COMS BC3262: Introduction to Cryptography

Lecture 16: Key Exchange and PKE

March 30, 2026

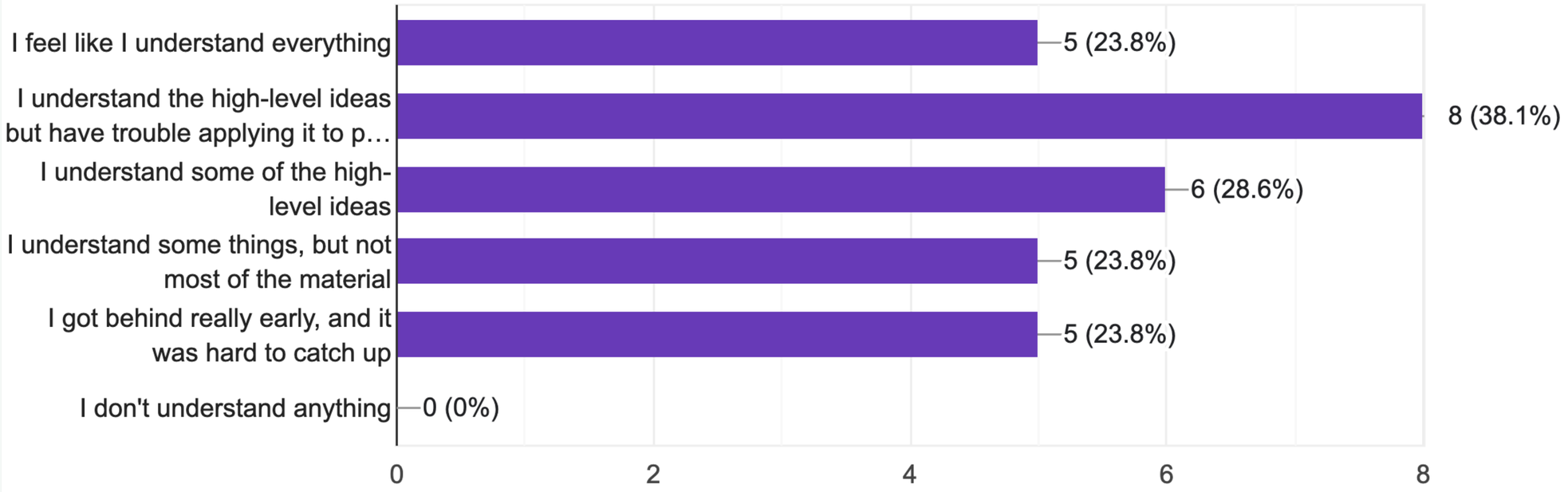
Before we start…

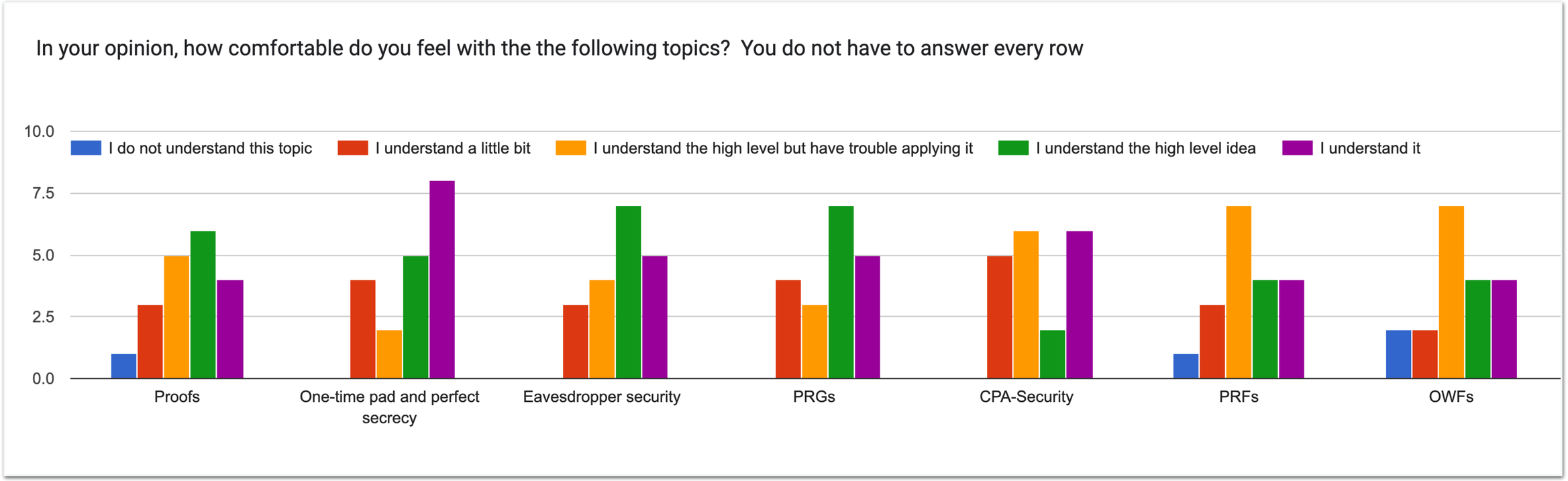
Mid-Semester Feedback

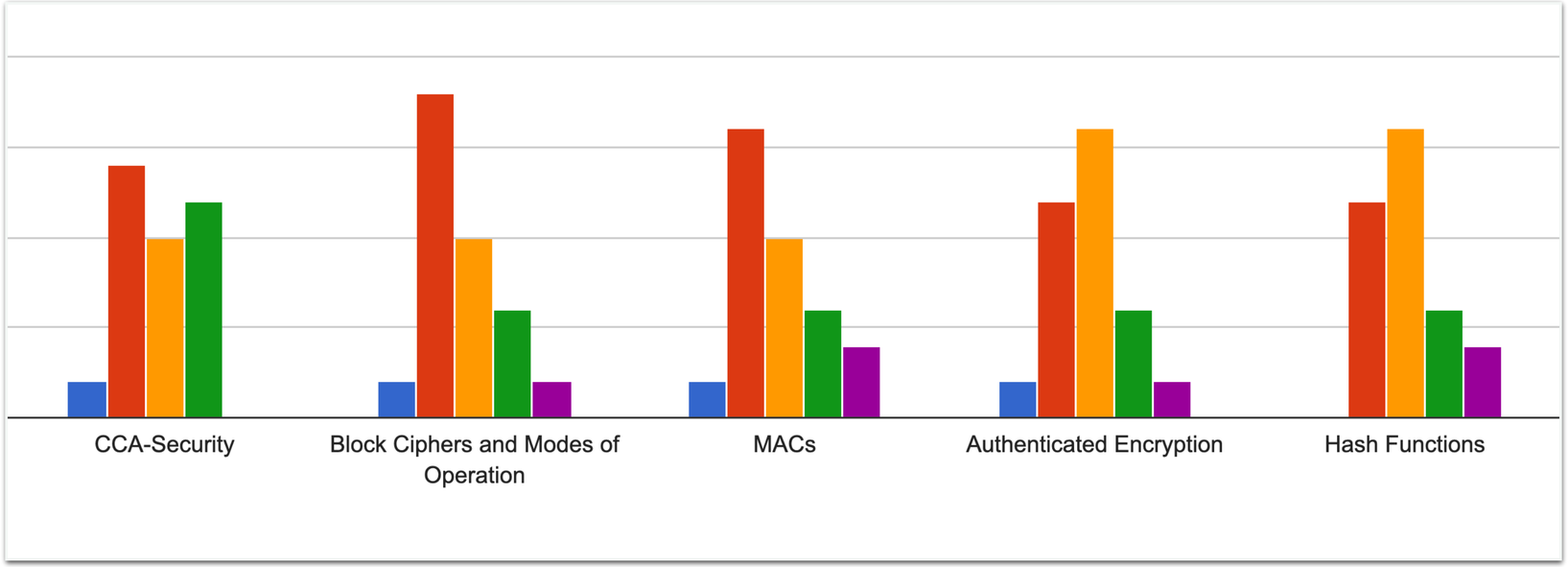
- Thank you everyone who gave feedback!
- To receive extra credit, you were asked to schedule an email to send Friday
  - I received far fewer emails than responses to the feedback form…
  - If you received an “incomplete” on Courseworks for this extra credit, please check you sent the email to claim credit
    - Please forward any relevant info to me so I can give you credit!
- I’m interested in this as a project in case anyone is interested… :)

In your opinion, how well do you feel you understand the material? You may select more than one option.

21 responses




In your opinion, how comfortable do you feel with the the following topics? You do not have to answer every row
10.0
7.5
5.0
2.5
0.0
I do not understand this topic
I understand a little bit
I understand the high level but have trouble applying it
I understand the high level idea
I understand it
Proofs
One-time pad and perfect secrecy
Eavesdropper security
PRGs
CPA-Security
PRFs
OWFs


CCA-Security
Block Ciphers and Modes of Operation
MACs
Authenticated Encryption
Hash Functions

Mid-Semester Feedback

Common suggestions: (currently trying to implement with varying success)

- Slow down!
  - Content and how fast I speak
  - Go slower in examples and have more interaction for next steps
- More problem solving (hopefully Tony's review session helped)

Mid-Semester Feedback

Unexpected results:

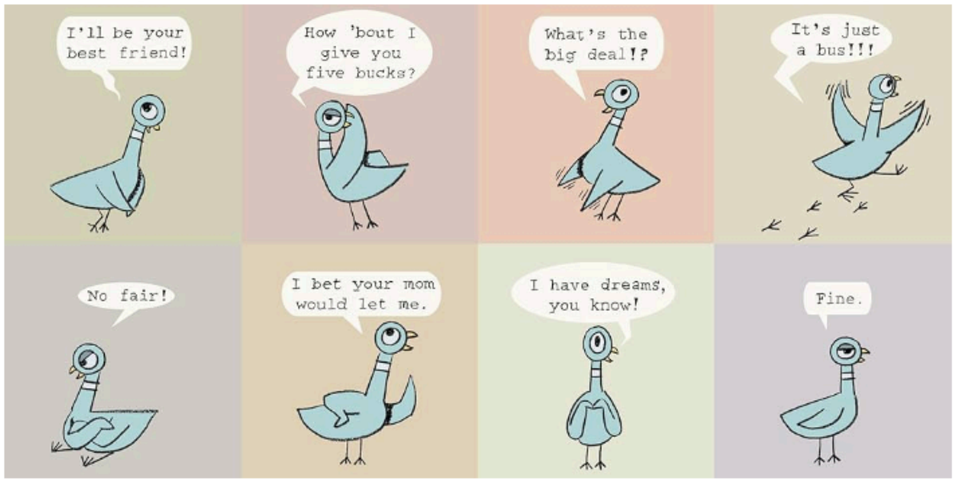
- I’m not the only one who loves pigeons

Hello Professor Lee,

Thank you for the extra credit. I attached a picture of Mr. Pigeon from Miraculous Ladybug.

