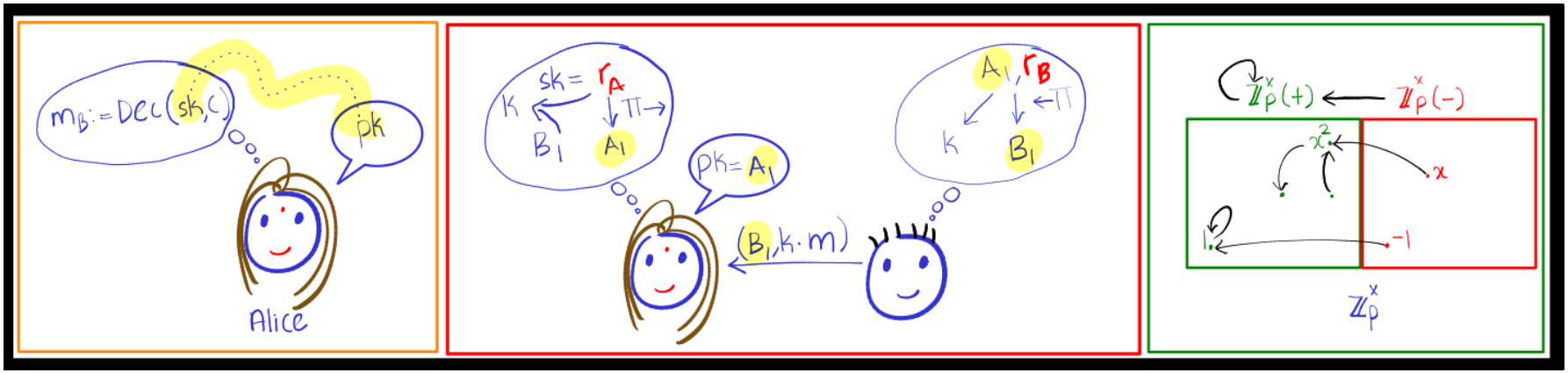



# CS783: Theoretical Foundations of Cryptography

Lecture 9 (27/Aug/24)

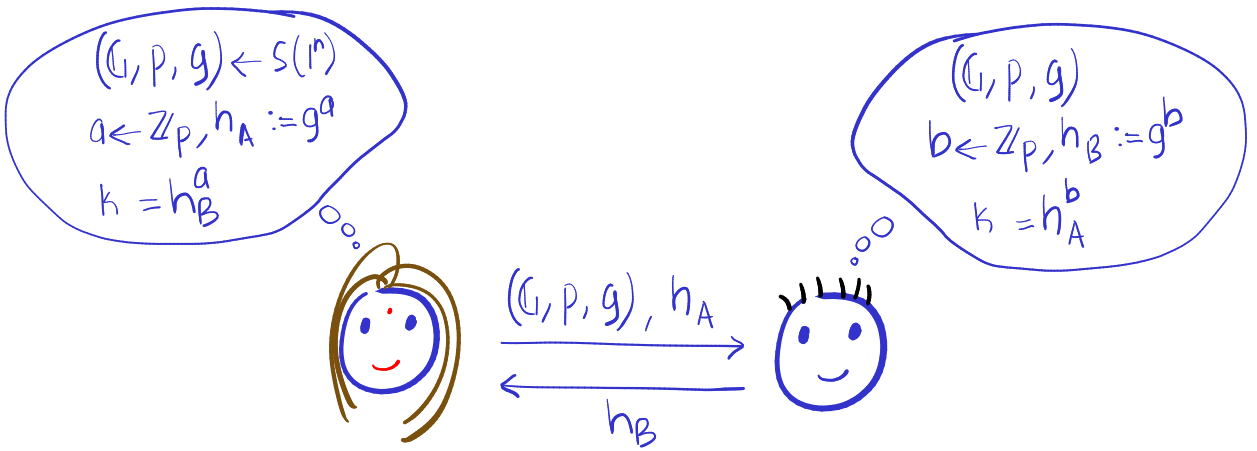
Instructor: Chethan Kamath

# Recall from Last Lecture

- Task 3: sharing key in presence of eavesdroppers
  - Modelled key exchange setting and security

# Recall from Last Lecture

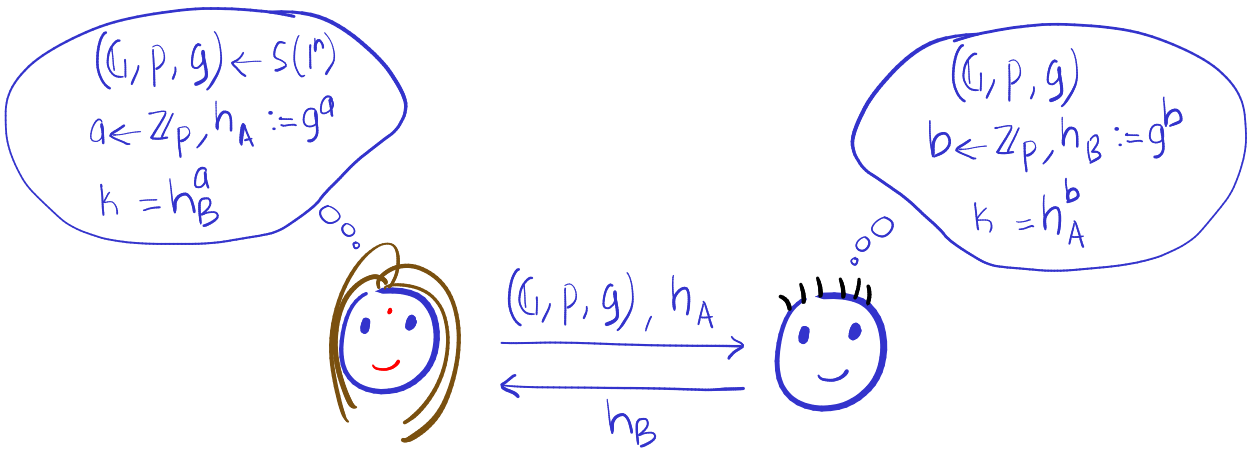
- Task 3: sharing key in presence of eavesdroppers
  - Modelled key exchange setting and security




  - Diffie-Hellman key exchange (DHKE) protocol
    - Basic introduction to groups
    - Based security on the DDH assumption in (prime-order) groups

# Recall from Last Lecture

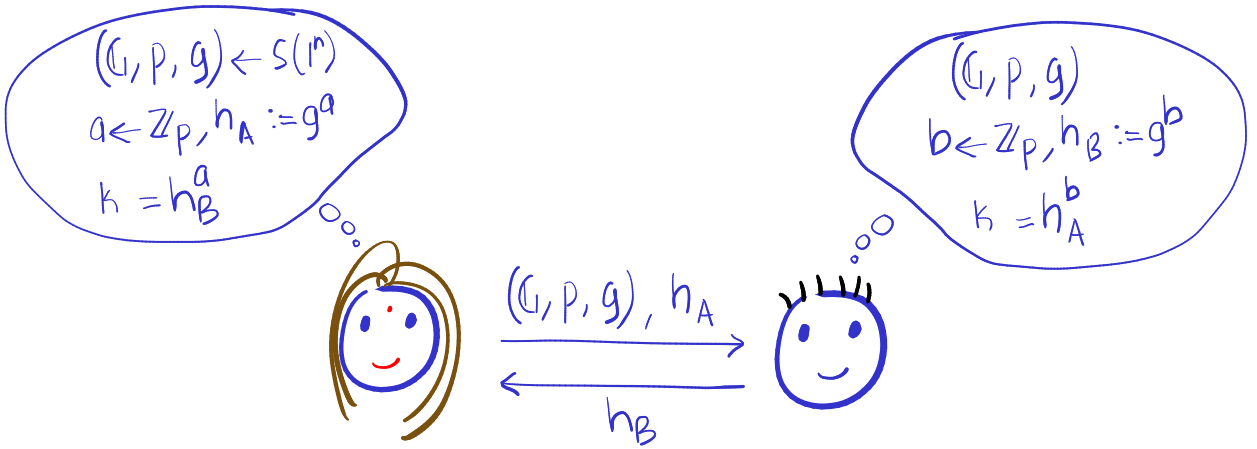
- Task 3: sharing key in presence of eavesdroppers
  - Modelled key exchange setting and security




  - Diffie-Hellman key exchange (DHKE) protocol
    - Basic introduction to groups
    - Based security on the DDH assumption in (prime-order) groups
  - Looked at multi-instance DHKE
    - First proof using hybrid argument
    - Second proof beats hybrid argument via *random self-reducibility*

# Recall from Last Lecture

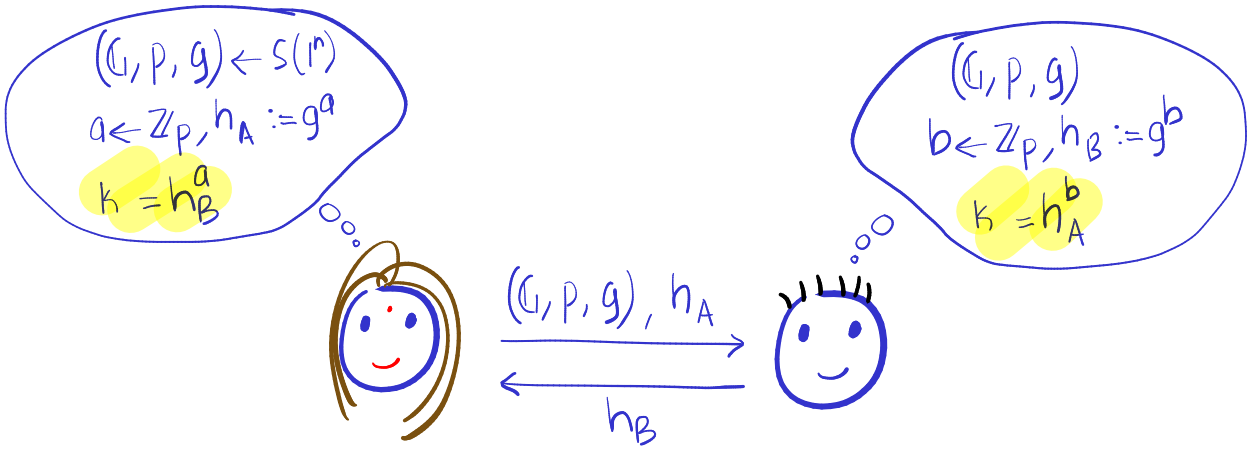
- Task 3: sharing key in presence of eavesdroppers
  - Modelled key exchange setting and security




  - Diffie-Hellman key exchange (DHKE) protocol
    - Basic introduction to groups
    - Based security on the DDH assumption in (prime-order) groups
  - Looked at multi-instance DHKE
    - First proof using hybrid argument
    - Second proof beats hybrid argument via *random self-reducibility*
  - Takeaway: structure vs. hardness
    - Some groups have *sufficient structure* while *remaining hard*

