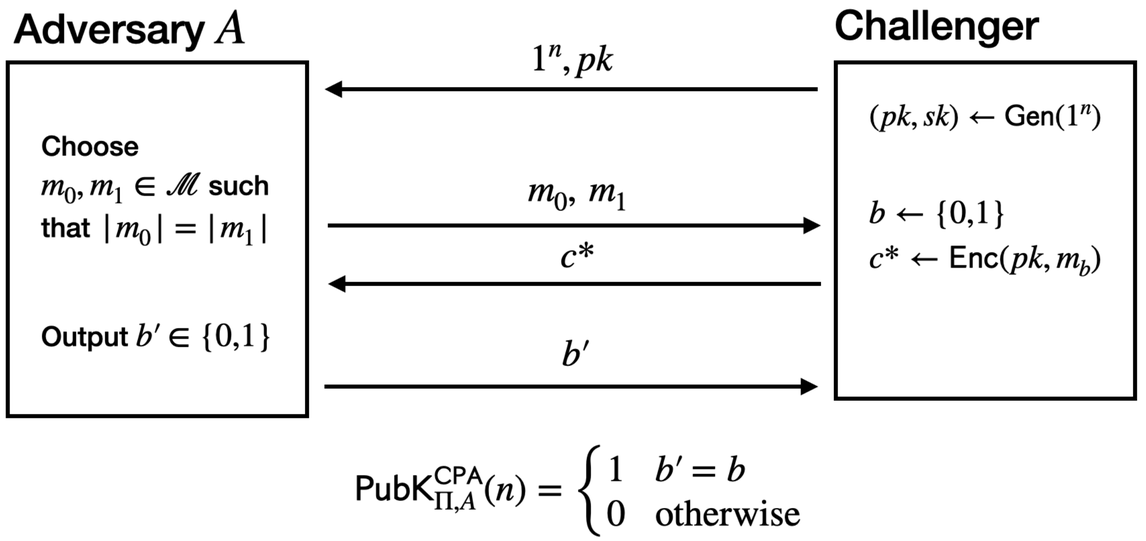


Recall: Chosen-Plaintext Attack (CPA)

Definition:

Π has indistinguishable encryptions under chosen-plaintext attack (or CPA-security) if for every PPT adversary A there exists a negligible function $\epsilon(\cdot)$ such that

$$\Pr[\text{PubK}^{\text{CPA}}_{\Pi,A}(n) = 1] \leq \frac{1}{2} + \epsilon(n)$$



$$\text{PubK}^{\text{CPA}}_{\Pi,A}(n) = \begin{cases} 1 & b' = b \\ 0 & \text{otherwise} \end{cases}$$

Notes:

- No encryption oracle in the public key setting
- Similar to the private-key setting, encryption must be **randomized**

Security of El-Gamal Encryption

Case 1: $(\mathbb{G}, q, g, g^x, g^y, g^{xy})$

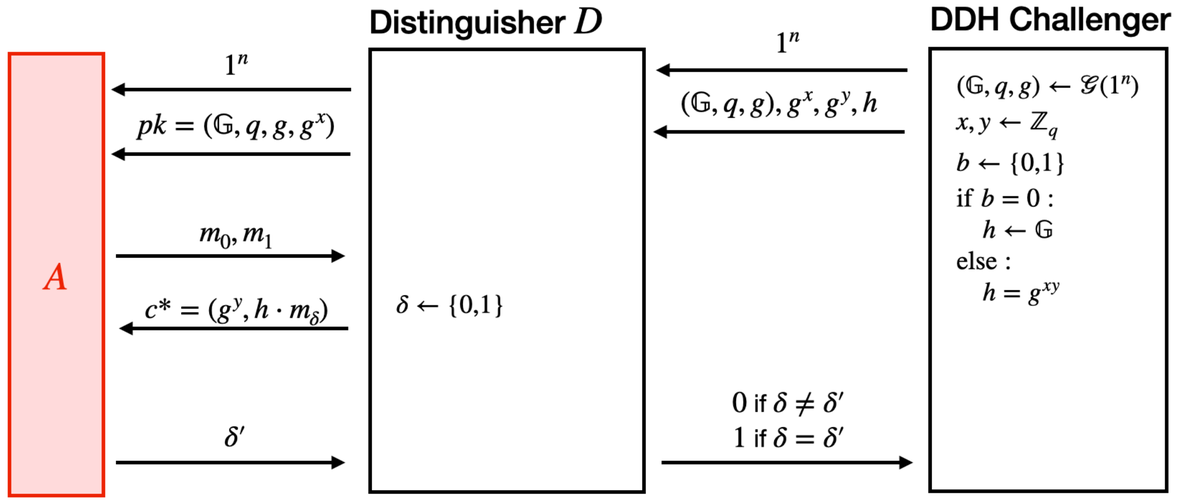
- View of A is identical to its view in the CPA-security experiment