Hi Prof. Lee,

Hi! I'm the bus driver! Listen, I've got to leave for a little while, so can you watch things for me until I get back? Thanks. Oh, and remember: Don't Let the Pigeon Drive the Bus!



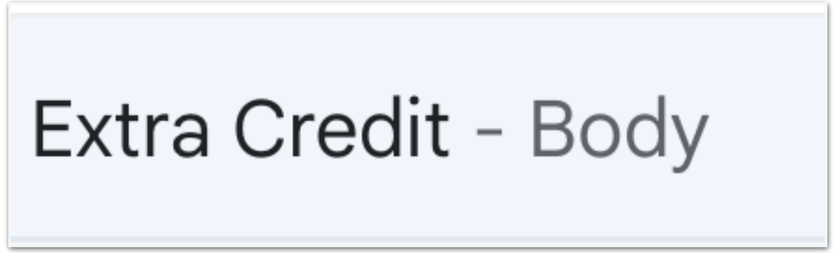
pigeon

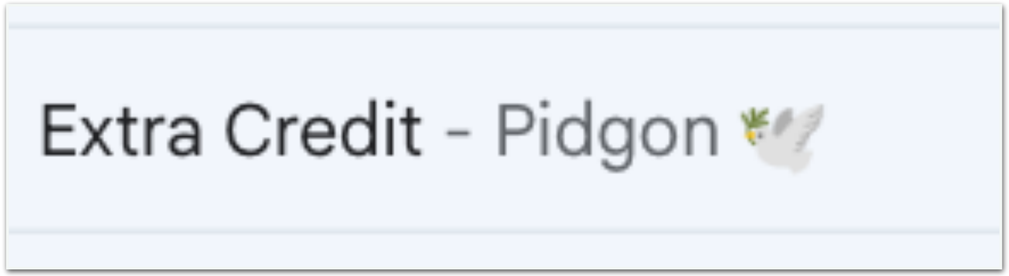
Hi Professor!

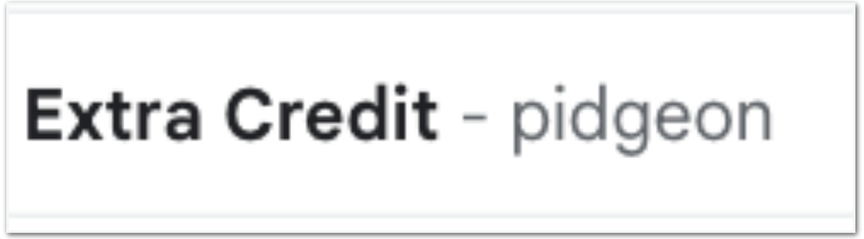
Mid-Semester Feedback

Unexpected results:

- I’m not the only one who loves pigeons
- Pigeon is hard to spell







Today’s Lecture

- More on DL
- More on DH Key Exchange
- Public Key Encryption
- El-Gamal Encryption

Last Time: DL vs CDH vs DDH

For PPT $\mathcal{G}(1^n) \to (\mathbb{G}, g, q)$, we defined the following assumptions:

- **Discrete log assumption**: Given g^x, it is hard to find x
- **Computational DH**: Given g^x, g^y, it is hard to find g^{xy}
- **Decisional DH**: Given g^x, g^y, h, it is hard to tell if $h = g^{xy}$ or if h is random

(See last time's slides for the formal version of these definitions that should be used for assignments)

Attacks on Discrete Log and Prime Order Groups

Attacks on DL and Prime Order Groups

- This next section is a bit of context of why we choose to work over certain groups
    - Why do we keep talking so much about certain groups and say they have to be of a certain form?
- This affects our choice of parameters, but you are *not* required to memorize all of this

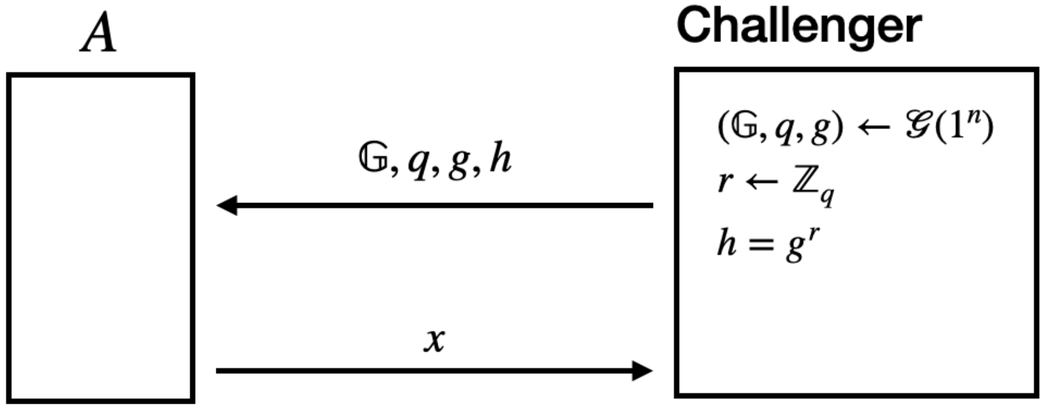
The Discrete Logarithm Assumption

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.

Definition:

The Discrete Log Assumption holds with respect to $\mathcal{G}$ if for all PPT adversaries A there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\Pr[\mathsf{DLA}_{A,\mathcal{G}}(n) = 1] \leq \mathsf{negl}(n)$$



$$\mathsf{DLA}_{A,\mathcal{G}}(n) = \begin{cases} 1 & g^x = h \\ 0 & \text{otherwise} \end{cases}$$

Pholig-Hellman Attack

Why do we require the order of $\mathbb{G}$ to be prime (or near prime)?

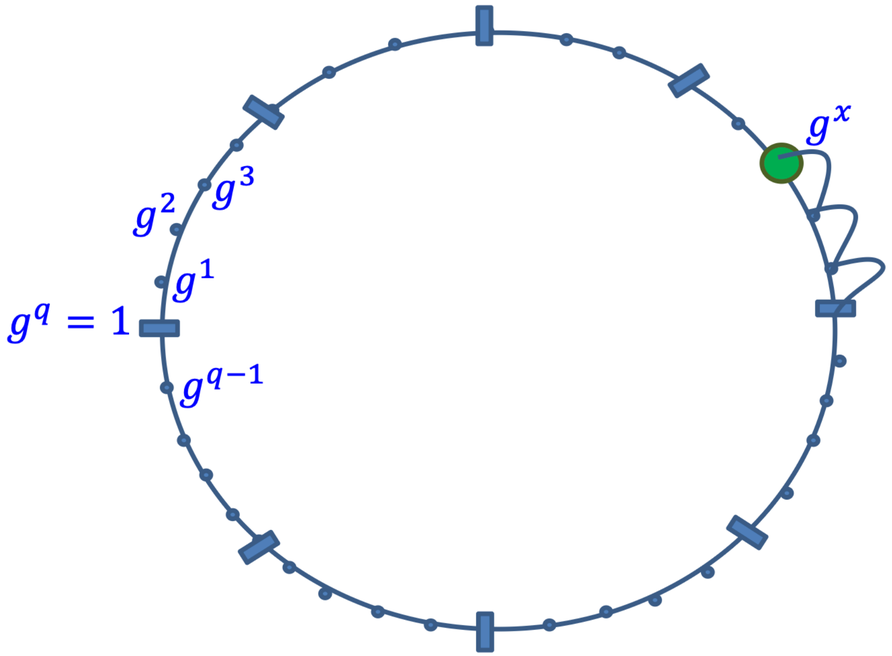
- If not, we can write the order of the group as $q = q_1 \cdot \ldots \cdot q_\ell$ for pairwise co-primes $q_1, \ldots, q_\ell$
- Using CRT, there are subgroups $\mathbb{G}_1, \ldots, \mathbb{G}_\ell$ such that $|\mathbb{G}_i| = q_i$ and $\mathbb{G} \cong \mathbb{G}_1 \times \ldots \times \mathbb{G}_\ell$
  - So the hardness of discrete log in $\mathbb{G}$ corresponds to the size of the maximal subgroup of $\mathbb{G}$
  - If $\mathbb{G}$ has prime order, there are no subgroups!

Baby-Step Giant-Step [Shanks]

Theorem: Let $\mathbb{G}$ be a group of order q. Then, the DL can be solved in $O(\sqrt{q})$ steps

Proof idea: (for simplicity, assume $\sqrt{q}$ is an integer

- **Giant step**:
  1. Create a table of all pairs of elements
     $$\left(1, g^{\sqrt{q}}\right), \left(2, g^{2\sqrt{q}}\right), \ldots, \left(\sqrt{q}, g^{q}\right)$$
  2. Sort by the second coordinate
- **Baby step**: Given g^x
  3. Compute $g^x, g^{x+1}, g^{x+2}, \ldots$
  4. Stop when $g^{x+j} = g^{i\sqrt{q}}$ is found
  5. Set $x = \left[i\sqrt{q} - j \bmod q \right]$



Other Attacks

- Downside of baby-step giant-step is that it requires $O(\sqrt{q})$ memory

- Pollard's Kangaroos (aka Pollard's Rho algorithm) is a randomized version that requires constant memory

- For generic groups (i.e., without exploiting any special properties of the group representation) these are the best known algorithms

- $\mathbb{Z}_q^*$ there are faster, sub-exponential attacks (index calculus)

Why Prime-Order Groups?