# Recall from Last Lecture

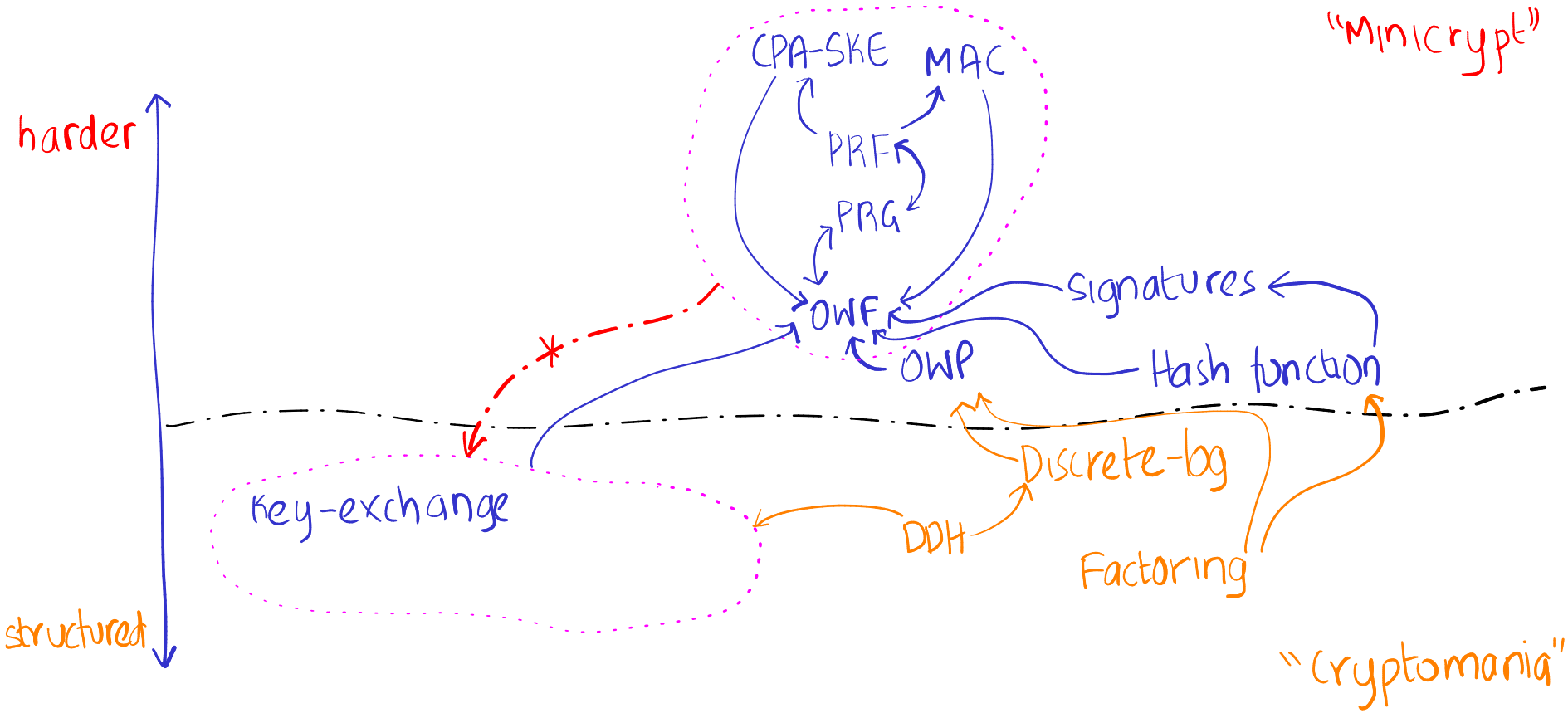
- Task 3: sharing key in presence of eavesdroppers
  - Modelled key exchange setting and security




  - Diffie-Hellman key exchange (DHKE) protocol
    - Basic introduction to groups
    - Based security on the DDH assumption in (prime-order) groups
  - Looked at multi-instance DHKE
    - First proof using hybrid argument
    - Second proof beats hybrid argument via *random self-reducibility*
  - Takeaway: structure vs. hardness
    - Some groups have *sufficient structure* while *remaining hard*

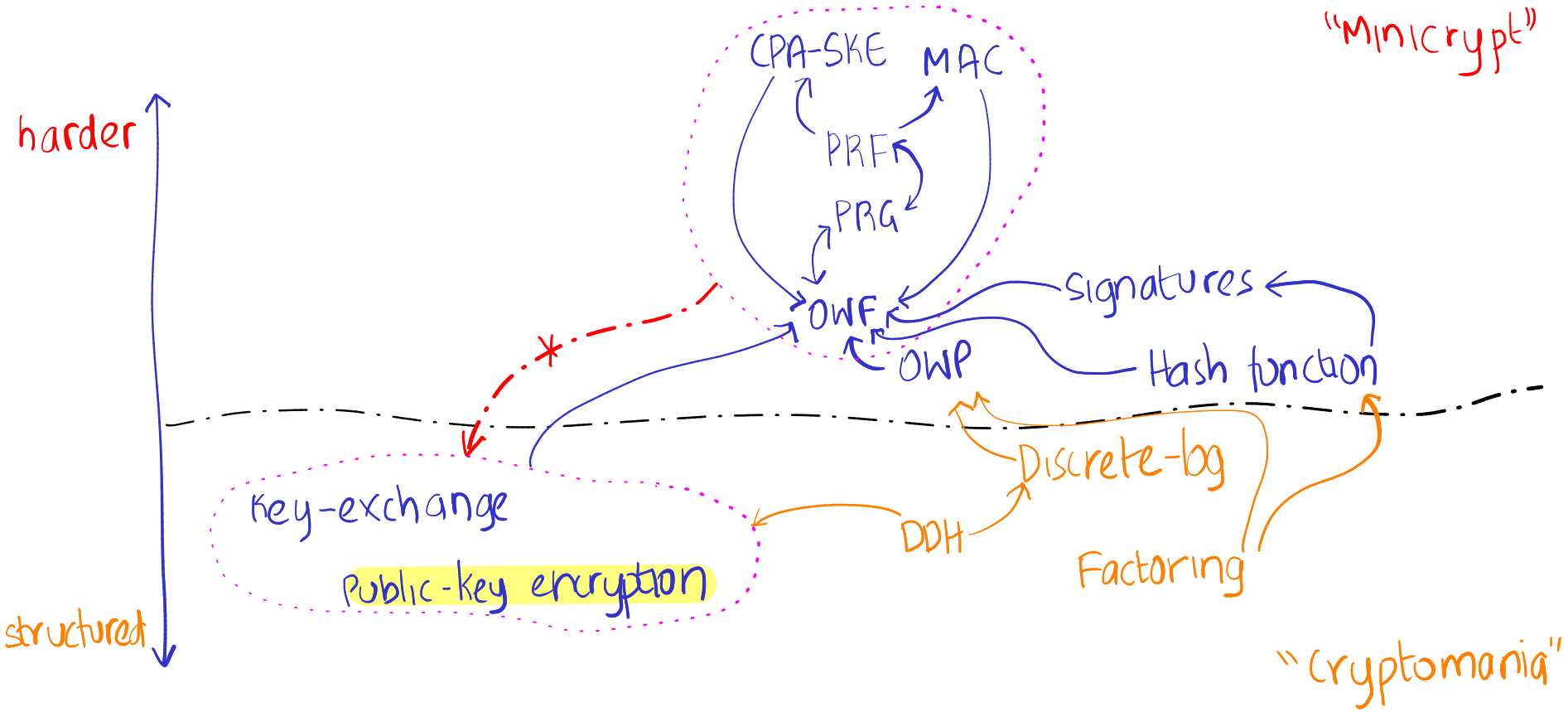
# Plan for This Lecture...