- $\Pr[D(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1] = \Pr[\text{PubK}^{\text{CPA}}_{\Pi,A}(n) = 1]$

Case 2: $(\mathbb{G}, q, g, g^x, g^y, g^z)$

- By our useful lemma, we have A's view is independent of m

- $\Pr[D(\mathbb{G}, q, g, g^x, g^y, g^z) = 1] = 1/2$



Security of El-Gamal Encryption

Case 1 and Case 2 together we get

$$\left|\Pr\left[D(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1\right] - \Pr\left[D(\mathbb{G}, q, g, g^x, g^y, g^z) = 1\right]\right|$$

$$= \left|\Pr[\mathsf{PubK}^{\mathsf{CPA}}_{\Pi,A}(n) = 1] - \frac{1}{2}\right|$$

Security of El-Gamal Encryption

Case 1 and Case 2 together we get

$$\left|\Pr\left[D(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1\right] - \Pr\left[D(\mathbb{G}, q, g, g^x, g^y, g^z) = 1\right]\right|$$

$$= \left|\Pr[\mathsf{PubK}^{\mathsf{CPA}}_{\Pi,A}(n) = 1] - \frac{1}{2}\right|$$

DDH assumption states

$$\left|\Pr\left[D(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1\right] - \Pr\left[D(\mathbb{G}, q, g, g^x, g^y, g^z) = 1\right]\right| \leq \mathsf{negl}(n)$$

Security of El-Gamal Encryption

Case 1 and Case 2 together we get

$$\left|\Pr\left[D(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1\right] - \Pr\left[D(\mathbb{G}, q, g, g^x, g^y, g^z) = 1\right]\right|$$

$$= \left|\Pr[\mathsf{PubK}^{\mathsf{CPA}}_{\Pi,A}(n) = 1] - \frac{1}{2}\right|$$

DDH assumption states

$$\left|\Pr\left[D(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1\right] - \Pr\left[D(\mathbb{G}, q, g, g^x, g^y, g^z) = 1\right]\right| \leq \mathsf{negl}(n)$$

Therefore, $\Pr[\mathsf{PubK}^{\mathsf{CPA}}_{\Pi,A}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n)$

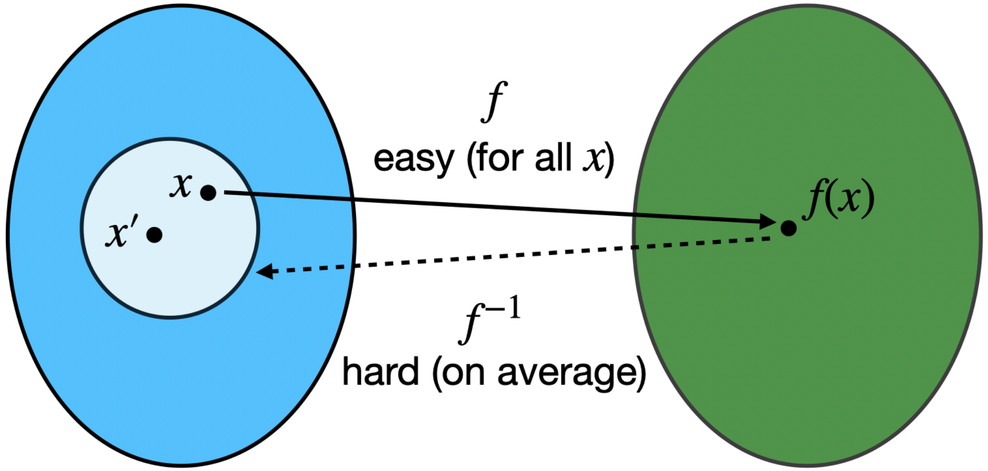
Factoring Assumption

Factoring Assumption

Informally, “multiplying two large primes is a OWF”:

Given a composite N, it’s hard to find p, q such that $pq = N$

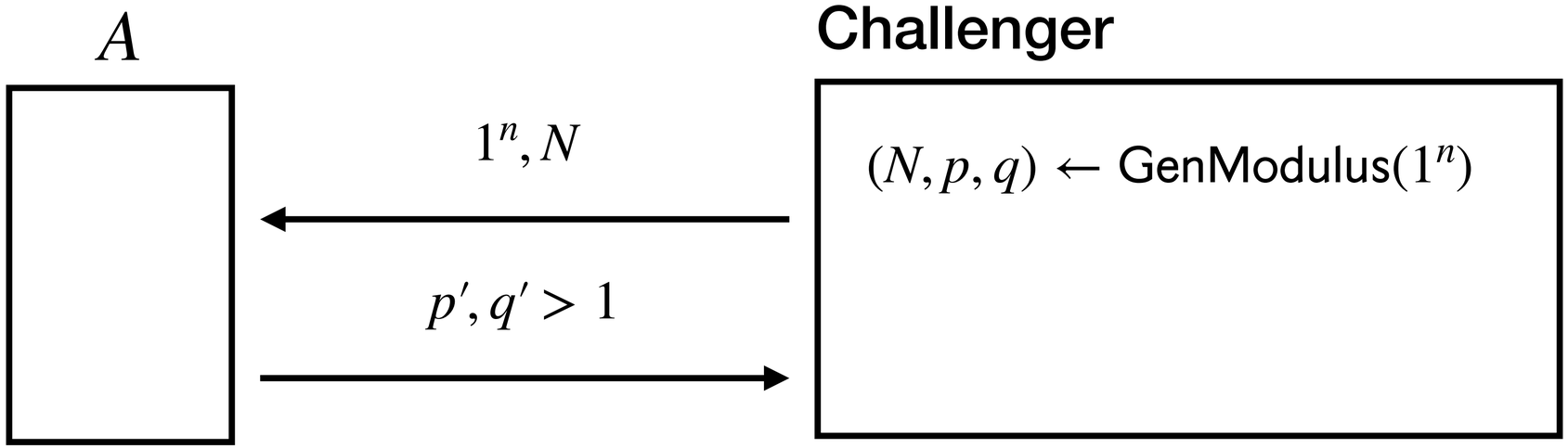
A function $f : \{0,1\}^* \to \{0,1\}^*$ that is easy to compute but hard to invert



See Lecture 8 for introduction of OWFs

Factoring Assumption

Suppose we had an algorithm GenModulus that on input 1^n outputs (N, p, q) such that $N = p \cdot q$ and p, q are n-bit primes



$$\text{Factor}_{A,\text{GenModulus}}(n) = \begin{cases} 1 & N = p'q' \\ 0 & \text{otherwise} \end{cases}$$

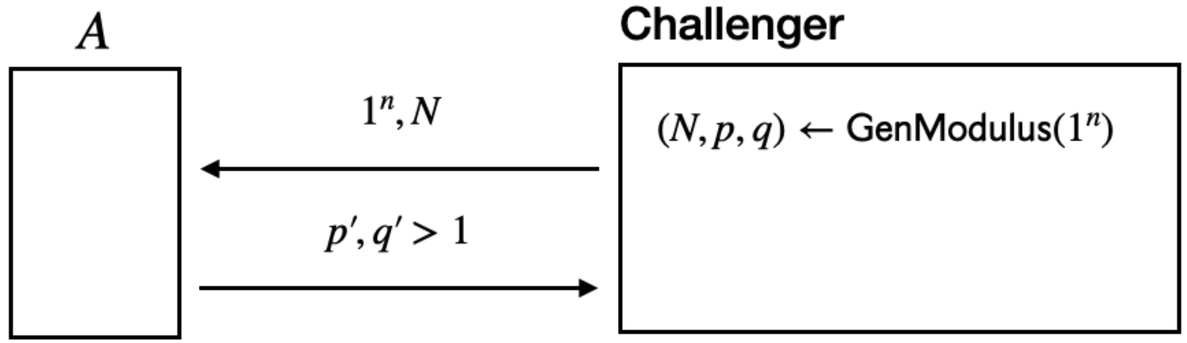
Factoring Assumption

Suppose we had an algorithm GenModulus that on input 1^n outputs (N, p, q) such that $N = p \cdot q$ and p, q are n-bit primes

Definition:

The Factoring is hard relative to GenModulus if for all PPT adversaries A there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\Pr[\mathsf{Factor}_{A,\mathsf{GenModulus}}(n) = 1] \leq \mathsf{negl}(n)$$



$$\mathsf{Factor}_{A,\mathsf{GenModulus}}(n) = \begin{cases} 1 & N = p'q' \\ 0 & \text{otherwise} \end{cases}$$

Factoring Assumption

Naive algorithm for finding the factors of N:

$$\begin{aligned}&\text{For } i = 2 \text{ to } \sqrt{N}\\&\quad \text{if } i \mid N\\&\quad\quad \text{return } i\\&\text{output } N \text{ is prime}\end{aligned}$$

How long does this take to run if N is an n-bit string?

$$O\left(\sqrt{N}\right) = O\left(\sqrt{2^n}\right) = O\left(2^{n/2}\right)$$

Factoring Assumption

- Do we have better algorithms for factoring?
  - There are better non-trivial algorithms, but we don’t have any polynomial-time algorithms in the classical setting
  - There are poly time quantum algorithms for factoring (and discrete log)
- There are poly time algorithms for determining if an input is prime or composite

Factoring Assumption

- The Factoring Assumption is a candidate OWF
- Can we use it to construct a public-key encryption scheme?
  - It’s more complicated than it looks!
  - Rabin constructed a public-key encryption scheme whose security is equivalent to factoring, but it’s not used in practice
  - We’ll talk more about this next time…