- We typically prefer working in prime-order groups
  - Every element (other than the identity) is a generator
  - Pholig-Hellman attack does not apply
- Note that $|\mathbb{Z}_p^*| = p - 1$

Prime-Order Subgroup of $\mathbb{Z}_p^*$

- p is a safe prime if $p = 2q + 1$ and q is prime (e.g., $p = 11$ and $q = 5$)

Theorem: Let $p = 2q + 1$ be a safe prime. Then the group of quadratic residues $\text{QR}_p = \{[h^2 \bmod q] \,|\, h \in \mathbb{Z}_p^*\}$ is of order q

Prime-Order Subgroup of $\mathbb{Z}_p^*$

Theorem: Let $p = 2q + 1$ be a safe prime. Then the group of quadratic residues $\text{QR}_p = \{[h^2 \bmod q] \,|\, h \in \mathbb{Z}_p^*\}$ is of order q

Note: We won't go into much detail about this, but DDH does *not* hold in $\mathbb{Z}_p^*$ but it is believed in hold in QR_p

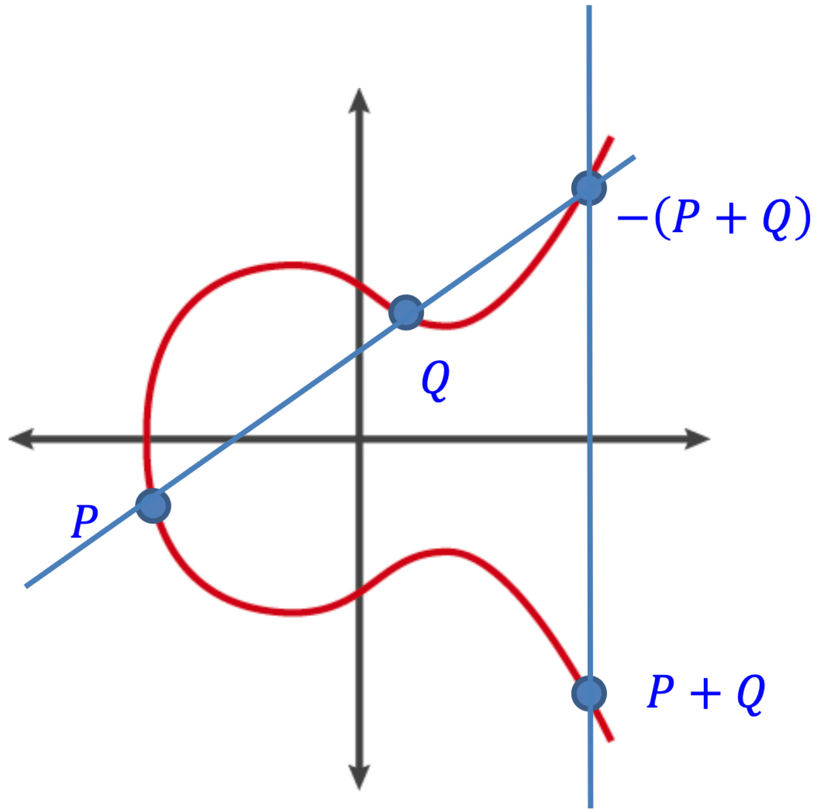
- When we talk about DH "mod p", we're actually working over QR_p
  - That is, $\mathcal{G}(1^n)$ chooses a safe prime p and sets $q = (p-1)/2$ and $g = \tilde{g}^2 \bmod p$ as the generator

Elliptic Curves

An elliptic curve is a geometric object defined by the equation

$$y^2 = x^3 + ax + b$$

where $4a^3 + 27b^2 \neq 0$



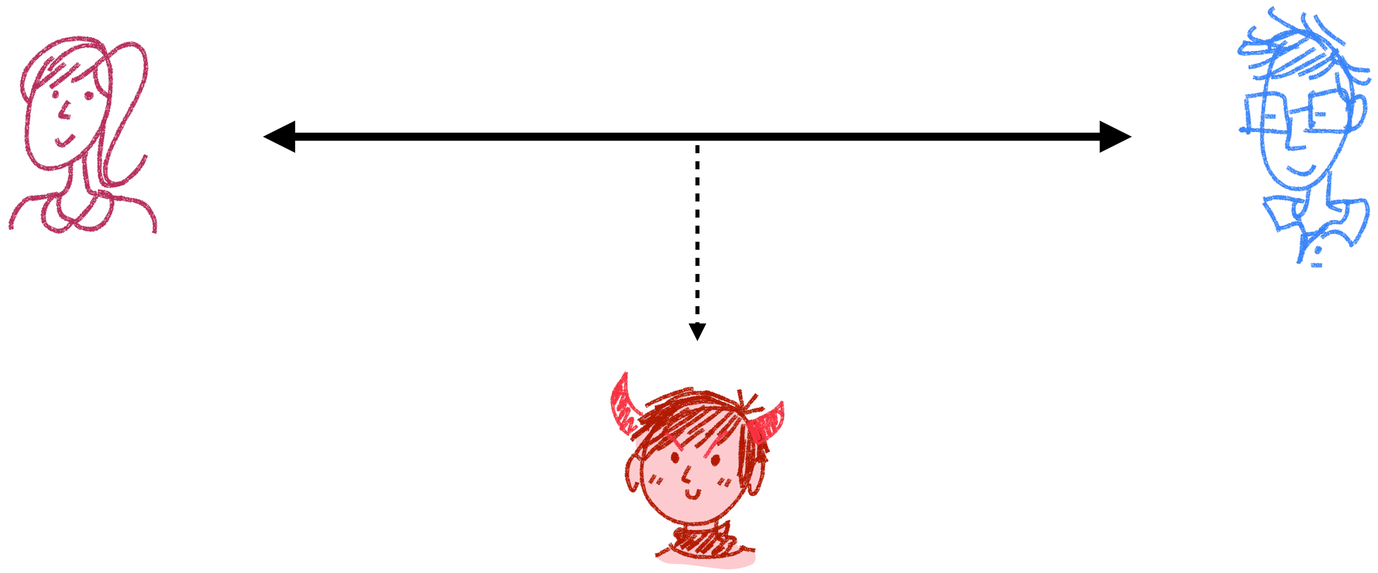
- The points on the curve form a group
- For the right choice of elliptic curve, there are no known subexponential attacks!

More on DH Key Exchange

Last Time: Key Exchange

Alice and Bob don’t have an existing shared secret key.

Can they communicate over a public channel to agree on a secret key?



Last Time: Diffie-Hellman Key Exchange

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

$x \leftarrow \mathbb{Z}_q$

$h_a = g^x$

$(\mathbb{G}, q, g),\ h_a$

h_b

$y \leftarrow \mathbb{Z}_q$

$h_b = g^y$

$k = (h_b)^x$

$k = (h_a)^y$

Last Time: Diffie-Hellman Key Exchange

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

$x \leftarrow \mathbb{Z}_q$

$h_a = g^x$

$(\mathbb{G}, q, g),\ h_a$ →

← h_b

$y \leftarrow \mathbb{Z}_q$

$h_b = g^y$

$k = (h_b)^x$

$k = (h_a)^y$

This protocol is secure against a passive adversary

(i.e., one that can only observe messages)

Last Time: Diffie-Hellman Key Exchange

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

$x \leftarrow \mathbb{Z}_q$

$h_a = g^x$

$(\mathbb{G}, q, g),\ h_a$ →

← h_b

$y \leftarrow \mathbb{Z}_q$

$h_b = g^y$

$k = (h_b)^x$

$k = (h_a)^y$

This protocol is secure against a passive adversary
(i.e., one that can only observe messages)

What about an active adversary?
(i.e., one that can change or insert messages)

Man-in-the-Middle Attack

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

$x \leftarrow \mathbb{Z}_q$

$h_a = g^x$

$(\mathbb{G}, q, g),\ h_a$ →

Man-in-the-Middle Attack

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

$x \leftarrow \mathbb{Z}_q$

$h_a = g^x$

$(\mathbb{G}, q, g),\ h_a$ ⟶

$z \leftarrow \mathbb{Z}_q$

$\tilde{h} = g^z$

Man-in-the-Middle Attack

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

$x \leftarrow \mathbb{Z}_q$

$h_a = g^x$

$(\mathbb{G}, q, g),\ h_a$

$z \leftarrow \mathbb{Z}_q$

$\tilde{h} = g^z$

$(\mathbb{G}, q, g),\ \tilde{h}$

$y \leftarrow \mathbb{Z}_q$

$h_b = g^y$

Man-in-the-Middle Attack

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

$x \leftarrow \mathbb{Z}_q$

$h_a = g^x$

$(\mathbb{G}, q, g),\ h_a$ →

$z \leftarrow \mathbb{Z}_q$

$\tilde{h} = g^z$

$(\mathbb{G}, q, g),\ \tilde{h}$ →

$y \leftarrow \mathbb{Z}_q$

$h_b = g^y$

← $\tilde{h}$

← h_b

$k_1 = (\tilde{h})^x$

$k_2 = (\tilde{h})^y$

Man-in-the-Middle Attack

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

$x \leftarrow \mathbb{Z}_q$

$h_a = g^x$

$(\mathbb{G}, q, g),\ h_a$ →

$z \leftarrow \mathbb{Z}_q$

$\tilde{h} = g^z$

$(\mathbb{G}, q, g),\ \tilde{h}$ →

$y \leftarrow \mathbb{Z}_q$

$h_b = g^y$

← $\tilde{h}$

← h_b

$k_1 = (\tilde{h})^x$

$k_1 = (h_a)^z$

$k_2 = (h_b)^z$

$k_2 = (\tilde{h})^y$

Man-in-the-Middle Attack

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

$x \leftarrow \mathbb{Z}_q$

$h_a = g^x$

$(\mathbb{G}, q, g),\ h_a$ →

$z \leftarrow \mathbb{Z}_q$

$\tilde{h} = g^z$

$(\mathbb{G}, q, g),\ \tilde{h}$ →

$y \leftarrow \mathbb{Z}_q$

$h_b = g^y$

← $\tilde{h}$

← h_b

$k_1 = (\tilde{h})^x$

$k_1 = (h_a)^z$

$k_2 = (h_b)^z$

$k_2 = (\tilde{h})^y$

Alice and Bob both think they're talking to the other, and
Eve learns the conversation without being detected!