- Minicrypt to Cryptomania



# Plan for This Lecture...

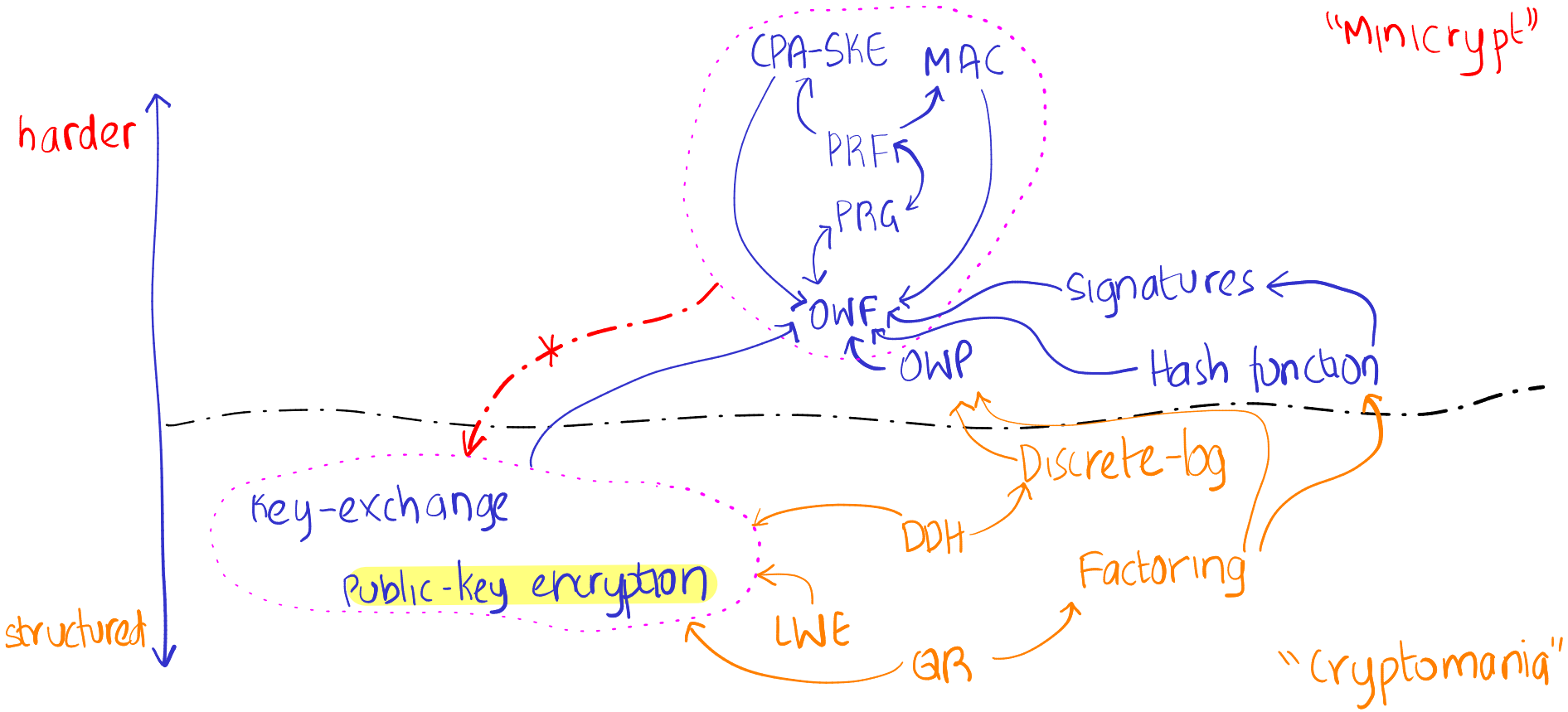
- Minicrypt to Cryptomania




- Today we focus on Task 5: encryption using *public keys*

# Plan for This Lecture...

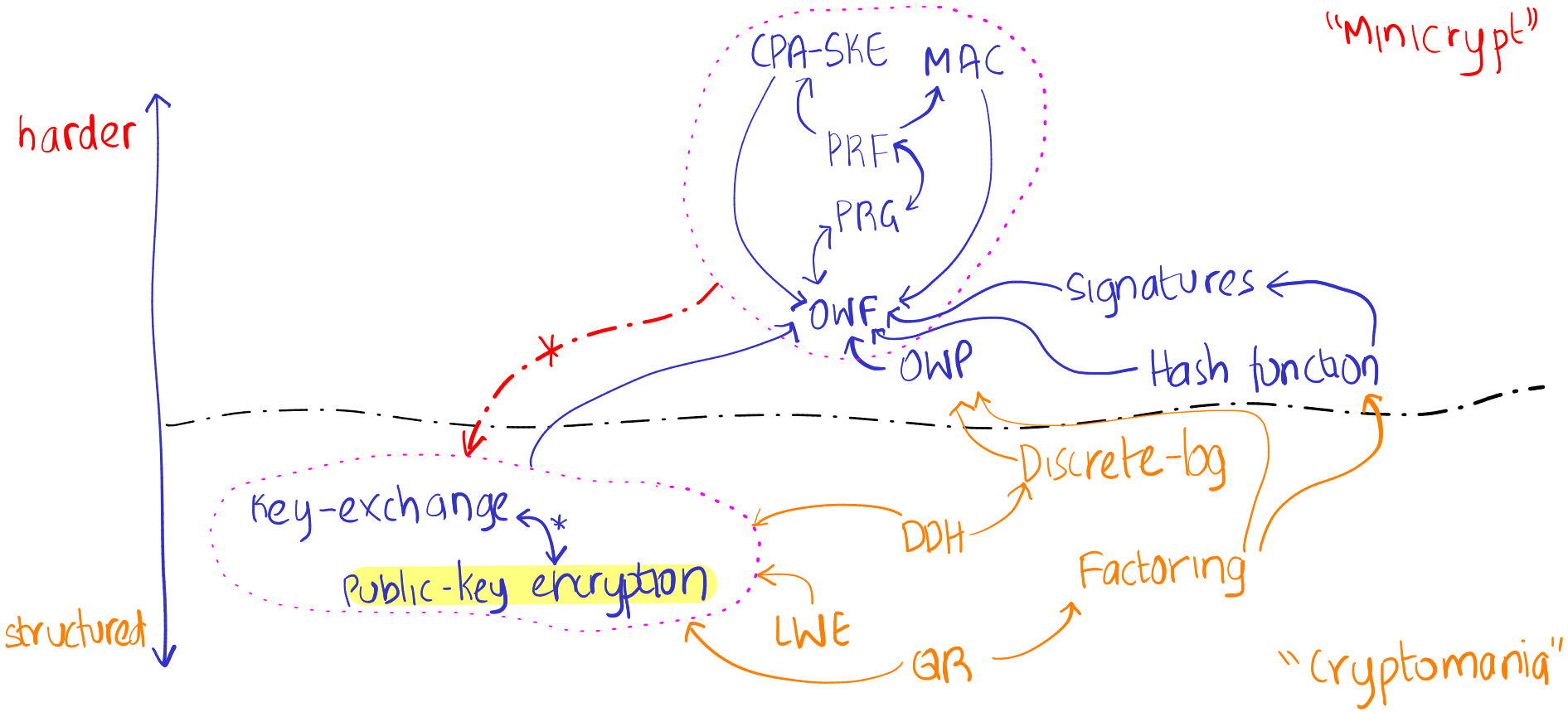
- Minicrypt to Cryptomania




- Today we focus on Task 5: encryption using *public keys*

# Plan for This Lecture...

- Minicrypt to Cryptomania




- Today we focus on Task 5: encryption using *public keys*

# Plan for This Lecture...

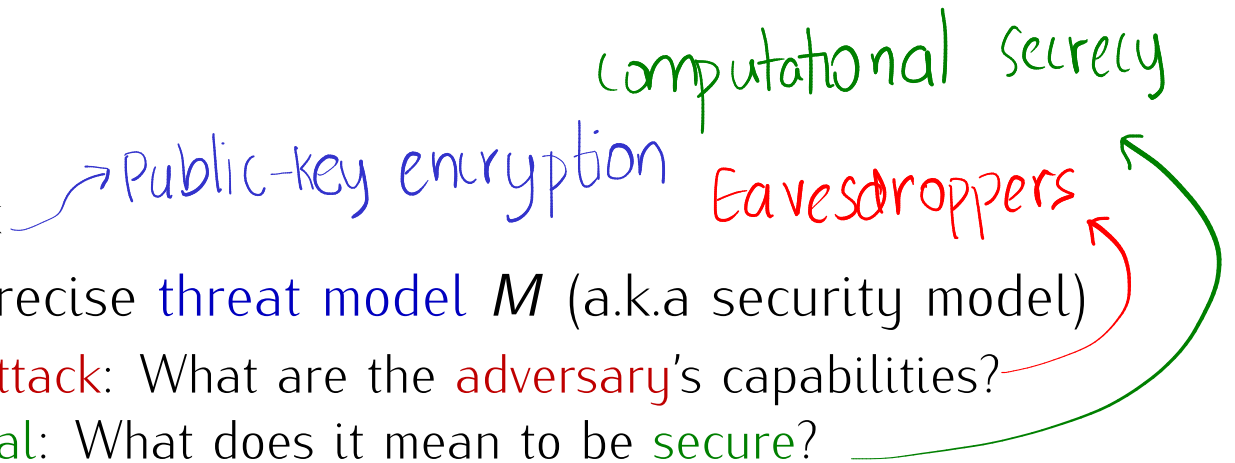
General *template*:

1. Identify the task → Public-key encryption
2. Come up with precise threat model $M$ (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities?
   - Security Goal: What does it mean to be secure?
3. Construct a scheme $\Pi$
4. Formally prove that $\Pi$ in secure in model $M$

# Plan for This Lecture...

General *template*:

1. Identify the task
2. Come up with precise threat model $M$ (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities?
   - Security Goal: What does it mean to be secure?
3. Construct a scheme $\Pi$
4. Formally prove that $\Pi$ in secure in model $M$



# Plan for This Lecture...

General *template*:

1. Identify the task → Public-key encryption
2. Come up with precise threat model $M$ (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities? → Eavesdroppers
   - Security Goal: What does it mean to be secure? → computational secrecy
3. Construct a scheme $\Pi$ 1) ElGamal PKE 2) Goldwasser-Micali PKE
4. Formally prove that $\Pi$ in secure in model $M$

# Plan for This Lecture...

General *template*:

1. Identify the task → Public-key encryption
2. Come up with precise threat model $M$ (a.k.a security model)
   - Adversary/Attack: What are the adversary's capabilities? → Eavesdroppers
   - Security Goal: What does it mean to be secure? → computational secrecy
3. Construct a scheme $\Pi$ 1) ElGamal PKE 2) Goldwasser-Micali PKE
4. Formally prove that $\Pi$ in secure in model $M$ ← 1) Assuming DDH 2) Assuming hardness of "quadratic residuosity" (QR) problem

# Plan for This Lecture...

1. Public-Key Encryption (PKE)
2. ElGamal PKE ← DDH
3. Goldwasser-Micali PKE ← QR

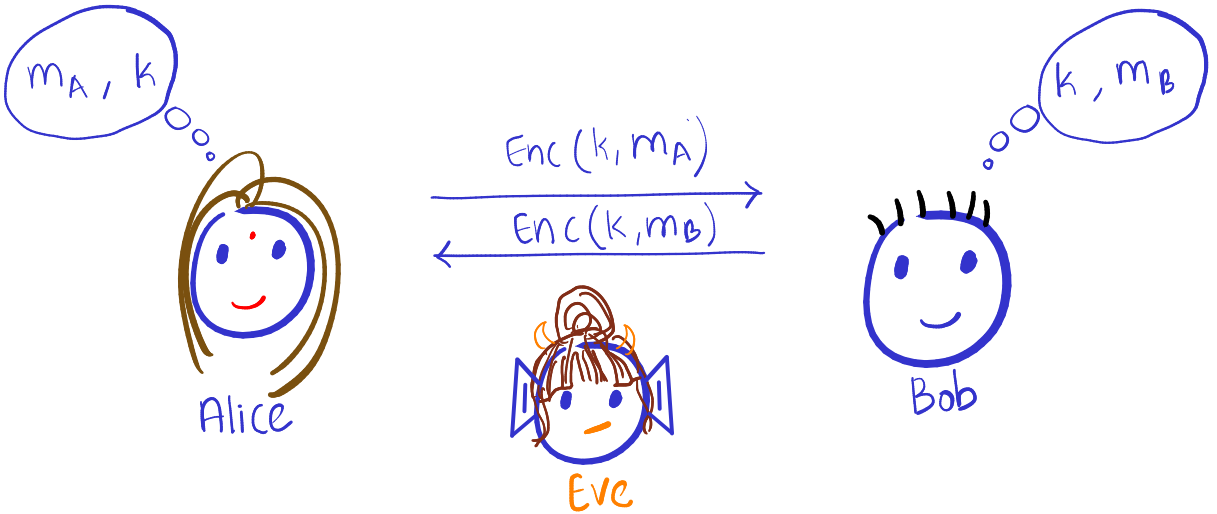
# Plan for This Lecture...

1. Public-Key Encryption (PKE)
2. ElGamal PKE ⟵ DDH
3. Goldwasser-Micali PKE ⟵ QR

# The Setting: Shared (Private) Keys vs Public Keys...


- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve

# The Setting: Shared (Private) Keys vs Public Keys...

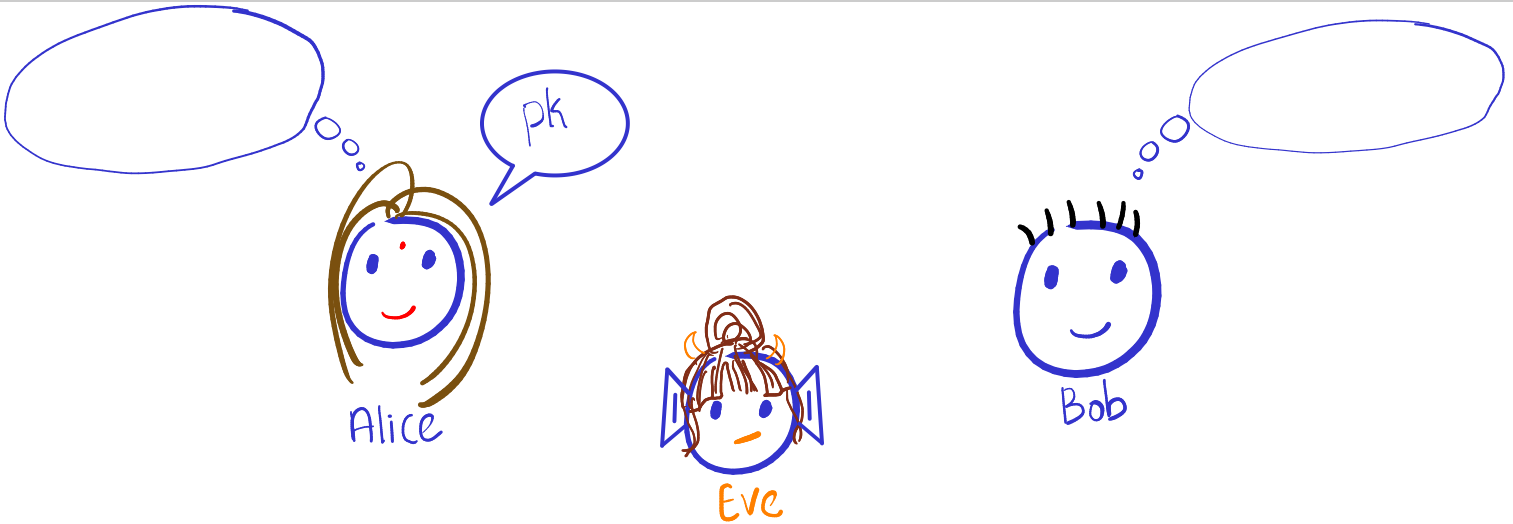



- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve

# The Setting: Shared (Private) Keys vs Public Keys...

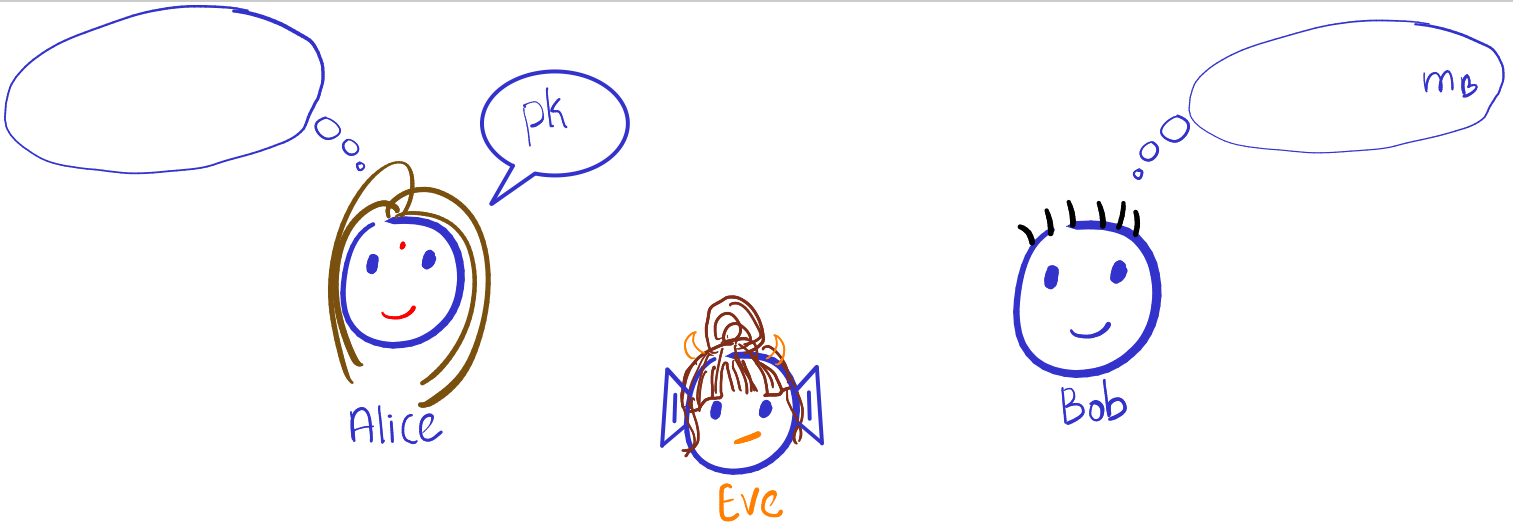

- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key* $pk$; known to *everyone*!
  2. Bob wants to use $pk$ to secretly send a message *to* Alice in presence of Eve