Man-in-the-Middle Attack

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

$x \leftarrow \mathbb{Z}_q$

$h_a = g^x$

$(\mathbb{G}, q, g),\ h_a$ →

$z \leftarrow$

$\tilde{h} =$

← $\tilde{h}$

$k_1 = (\tilde{h})^x$

$k_1 = (h_a)^z$

$k_2 = (h_b)^z$

$k_2 = (\tilde{h})^y$

Preventing this type of attack is a rich area of research (e.g., “certificate authorities” and authenticated channels) that is outside of the scope of this class

Alice and Bob both think they’re talking to the other, and Eve learns the conversation without being detected!

Public Key Encryption

Recall: Private-Key Encryption

In symmetric key (private-key) encryption, Enc and Dec using the same key

Recall: Private-Key Encryption

Security definitions for symmetric encryption:

- EAV-security
- CPA-Security: Additional encryption oracle
- CCA-Security: Additional decryption oracle
- Authenticated encryption: Adversary cannot create new ciphertexts

Public-Key Encryption

pk

c

sk

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

pk

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

Public encryption key

Secret decryption key

Public-Key Encryption

pk

c

sk

pk

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

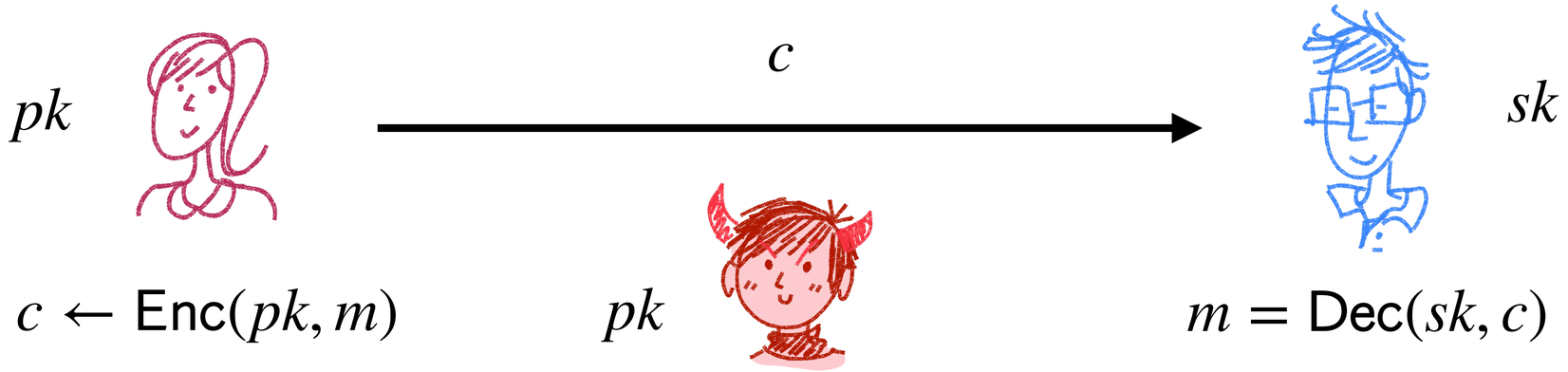
$m = \mathsf{Dec}(sk, c)$

Public encryption key

Secret decryption key

How should we define security for this setting?

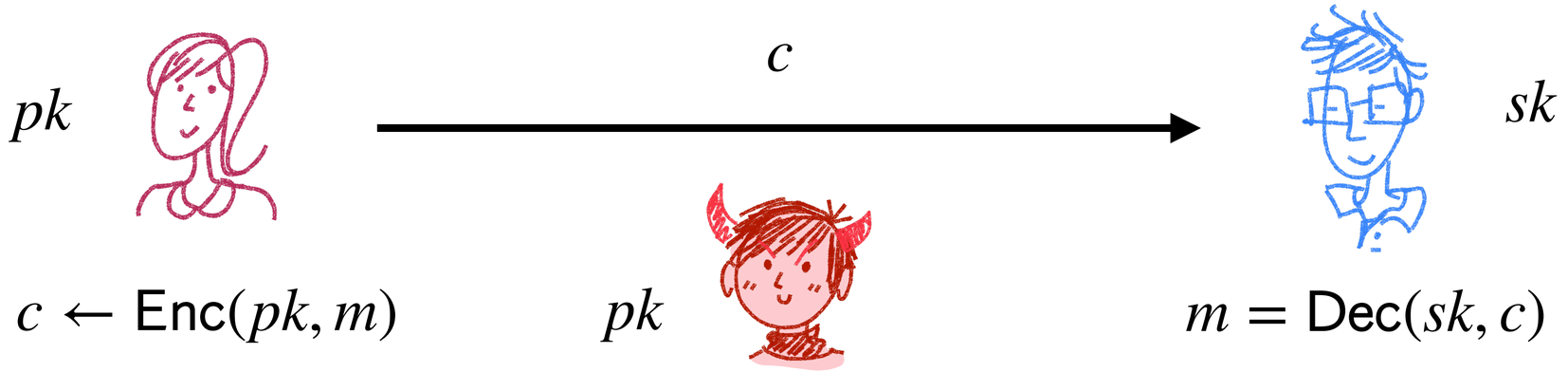
Public-Key Encryption



Candidate security definitions (from symmetric encryption):

- EAV-security
- CPA-Security: Additional encryption oracle
- CCA-Security: Additional decryption oracle
- Authenticated encryption: Adversary cannot create new ciphertexts

Public-Key Encryption

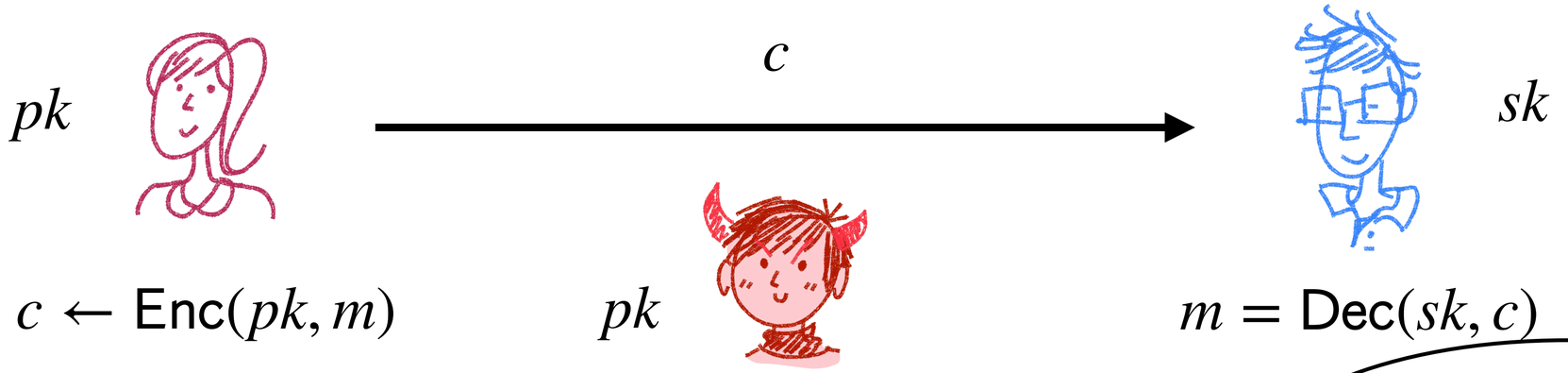


Candidate security definitions (from symmetric encryption):

- EAV-security
- CPA-Security: Additional encryption oracle
- CCA-Security: Additional decryption oracle
- Authenticated encryption: Adversary cannot create new ciphertexts

What's the difference between EAV and CPA?

Public-Key Encryption



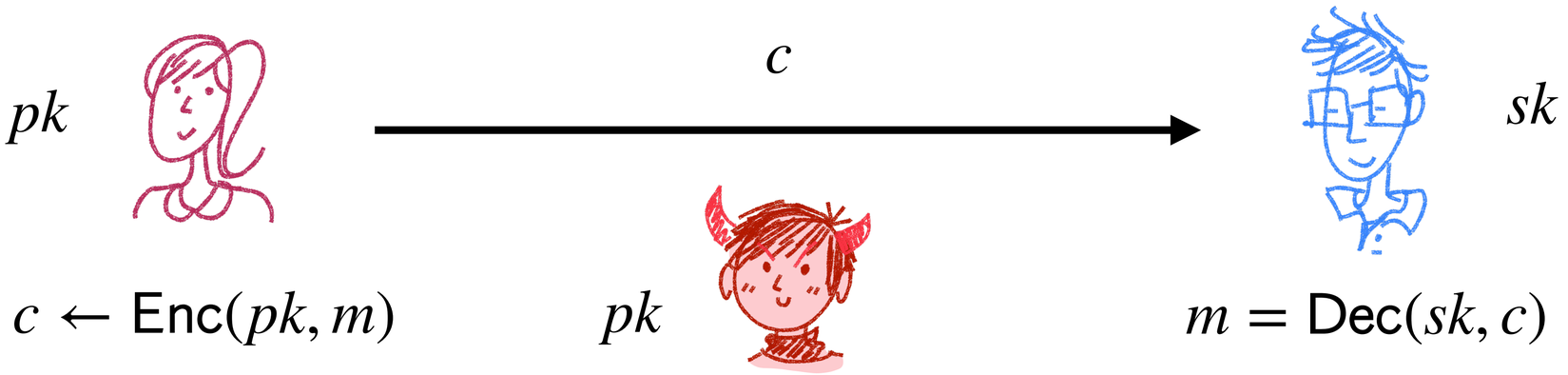
Candidate security definitions (from symmetric encrypti

- EAV-security
- CPA-Security: Additional encryption oracle
- CCA-Security: Additional decryption oracle
- Authenticated encryption: Adversary cannot create new ciphertexts

What's the difference between EAV and CPA?