# The Setting: Shared (Private) Keys vs Public Keys...



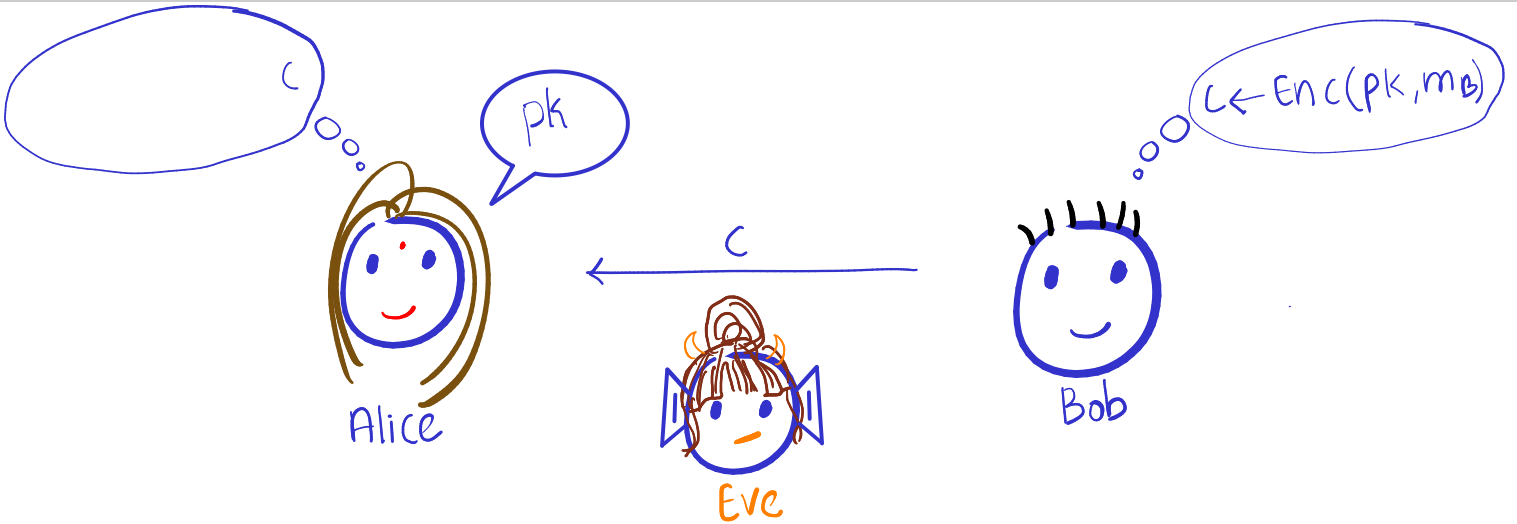

- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key* $pk$; known to *everyone*!
  2. Bob wants to use $pk$ to secretly send a message *to* Alice in presence of Eve

# The Setting: Shared (Private) Keys vs Public Keys...



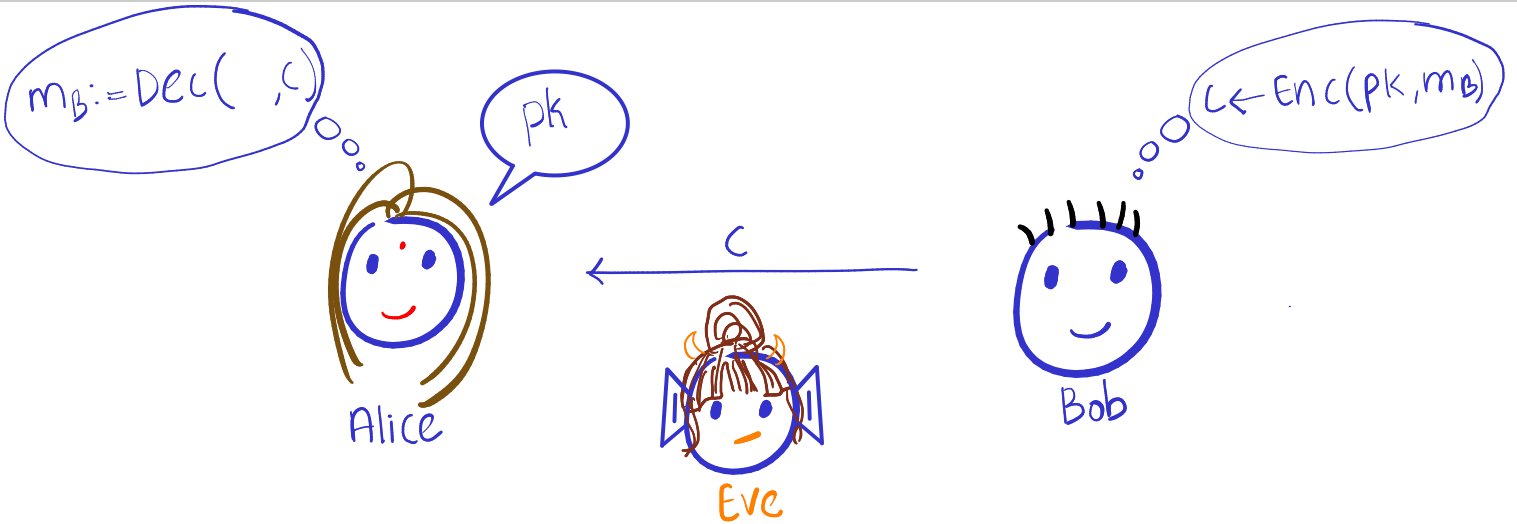

- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key* $pk$; known to *everyone*!
  2. Bob wants to use $pk$ to secretly send a message *to* Alice in presence of Eve

# The Setting: Shared (Private) Keys vs Public Keys...



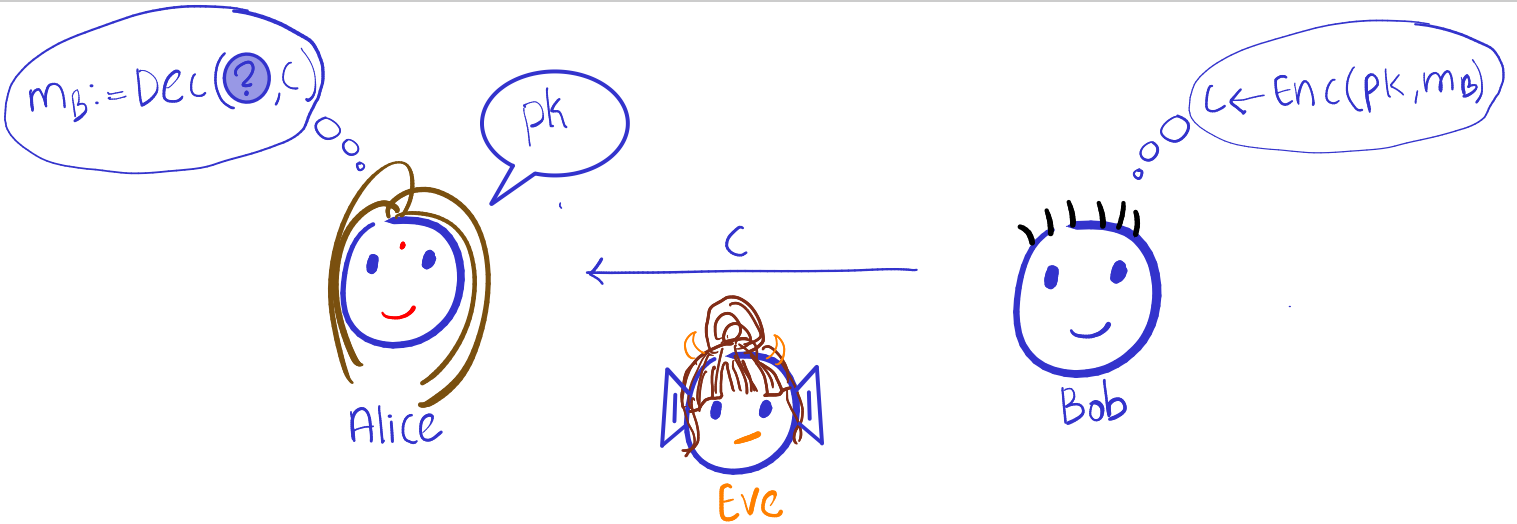

- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key* $pk$; known to *everyone*!
  2. Bob wants to use $pk$ to secretly send a message *to* Alice in presence of Eve

# The Setting: Shared (Private) Keys vs Public Keys...



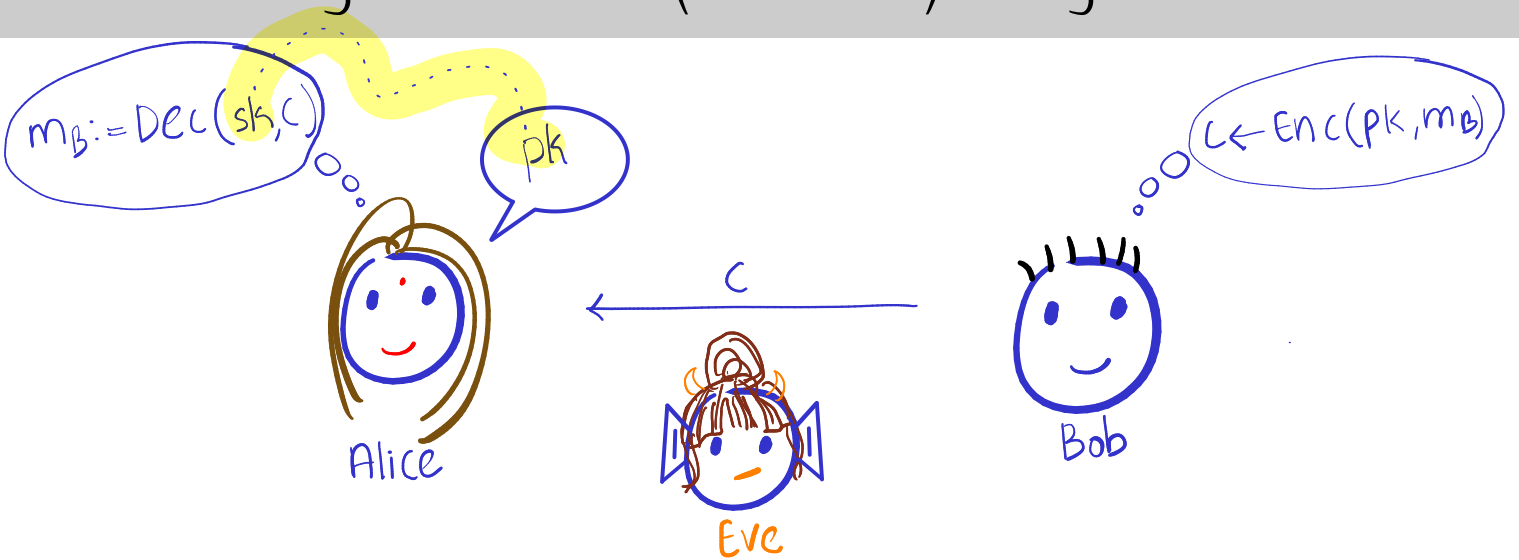

- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key* $pk$; known to *everyone*!
  2. Bob wants to use $pk$ to secretly send a message *to* Alice in presence of Eve

# The Setting: Shared (Private) Keys vs Public Keys...



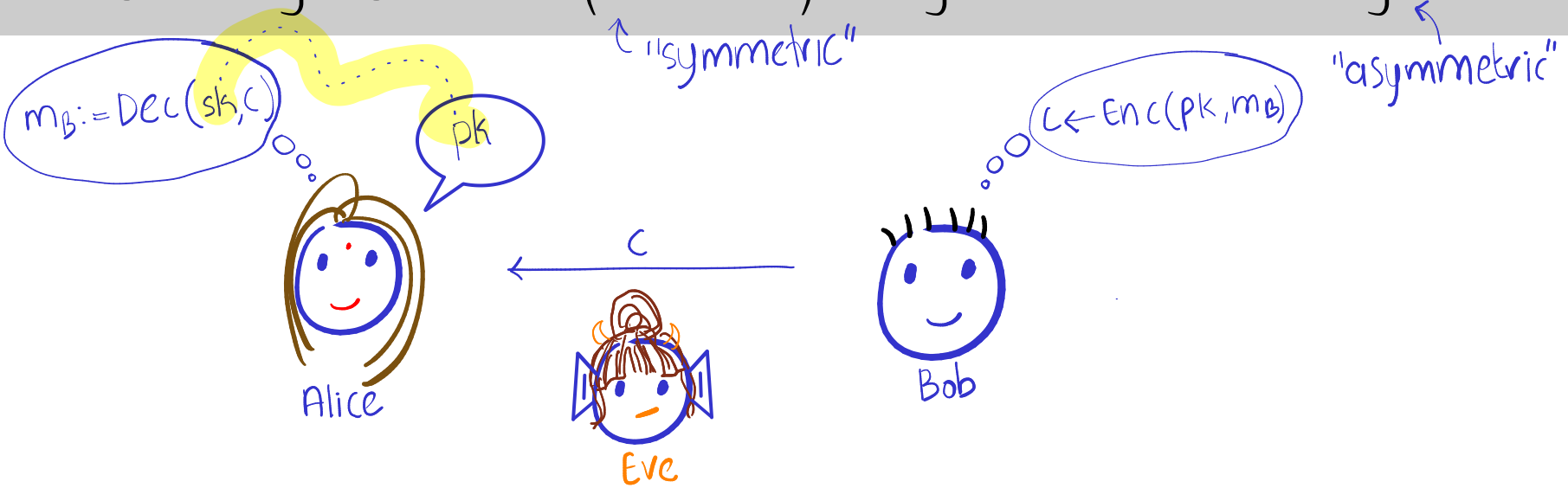

- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key* $pk$; known to *everyone*!
  2. Bob wants to use $pk$ to secretly send a message *to* Alice in presence of Eve

# The Setting: Shared (Private) Keys vs Public Keys...



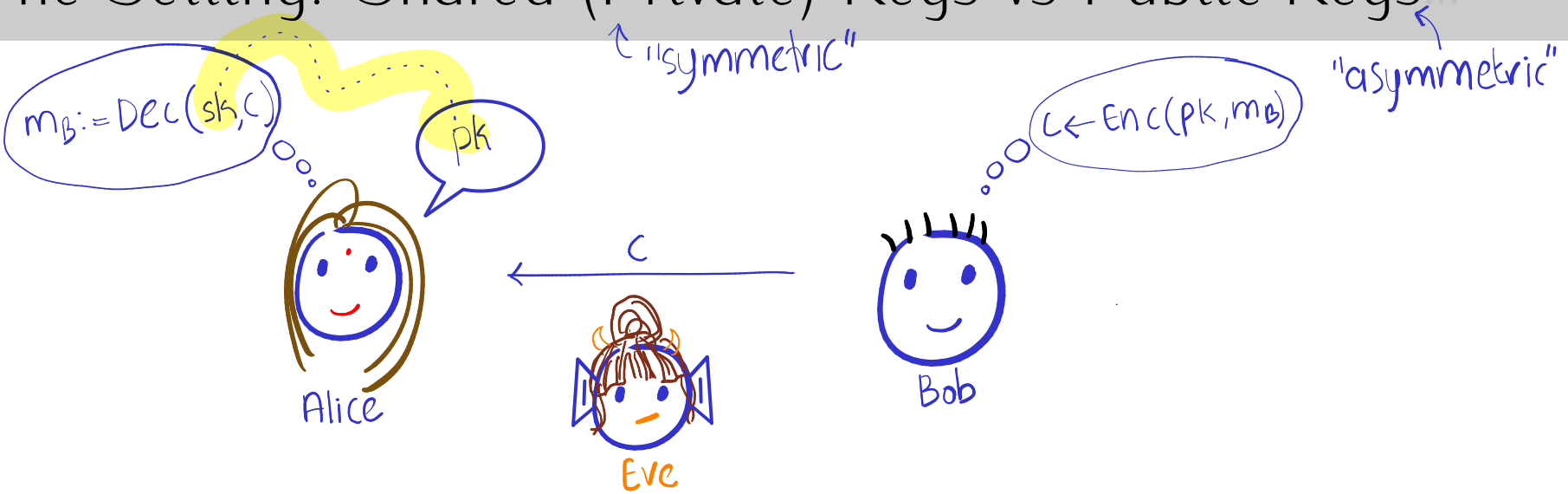

- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key* ***pk***; known to *everyone*!
  2. Bob wants to use ***pk*** to secretly send a message *to* Alice in presence of Eve
  3. Alice decrypts using her *secret key* ***sk*** (related to ***pk***)

# The Setting: Shared (Private) Keys vs Public Keys



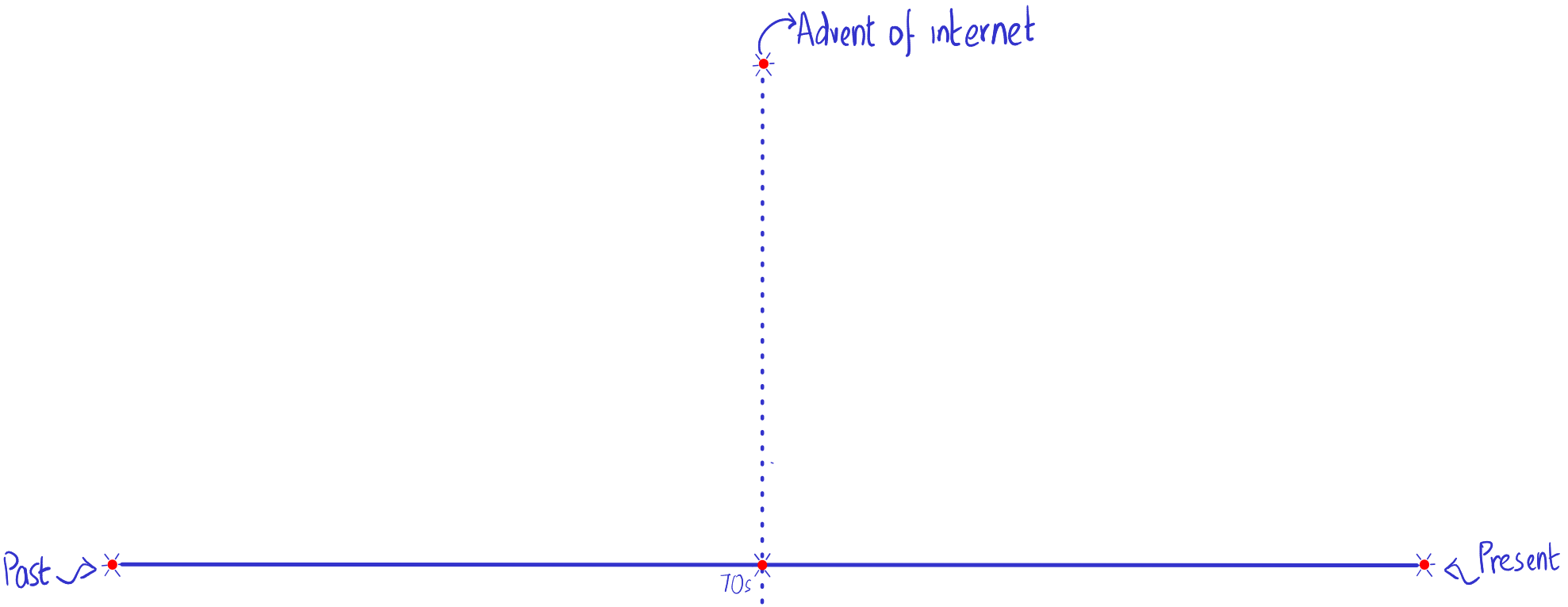

- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key* ***pk***; known to *everyone*!
  2. Bob wants to use ***pk*** to secretly send a message *to* Alice in presence of Eve
  3. Alice decrypts using her *secret key* ***sk*** (related to ***pk***)

# The Setting: Shared (Private) Keys vs Public Keys




- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key* ***pk***; known to *everyone*!
  2. Bob wants to use ***pk*** to secretly send a message *to* Alice in presence of Eve
  3. Alice decrypts using her *secret key* ***sk*** (related to ***pk***)
- Advantage: scalability! It suffices to have one "key" per user

# The Setting: Shared (Private) Keys vs Public Keys...



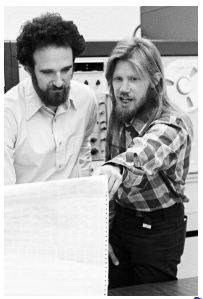

Credit for images: (**icour.fr) (***cs.miami.edu (Rosenberg)) (****Oded Goldreich)

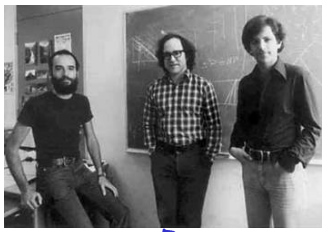
# The Setting: Shared (Private) Keys vs Public Keys...

Programming Techniques — S. L. Graham, R. L. Rivest Editors

**Secure Communications Over Insecure Channels**

Ralph C. Merkle
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley







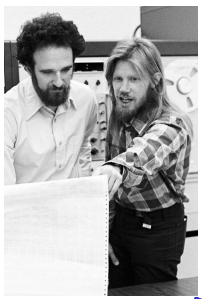

Credit for images: (**icour.fr) (***cs.miami.edu (Rosenberg)) (****Oded Goldreich)

# The Setting: Shared (Private) Keys vs Public Keys...

Programming Techniques — S. L. Graham, R. L. Rivest Editors

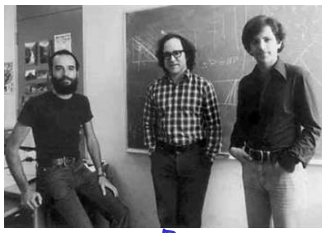
**Secure Communications Over Insecure Channels**

Ralph C. Merkle
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley



**

DHKE



***

RSA cryptosystem

Advent of internet

****

Goldwasser-Micali PKE

Past → 70s 74 76 78 82 ← Present

Credit for images: (**icour.fr) (***cs.miami.edu (Rosenberg)) (****Oded Goldreich)

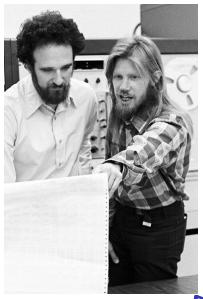
# The Setting: Shared (Private) Keys vs Public Keys...

Programming Techniques | S. L. Graham, R. L. Rivest Editors

**Secure Communications Over Insecure Channels**

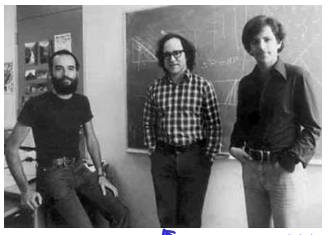
Ralph C. Merkle
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

THE POSSIBILITY OF SECURE NON-SECRET DIGITAL ENCRYPTION
J. H. Ellis, January 1970



**

DHKE



***

RSA cryptosystem

Advent of internet

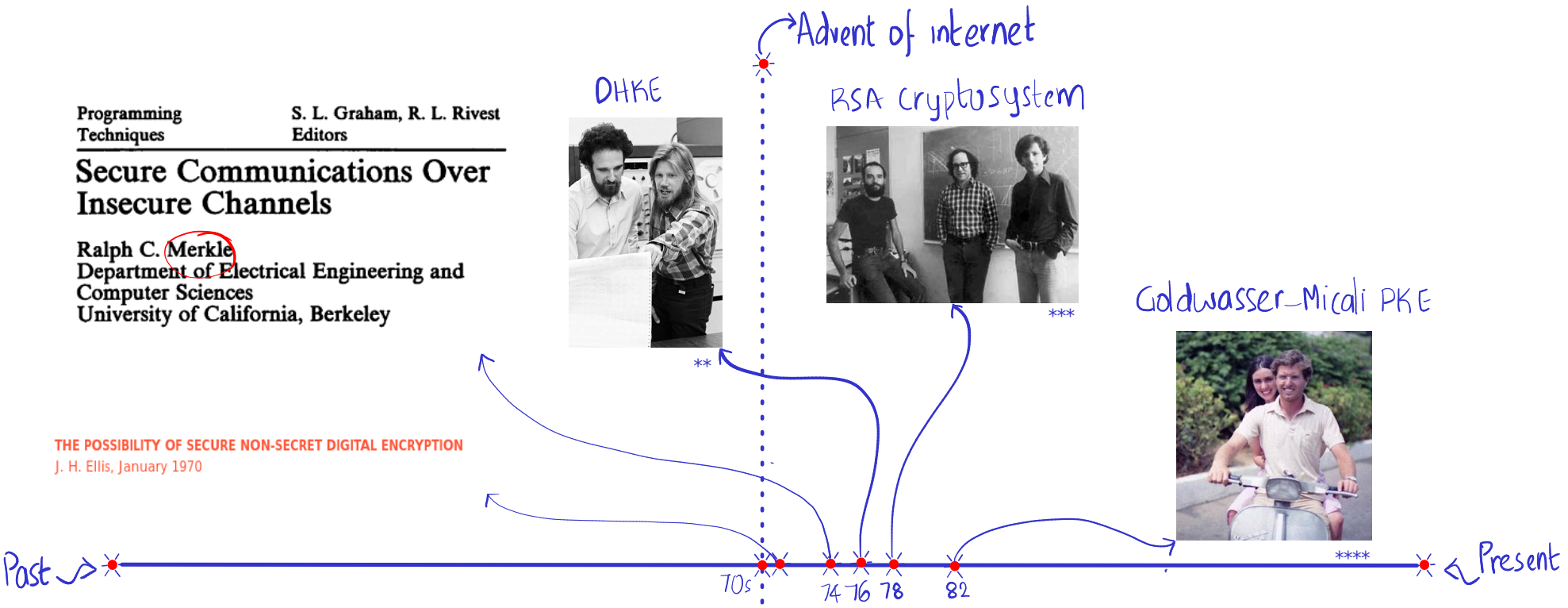
Goldwasser-Micali PKE

****

Past → 70s 74 76 78 82 ← Present

Credit for images: (**icour.fr) (***cs.miami.edu (Rosenberg)) (****Oded Goldreich)