Is this reasonable?

Public-Key Encryption



Syntax: $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$

- Key generation: $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$
- Encryption: $c \leftarrow \mathsf{Enc}(pk, m)$
- Decryption: $m = \mathsf{Dec}(sk, c)$

Correctness:

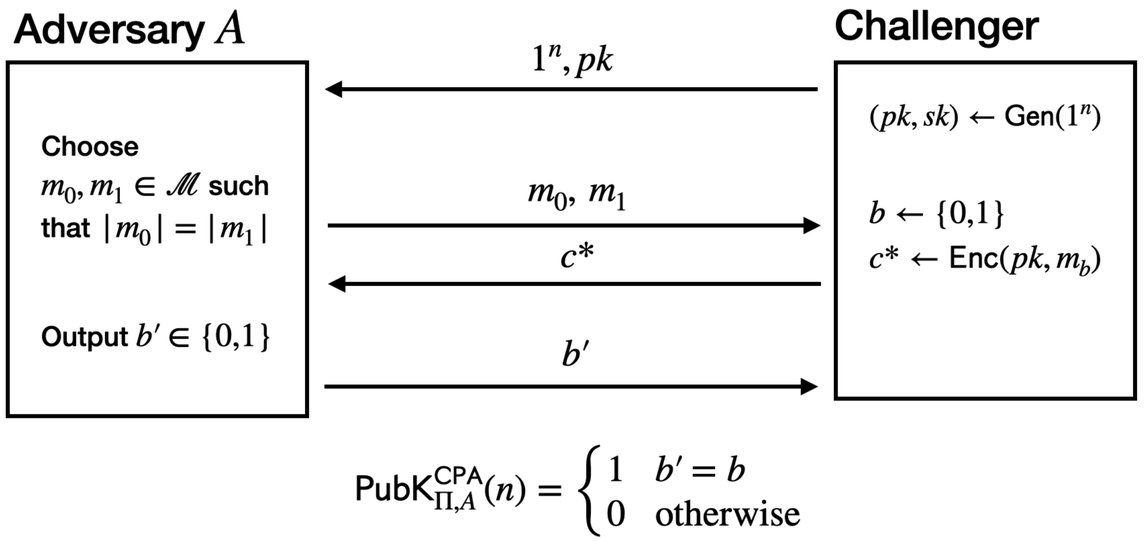
$$\Pr[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m] = 1$$

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:

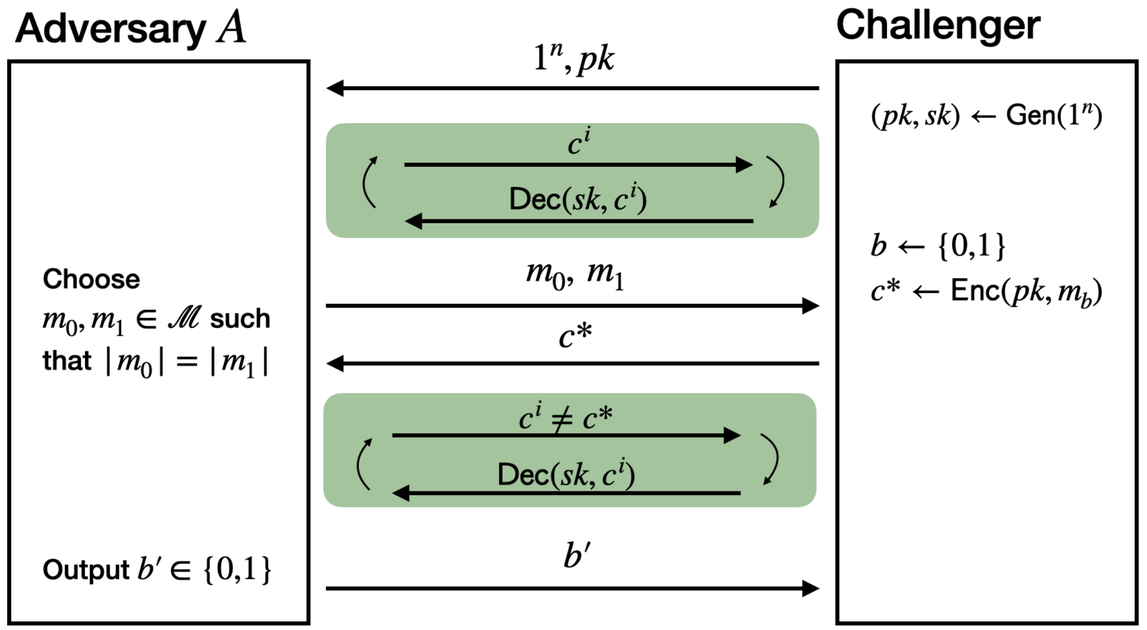
- No encryption oracle in the public key setting
- Similar to the private-key setting, encryption must be **randomized**

Chosen-Ciphertext Attack (CCA)

Definition:

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

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



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

El-Gamal Encryption

Recall: Diffie-Hellman Key Exchange

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

$x \leftarrow \mathbb{Z}_q$

$h_a = g^x$

$(\mathbb{G}, q, g),\ h_a$ →

← h_b

$k = (h_b)^x$

$y \leftarrow \mathbb{Z}_q$

$h_b = g^y$

$k = (h_a)^y$

Let's say Alice and Bob run DH Key Exchange.
Can Bob use this as a basis to send a message to Alice?

Recall: Diffie-Hellman Key Exchange

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

$x \leftarrow \mathbb{Z}_q$

$h_a = g^x$

$(\mathbb{G}, q, g),\ h_a$ →

$y \leftarrow \mathbb{Z}_q$

$h_b = g^y$

← h_b

$k = (h_b)^x$

$k = (h_a)^y$

← $k \cdot m$

$m \in \mathbb{G}$

Let's say Alice and Bob run DH Key Exchange.
Can Bob use this as a basis to send a message to Alice?

A Useful Lemma

Lemma: Let $\mathbb{G}$ be a finite group and let $m \in \mathbb{G}$ be arbitrary.

If $k \leftarrow \mathbb{G}$ is uniformly distributed, then $k' = k \cdot m$ is also uniformly distributed.

Proof:

- It is enough to show that for every $h \in \mathbb{G}$ it holds that

$$\Pr[k \cdot m = h] = 1/|\mathbb{G}|$$

- Fix an arbitrary $h \in \mathbb{G}$. Then $\Pr[k \cdot m = h] = \Pr[k = h \cdot m^{-1}]$
- Since k is uniform, it holds that $\Pr[k = h \cdot m^{-1}] = 1/|\mathbb{G}|$

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

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 $m = c_2 / c_1^x$

There are methods of encoding binary strings as group elements. For simplicity we assume $m \in \mathbb{G}$

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

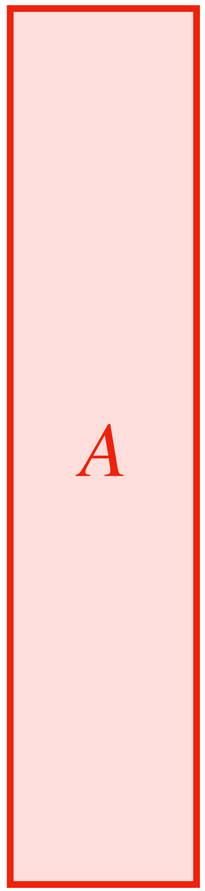
Theorem: Under the DDH assumption, El-Gamal encryption is CPA-secure

Proof:

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

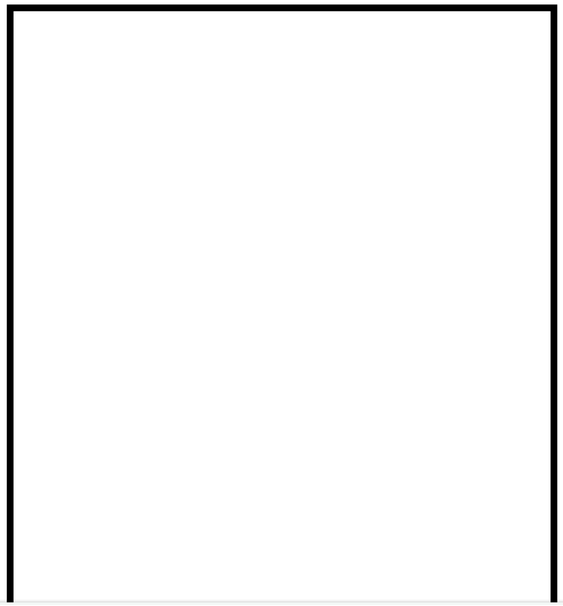
- We will construct a distinguisher D that breaks the DDH assumptions (distinguishes between (g, g^x, g^y, g^{xy}) and (g, g^x, g^y, g^z))

Security of El-Gamal Encryption



Distinguisher D

DDH Challenger

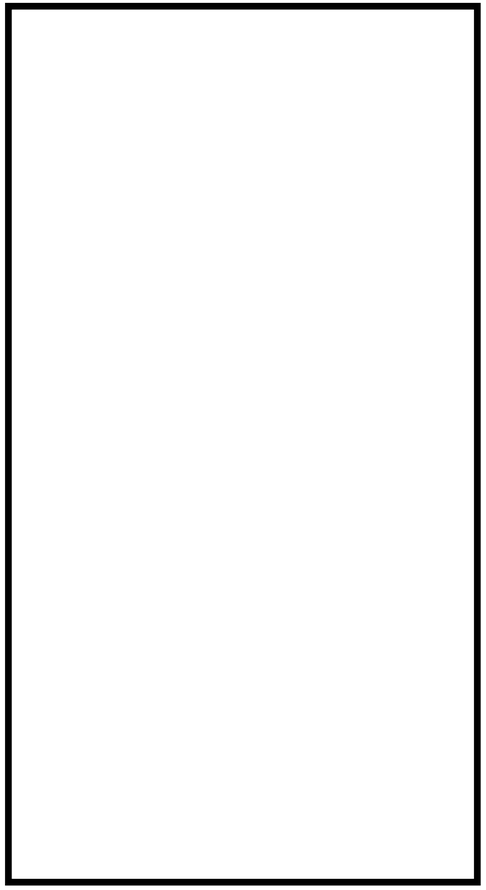


Gen(1^n): Sample $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ and $x \leftarrow \mathbb{Z}_q$. Set $h = g^x$.

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 :
  $h = g^{xy}$

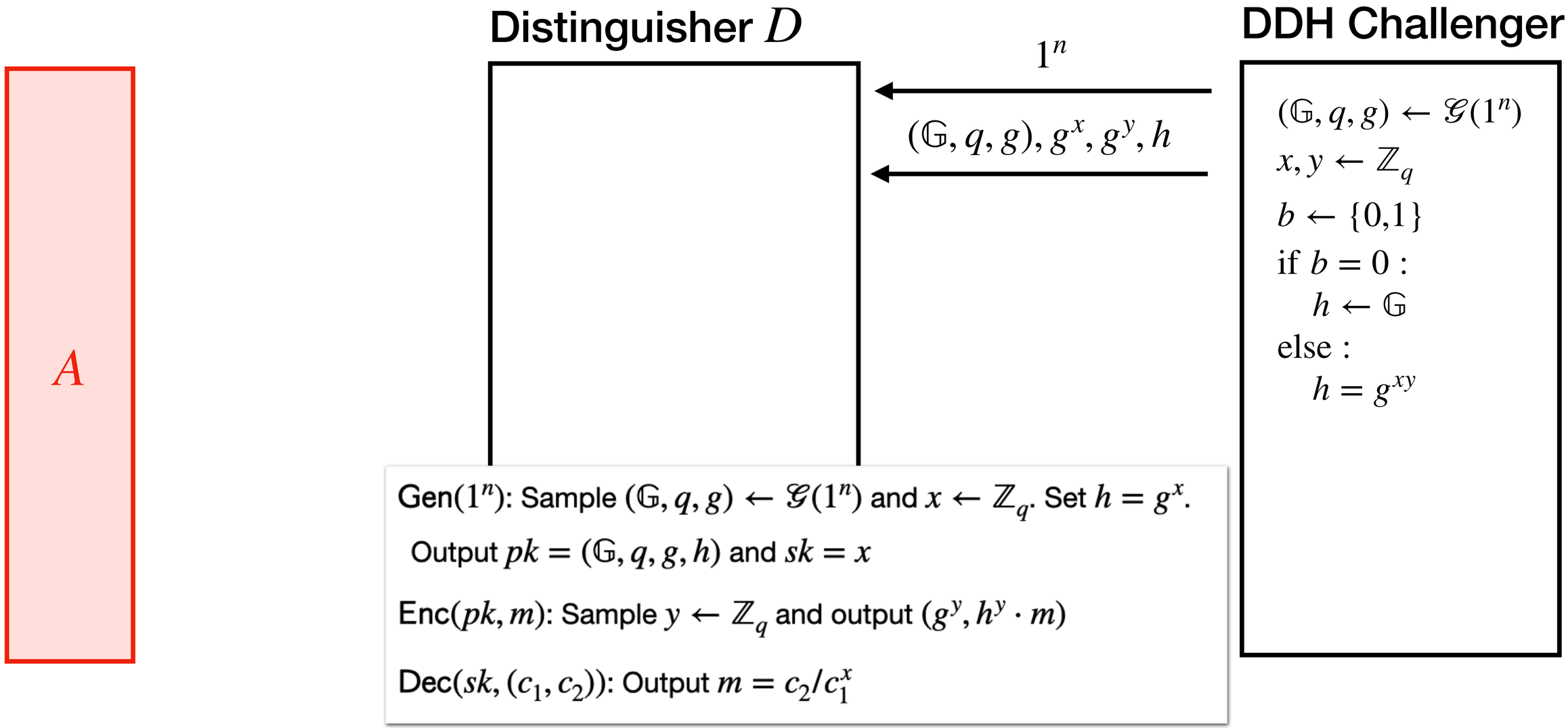
Gen(1^n): Sample $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ and $x \leftarrow \mathbb{Z}_q$. Set $h = g^x$.

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

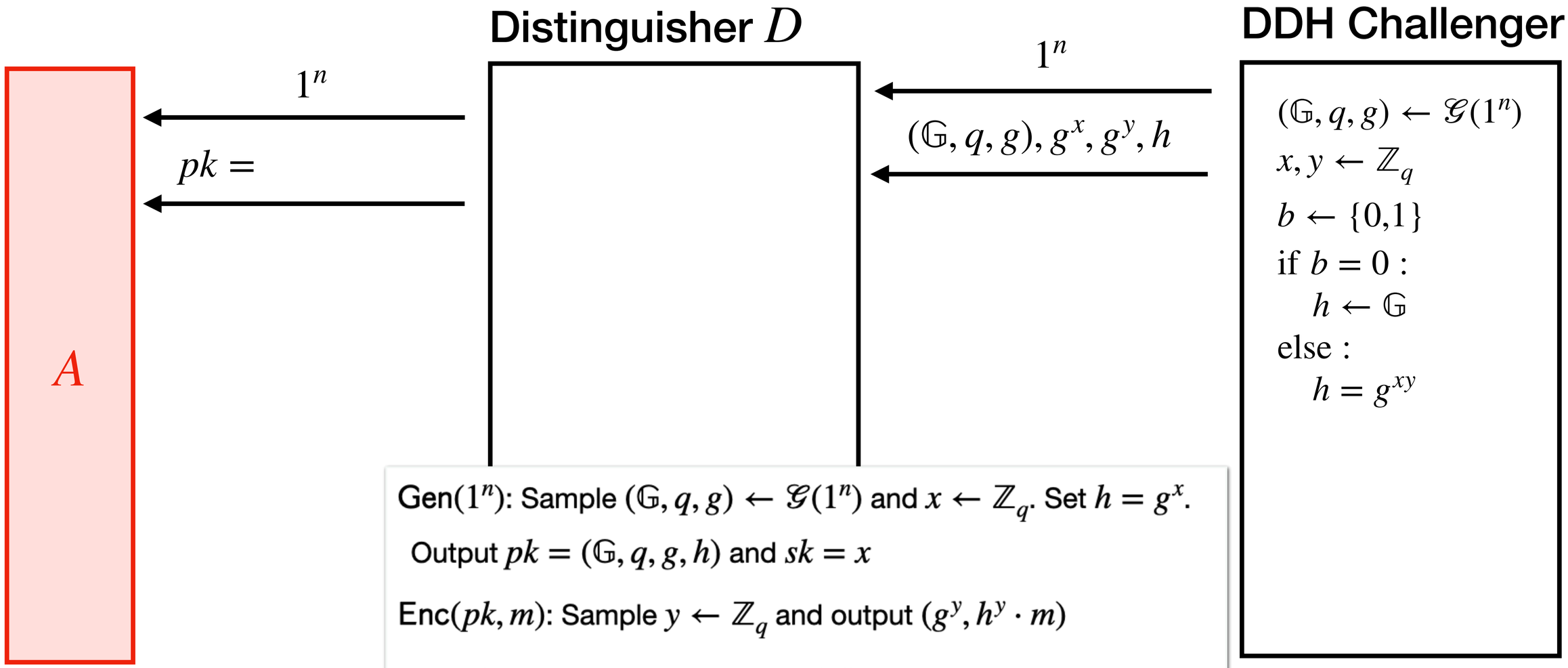
Enc(pk, m): Sample $y \leftarrow \mathbb{Z}_q$ and output $(g^y, h^y \cdot m)$