# The Setting: Shared (Private) Keys vs Public Keys...




- PKE IRL: PGP, hybrid encryption

Credit for images: (**icour.fr) (***cs.miami.edu (Rosenberg)) (****Oded Goldreich)

# Syntax of Public-Key Encryption

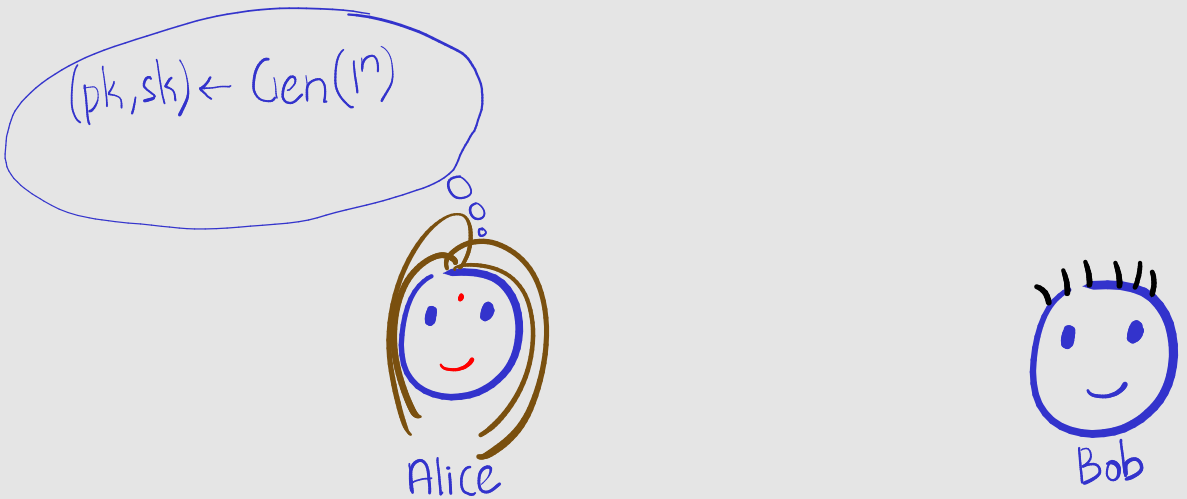
## Defintion 1 (Public-Key Encryption (PKE))

*A PKE* $\Pi$ *is a triple of efficient algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with the following syntax:*


# Syntax of Public-Key Encryption

## Defintion 1 (Public-Key Encryption (PKE))

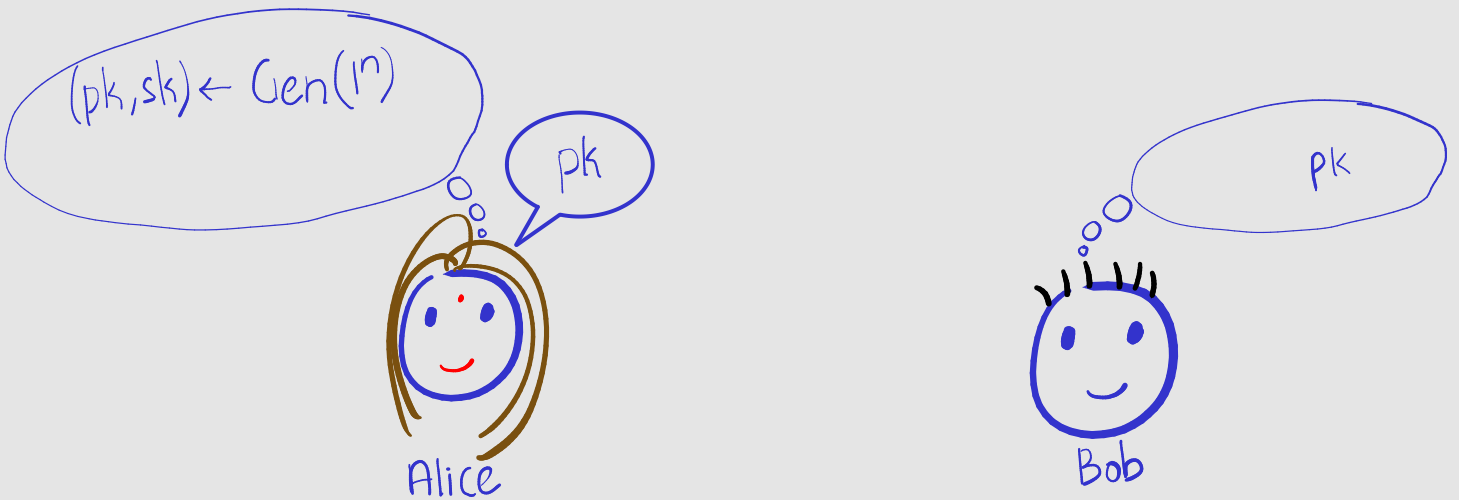
*A PKE* $\Pi$ *is a triple of efficient algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with the following syntax:*



# Syntax of Public-Key Encryption

## Defintion 1 (Public-Key Encryption (PKE))

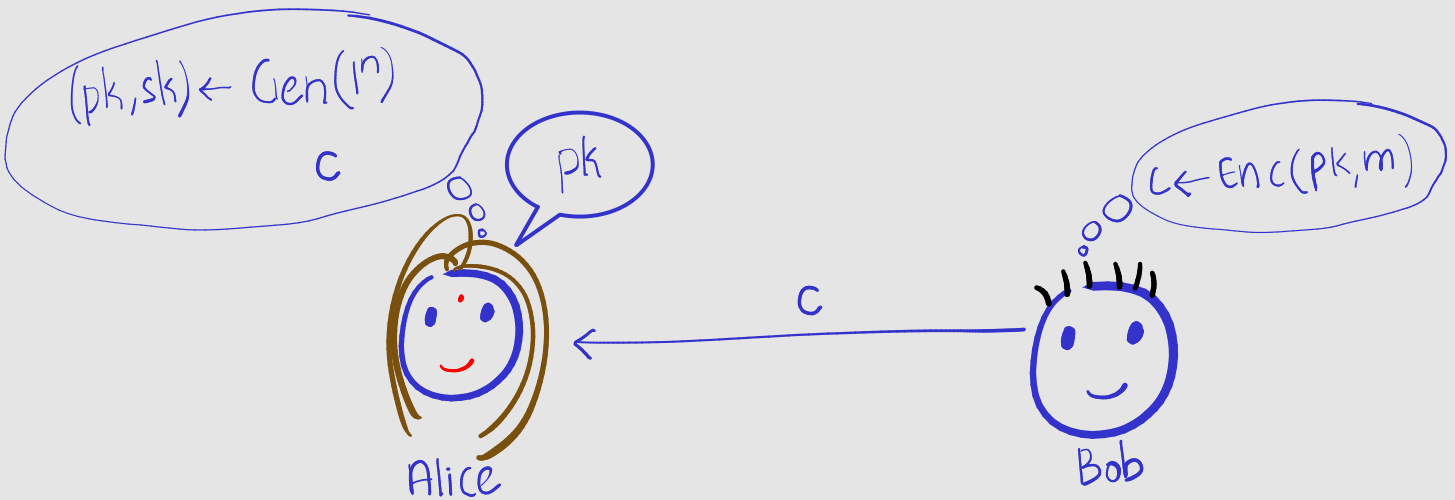
*A PKE* $\Pi$ *is a triple of efficient algorithms* (**Gen**, **Enc**, **Dec**) *with the following syntax:*



# Syntax of Public-Key Encryption

## Defintion 1 (Public-Key Encryption (PKE))

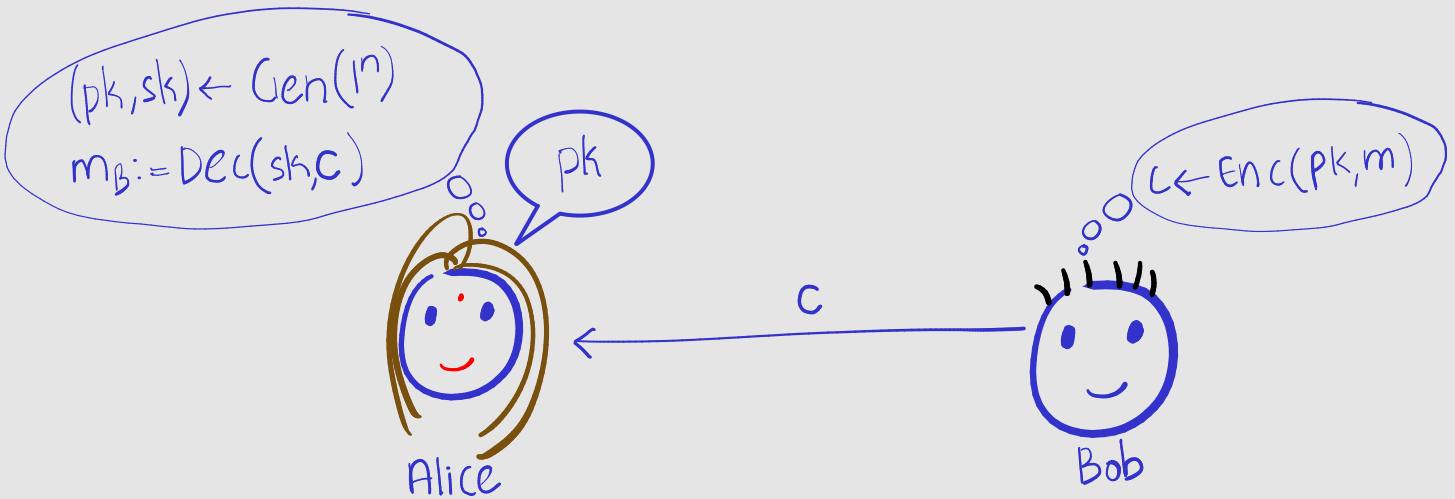
*A PKE* $\Pi$ *is a triple of efficient algorithms* (**Gen**, **Enc**, **Dec**) *with the following syntax:*



# Syntax of Public-Key Encryption

## Defintion 1 (Public-Key Encryption (PKE))

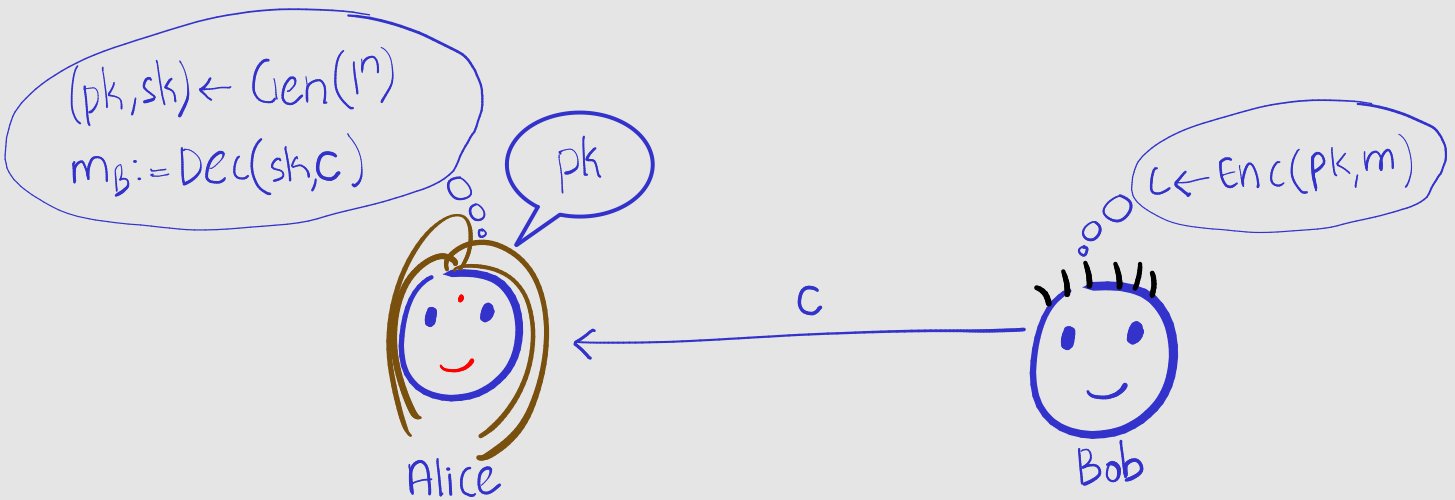
*A PKE* $\Pi$ *is a triple of efficient algorithms* (**Gen**, **Enc**, **Dec**) *with the following syntax:*



# Syntax of Public-Key Encryption

## Defintion 1 (Public-Key Encryption (PKE))

*A PKE* $\Pi$ *is a triple of efficient algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with the following syntax:*



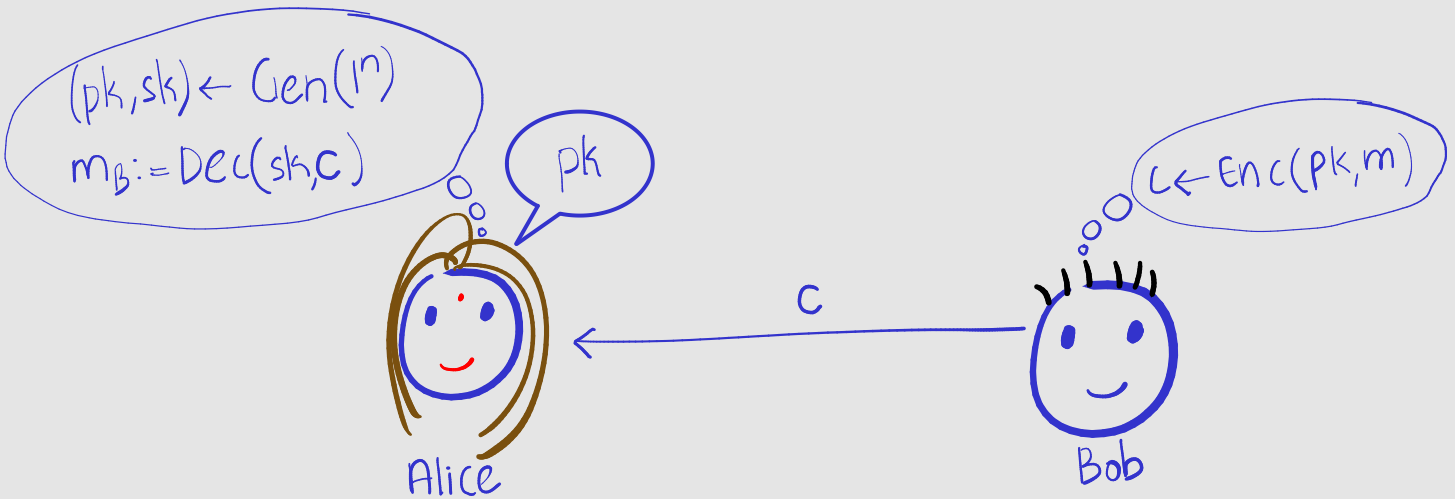

- *Correctness of decryption: for every* $n \in \mathbb{N}$*, message* $m \in \mathcal{M}_n$*,*

$$\Pr_{(pk,sk)\leftarrow \mathsf{Gen}(1^n), c\leftarrow \mathsf{Enc}(pk,m)}[\mathsf{Dec}(sk, c) = m] = 1$$

# Syntax of Public-Key Encryption

## Defintion 1 (Public-Key Encryption (PKE))

*A PKE* $\Pi$ *is a triple of efficient algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with the following syntax:*



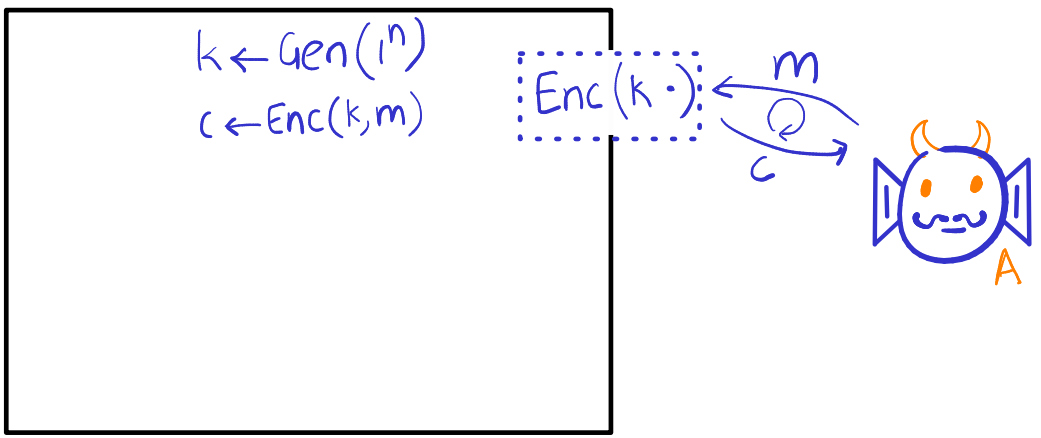

- *Correctness of decryption: for every* $n \in \mathbb{N}$*, message* $m \in \mathcal{M}_n$*,*

$$\Pr_{(pk,sk)\leftarrow\mathsf{Gen}(1^n),c\leftarrow\mathsf{Enc}(pk,m)}[\mathsf{Dec}(sk,c)=m]=1$$

How can an unbounded eavesdropper Eve break PKE?

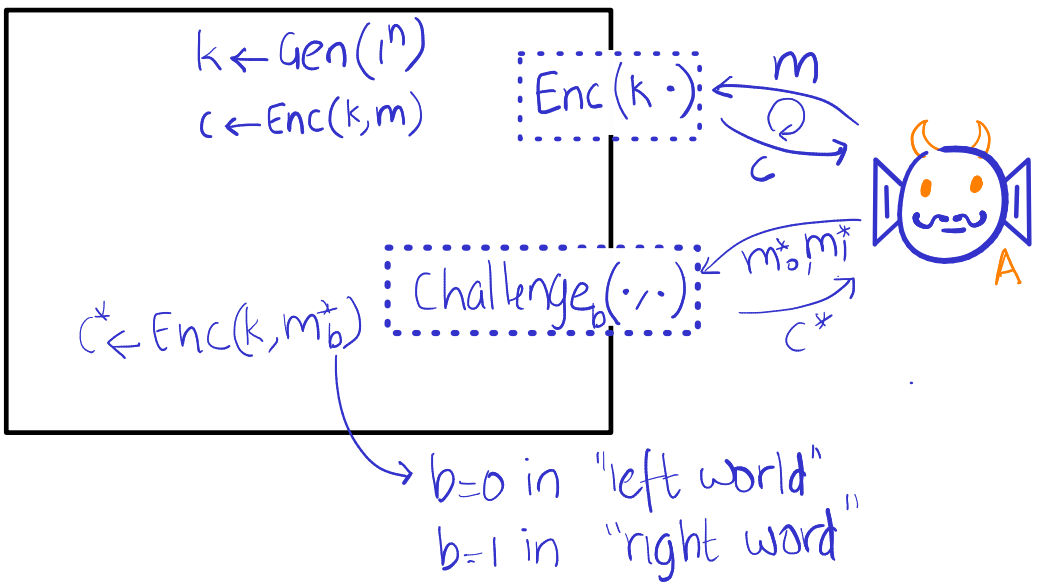
# How to Define Security?...

- Recall CPA-secrecy requirement in the SKE setting



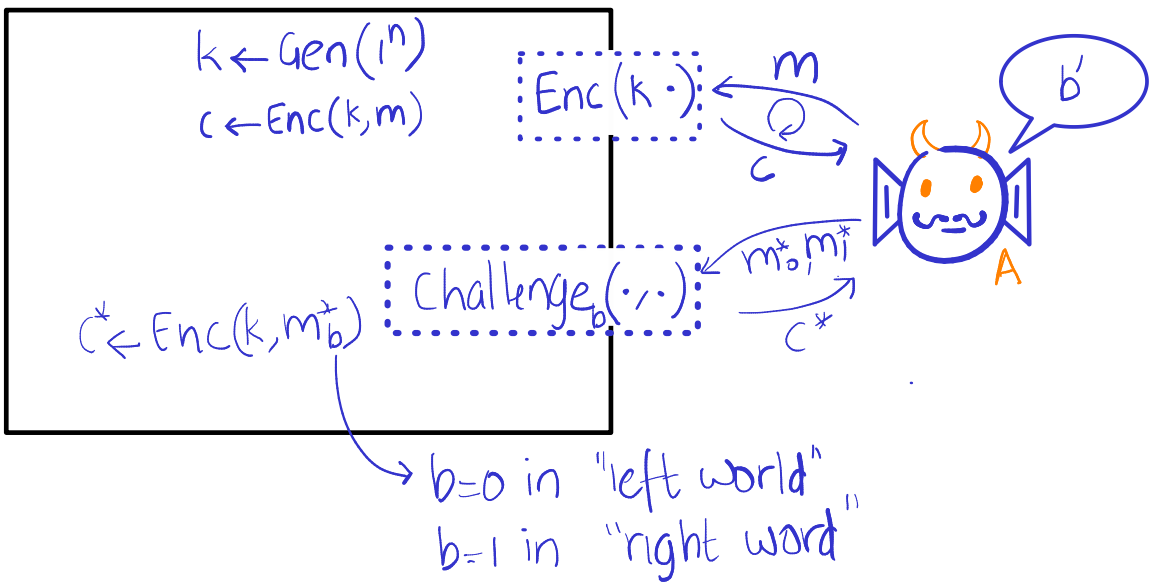
# How to Define Security?...

- Recall CPA-secrecy requirement in the SKE setting



# How to Define Security?...

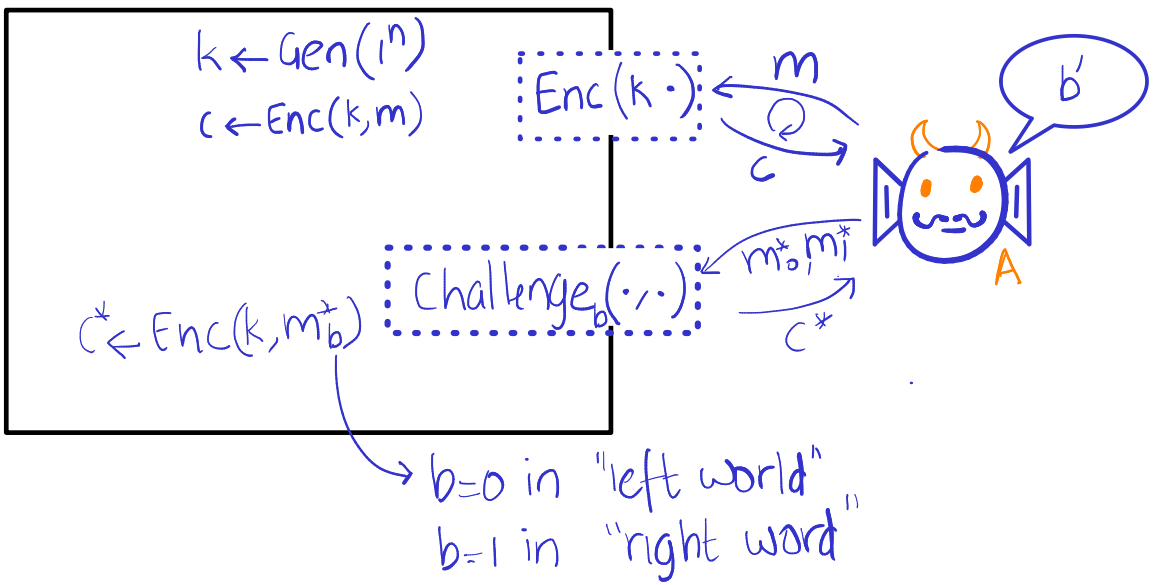
- Recall CPA-secrecy requirement in the SKE setting



# How to Define Security?...

- Recall CPA-secrecy requirement in the SKE setting

(?) What is different in the PKE setting?