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

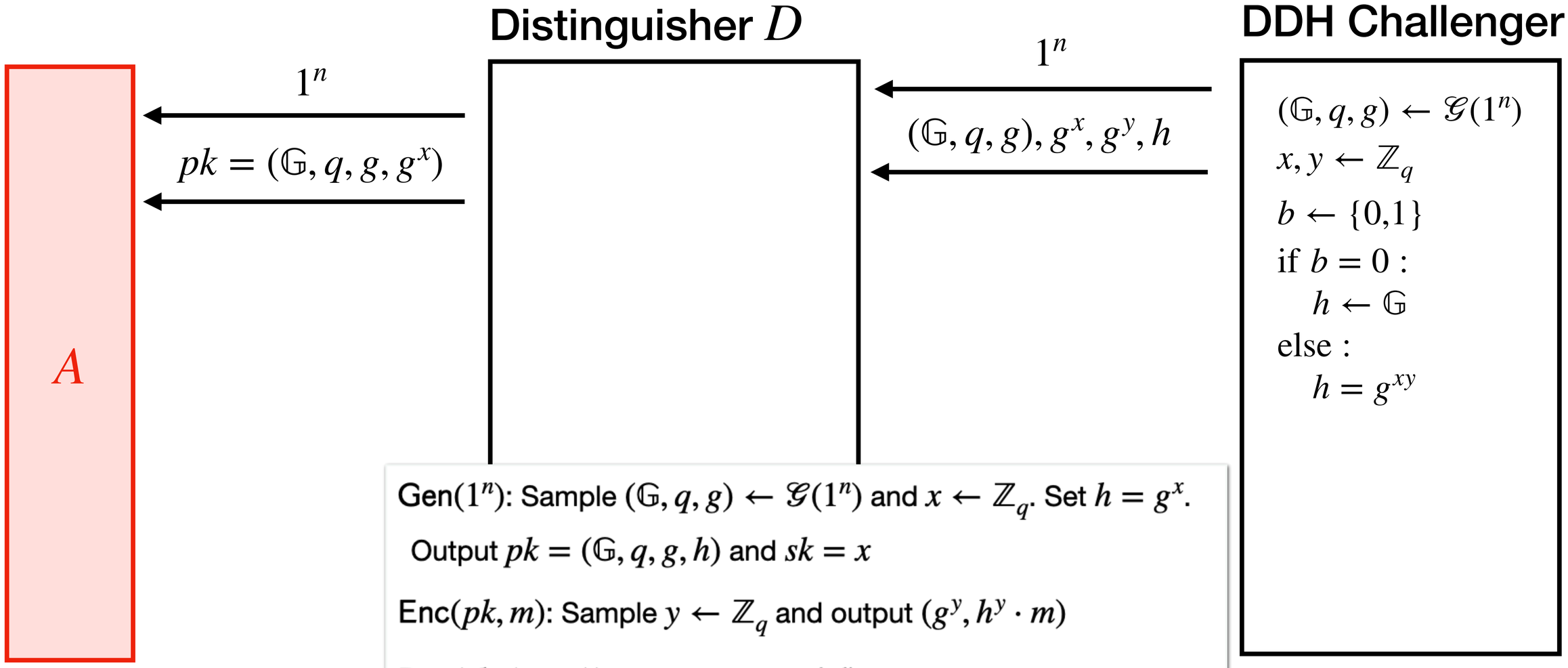
Security of El-Gamal Encryption



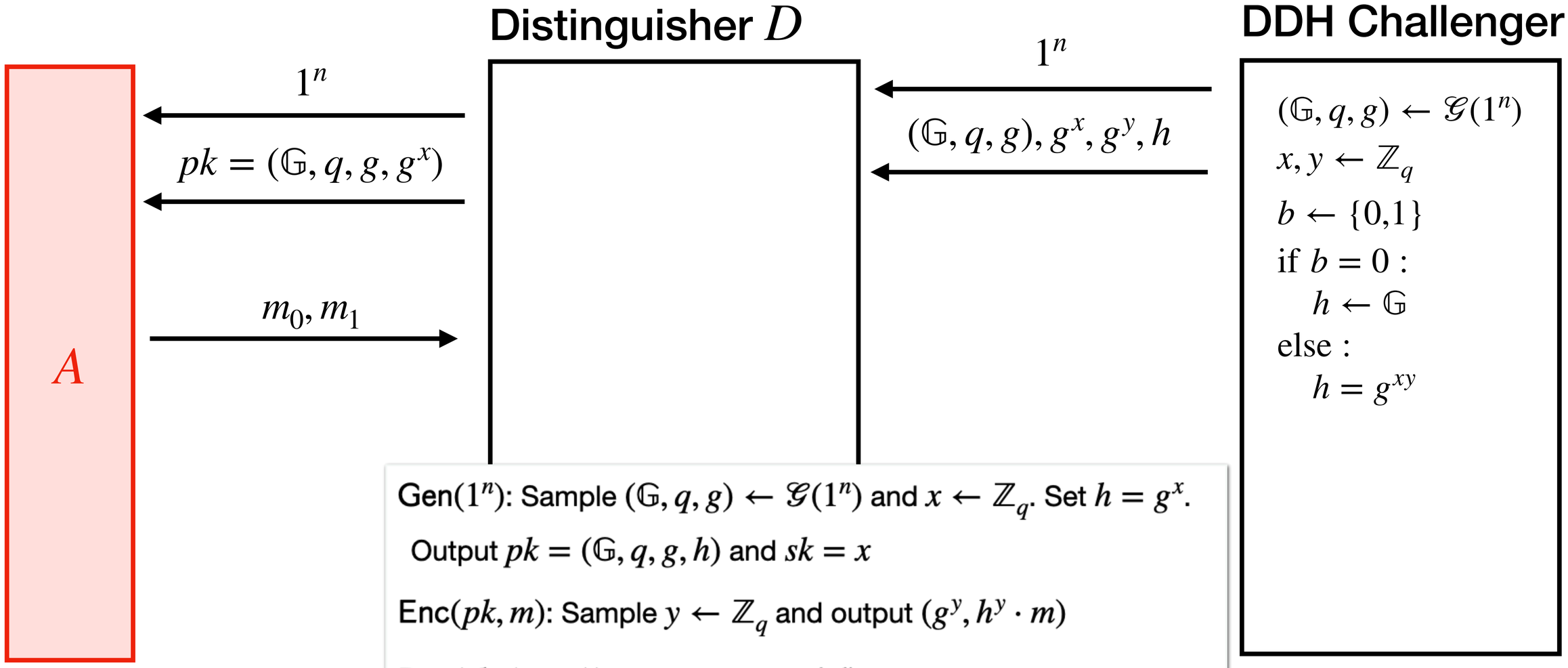
Security of El-Gamal Encryption



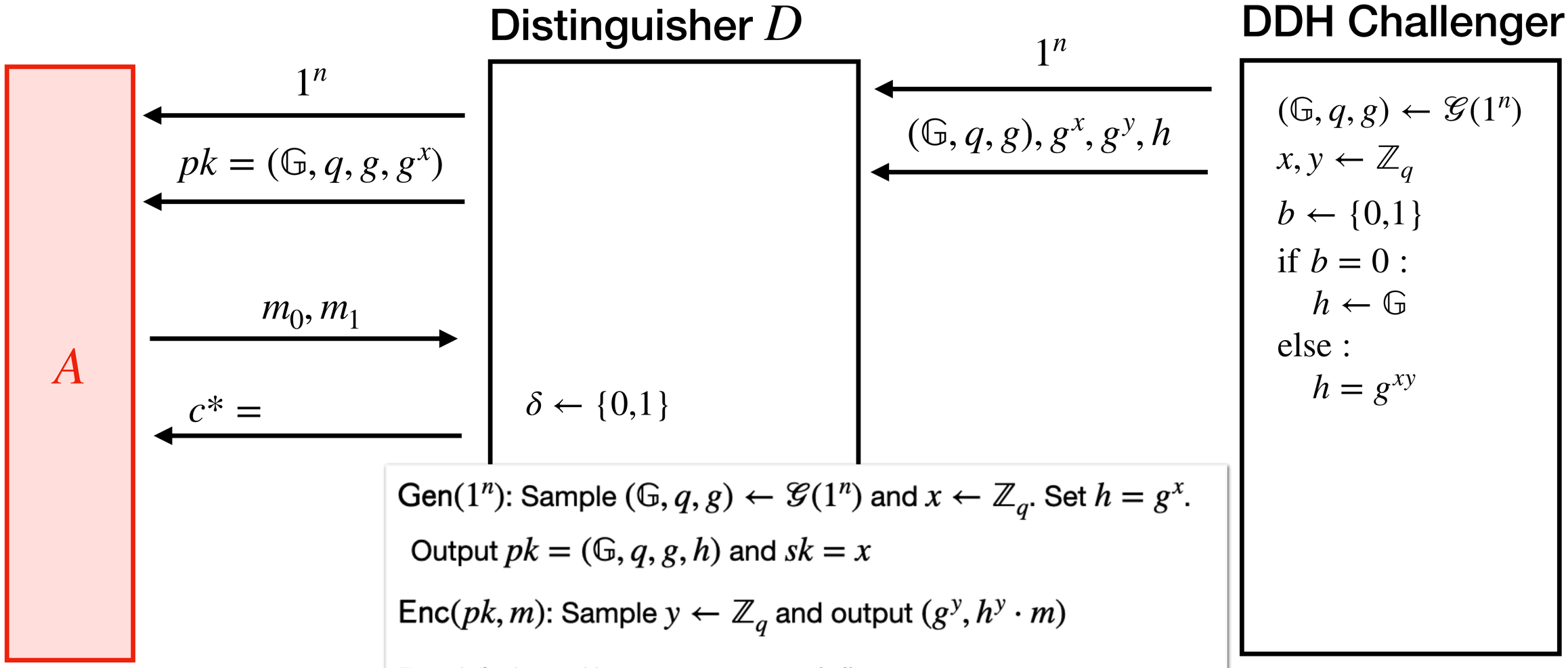
Security of El-Gamal Encryption



Security of El-Gamal Encryption



Security of El-Gamal Encryption



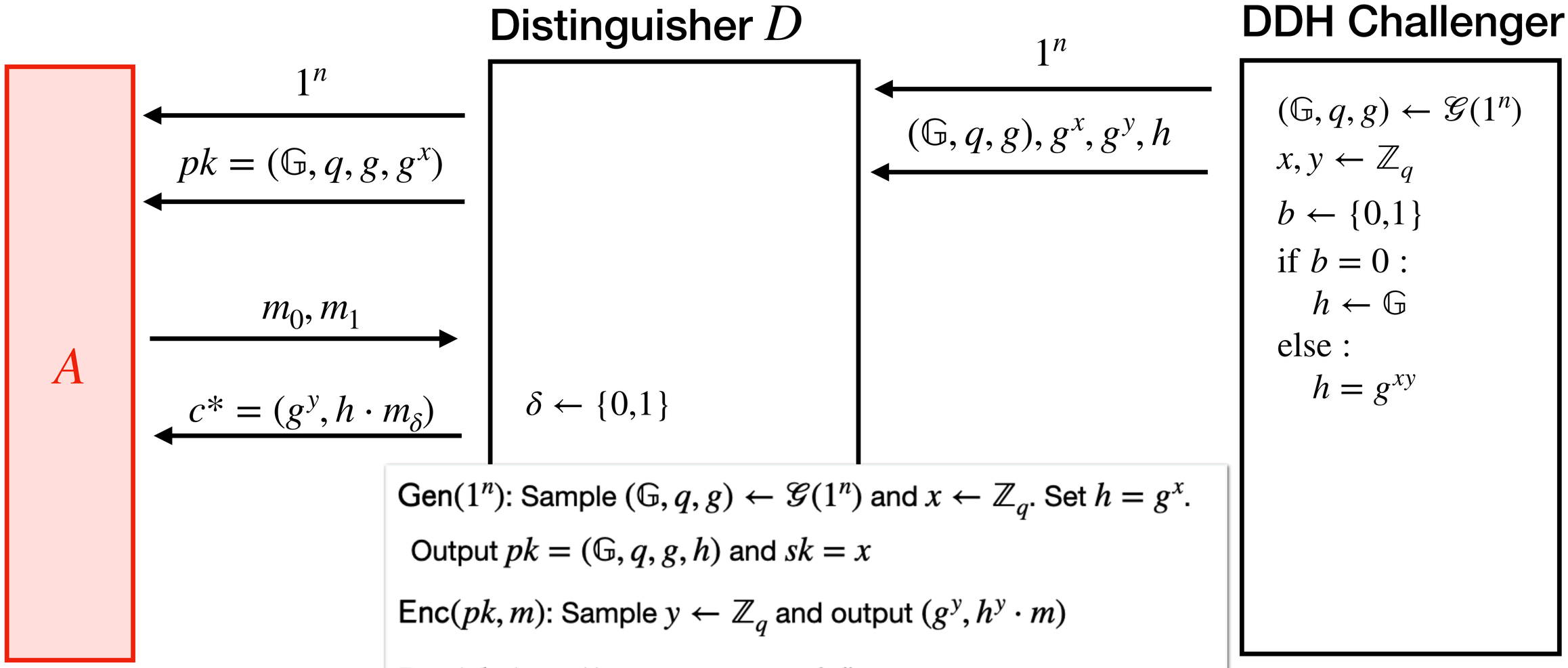
Gen(1^n): Sample $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ and $x \leftarrow \mathbb{Z}_q$. Set $h = g^x$.

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



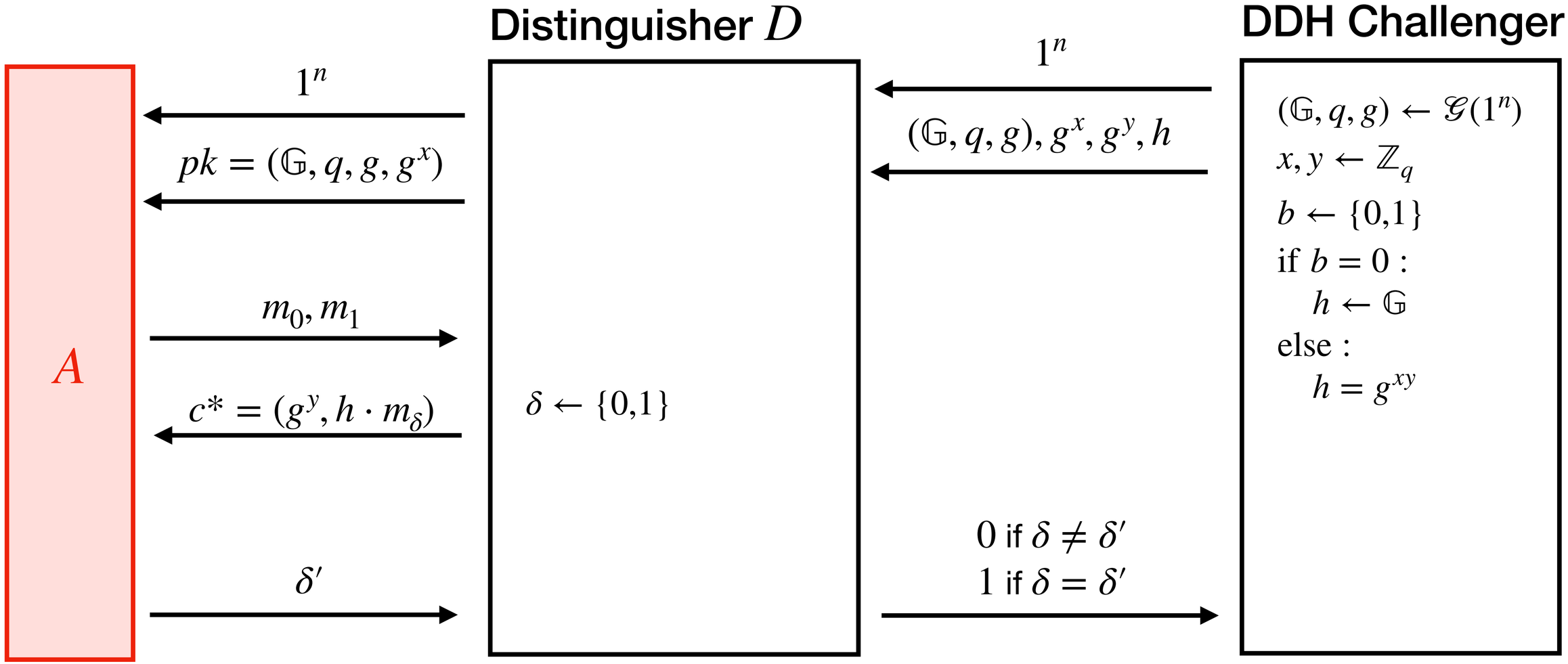
Gen(1^n): Sample $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ and $x \leftarrow \mathbb{Z}_q$. Set $h = g^x$.

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

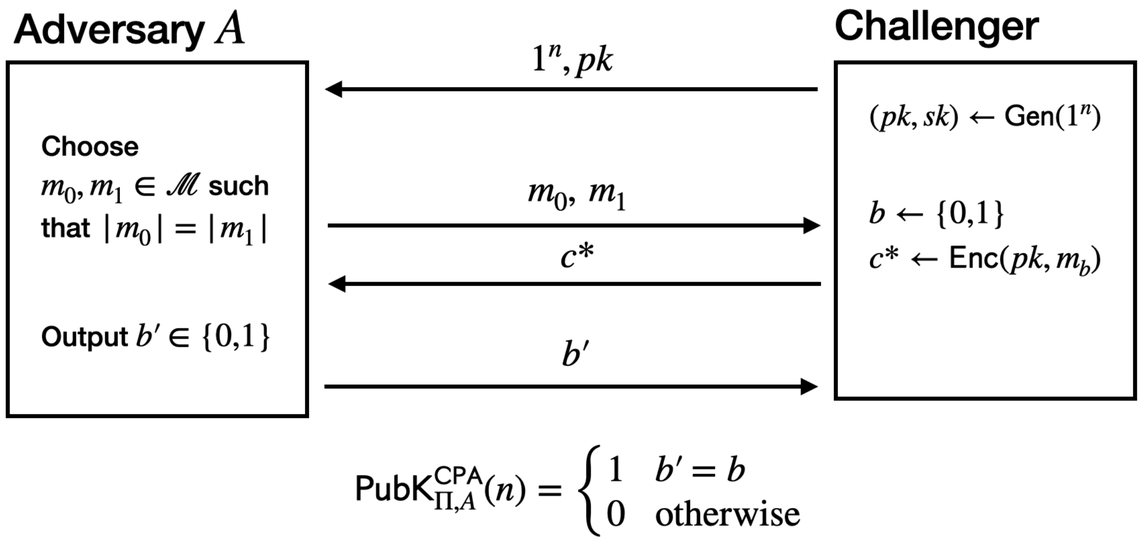


Recall: Chosen-Plaintext Attack (CPA)

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

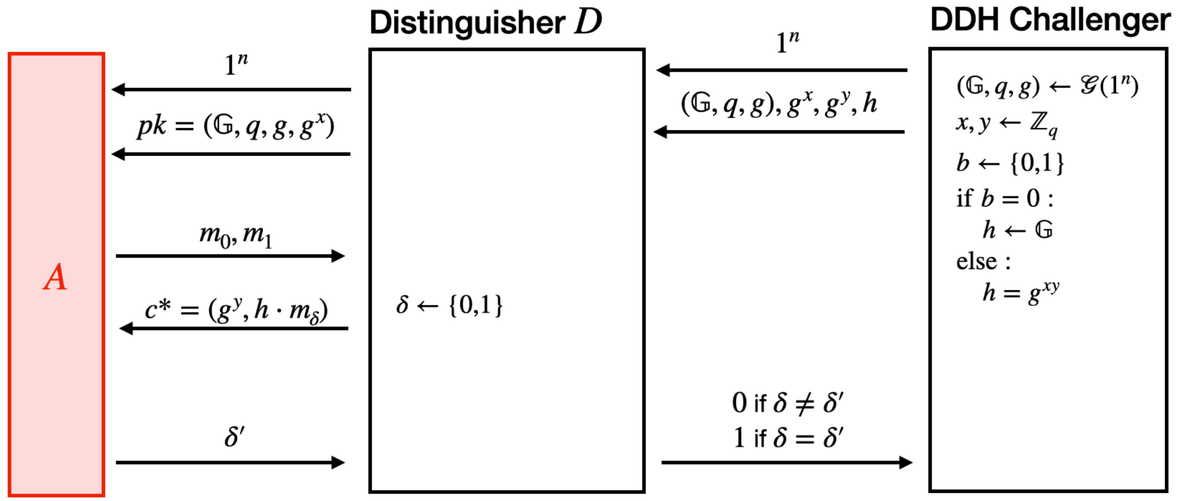
- View of A is identical to its view in the CPA-security experiment

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

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