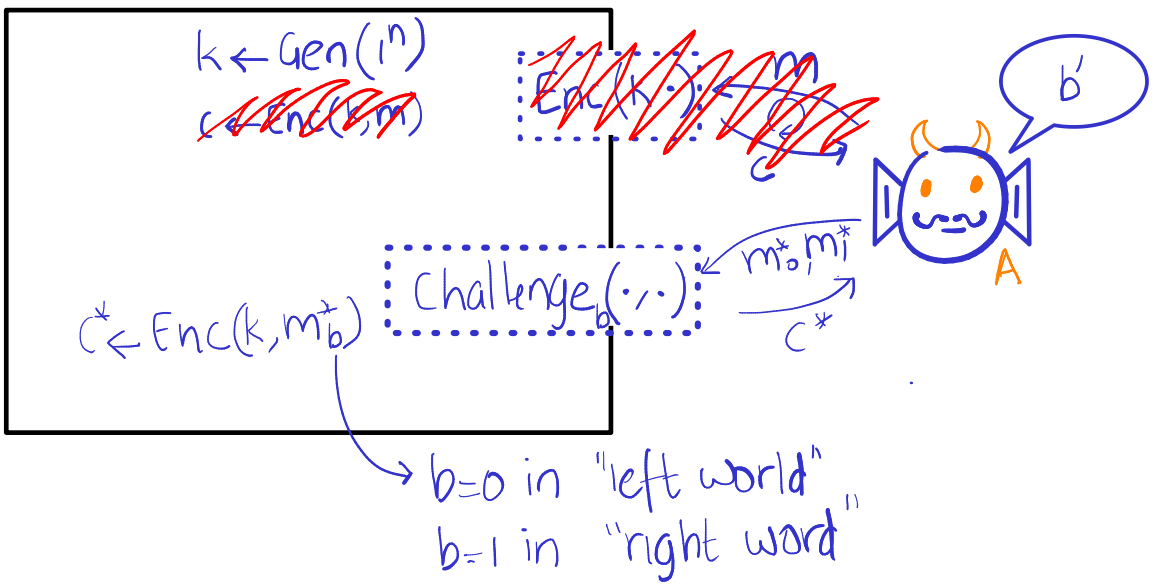


# How to Define Security?...

- Recall CPA-secrecy requirement in the SKE setting

(?) What is different in the PKE setting?

  - The public key known to **Eve** $\Rightarrow$ encryption oracle "redundant"



# How to Define Security?...

- Recall CPA-secrecy requirement in the SKE setting
- What is different in the PKE setting?
  - The public key known to **Eve** $\Rightarrow$ encryption oracle "redundant"
  - Eavesdropper=chosen-plaintext attacker!

## Defintion 2 (CPA Secrecy for PKE)

*A PKE* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is CPA-secret if for every PPT eavesdropper Eve, the following is negligible:*

$$\delta(n) := \left| \Pr_{\substack{(pk,sk)\leftarrow \mathsf{Gen}(1^n) \\ (m_0,m_1)\leftarrow \mathsf{Eve}(pk) \\ c\leftarrow \mathsf{Enc}(pk,m_0)}} [\mathsf{Eve}(c) = 0] - \Pr_{\substack{(pk,sk)\leftarrow \mathsf{Gen}(1^n) \\ (m_0,m_1)\leftarrow \mathsf{Eve}(pk) \\ c\leftarrow \mathsf{Enc}(pk,m_1)}} [\mathsf{Eve}(c) = 0] \right|$$

# How to Define Security?...

- Recall CPA-secrecy requirement in the SKE setting
- What is different in the PKE setting?
  - The public key known to **Eve** $\Rightarrow$ encryption oracle "redundant"
  - Eavesdropper=chosen-plaintext attacker!

## Defintion 2 (CPA Secrecy for PKE)

*A PKE* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is CPA-secret if for every PPT eavesdropper Eve, the following is negligible:*

$$\delta(n) := \left| \Pr_{\substack{(pk,sk)\leftarrow \mathsf{Gen}(1^n) \\ (m_0,m_1)\leftarrow \mathsf{Eve}(pk) \\ c\leftarrow \mathsf{Enc}(pk,m_0)}} [\mathsf{Eve}(c) = 0] - \Pr_{\substack{(pk,sk)\leftarrow \mathsf{Gen}(1^n) \\ (m_0,m_1)\leftarrow \mathsf{Eve}(pk) \\ c\leftarrow \mathsf{Enc}(pk,m_1)}} [\mathsf{Eve}(c) = 0] \right|$$

"Left world" "Right world"

# How to Define Security?...

- Recall CPA-secrecy requirement in the SKE setting

❓ What is different in the PKE setting?

- The public key known to **Eve** ⇒ encryption oracle "redundant"
- Eavesdropper=chosen-plaintext attacker!

## Defintion 2 (CPA Secrecy for PKE)

*A PKE* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is CPA-secret if for every PPT eavesdropper Eve, the following is negligible:*

$$\delta(n) := \left| \Pr_{\substack{(pk,sk)\leftarrow \mathsf{Gen}(1^n) \\ (m_0,m_1)\leftarrow \mathsf{Eve}(pk) \\ c\leftarrow \mathsf{Enc}(pk,m_0)}} [\mathsf{Eve}(c) = 0] - \Pr_{\substack{(pk,sk)\leftarrow \mathsf{Gen}(1^n) \\ (m_0,m_1)\leftarrow \mathsf{Eve}(pk) \\ c\leftarrow \mathsf{Enc}(pk,m_1)}} [\mathsf{Eve}(c) = 0] \right|$$

"Left world" "Right world"

≡ Alternative, equivalent notion: semantic security

- Ciphertext doesn't leak (non-trivial) information about plaintext

# How to Define Security?...

- Recall CPA-secrecy requirement in the SKE setting

(?) What is different in the PKE setting?

- The public key known to **Eve** $\Rightarrow$ encryption oracle "redundant"
- Eavesdropper=chosen-plaintext attacker!

## Defintion 2 (CPA Secrecy for PKE)

*A PKE* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is CPA-secret if for every PPT eavesdropper Eve, the following is negligible:*

$$\delta(n) := \left| \Pr_{\substack{(pk,sk)\leftarrow \mathsf{Gen}(1^n) \\ (m_0,m_1)\leftarrow \mathsf{Eve}(pk) \\ c\leftarrow \mathsf{Enc}(pk,m_0)}} [\mathsf{Eve}(c) = 0] - \Pr_{\substack{(pk,sk)\leftarrow \mathsf{Gen}(1^n) \\ (m_0,m_1)\leftarrow \mathsf{Eve}(pk) \\ c\leftarrow \mathsf{Enc}(pk,m_1)}} [\mathsf{Eve}(c) = 0] \right|$$

"Left world" "Right world"

$\equiv$ Alternative, equivalent notion: semantic security

- Ciphertext doesn't leak (non-trivial) information about plaintext

+1 Stronger notion: secrecy against chosen-ciphertext attacker

# Plan for this Lecture

1. Public-Key Encryption (PKE)
2. ElGamal PKE ⟵ DDH
3. Goldwasser-Micali PKE ⟵ QR

# Recall Group Theory from Last Lecture...

**Defintion 3 (Group axioms)**

*A group $\mathbb{G}$ is a set $\mathcal{G}$ with a binary operation $\cdot$ satisfying: 1) closure 2) associativity, 3) existence of identity and 4) existence of inverse. $\mathbb{G}$ Abelian if it additionally satisfies 5) commutativity.*

# Recall Group Theory from Last Lecture...

**Defintion 3 (Group axioms)**

*A group $\mathbb{G}$ is a set $\mathcal{G}$ with a binary operation $\cdot$ satisfying: 1) closure 2) associativity, 3) existence of identity and 4) existence of inverse. $\mathbb{G}$ Abelian if it additionally satisfies 5) commutativity.*

**Defintion 4 (Group terminology)**

- *Order of the group:* $|\mathcal{G}|$
- *Order of an element* $g$*: smallest* $\ell$ *such that* $g^\ell = 1$
- *Cyclic group: there exists an element* $g \in \mathcal{G}$ *(a "generator") with order* $\ell = |\mathcal{G}|$

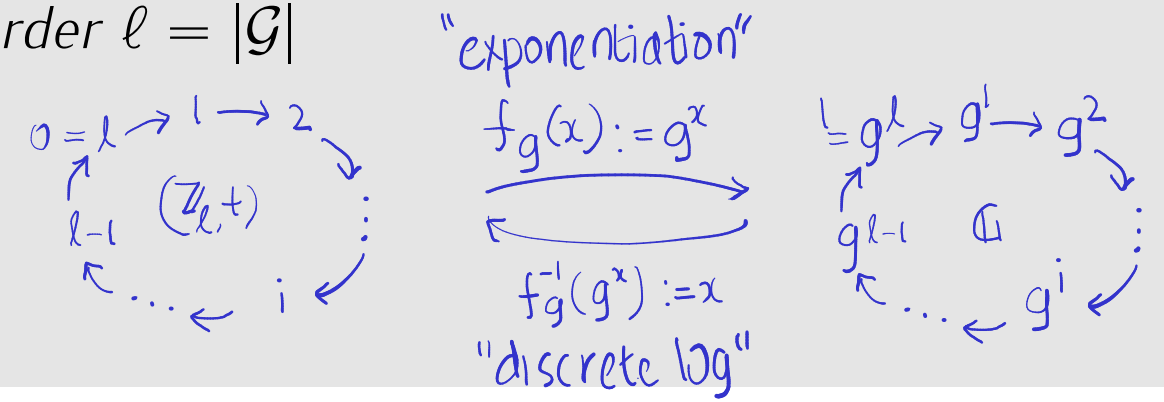
# Recall Group Theory from Last Lecture...

## Defintion 3 (Group axioms)

*A group $\mathbb{G}$ is a set $\mathcal{G}$ with a binary operation $\cdot$ satisfying: 1) closure 2) associativity, 3) existence of identity and 4) existence of inverse. $\mathbb{G}$ Abelian if it additionally satisfies 5) commutativity.*

## Defintion 4 (Group terminology)

- *Order of the group:* $|\mathcal{G}|$
- *Order of an element* $g$*: smallest* $\ell$ *such that* $g^\ell = 1$
- *Cyclic group: there exists an element* $g \in \mathcal{G}$ *(a "generator") with order* $\ell = |\mathcal{G}|$



# Recall Group Theory from Last Lecture...

**Assumption 1 (DLog assumption in $\mathbb{G}$ w.r.to $\mathsf{S}$)**

*The DLog assumption in cyclic group $\mathbb{G}$ w.r.to $\mathsf{S}$ holds if for all PPT inverters $\mathsf{Inv}$, the following is negligible:*

$$\delta(n) := \Pr_{\substack{(\mathbb{G},\ell,g)\leftarrow\mathsf{S}(1^n)\\ a\leftarrow\mathbb{Z}_\ell}}[\mathsf{Inv}((\mathbb{G},\ell,g),g^a)=a]$$

# Recall Group Theory from Last Lecture...

## Assumption 1 (DLog assumption in $\mathbb{G}$ w.r.to $\mathsf{S}$)

*The DLog assumption in cyclic group $\mathbb{G}$ w.r.to $\mathsf{S}$ holds if for all PPT inverters $\mathsf{Inv}$, the following is negligible:*

$$\delta(n) := \Pr_{\substack{(\mathbb{G},\ell,g)\leftarrow\mathsf{S}(1^n)\\ a\leftarrow\mathbb{Z}_\ell}}[\mathsf{Inv}((\mathbb{G},\ell,g),g^a)=a]$$

## Assumption 2 (DDH assumption in in $\mathbb{G}$ w.r.to $\mathsf{S}$)

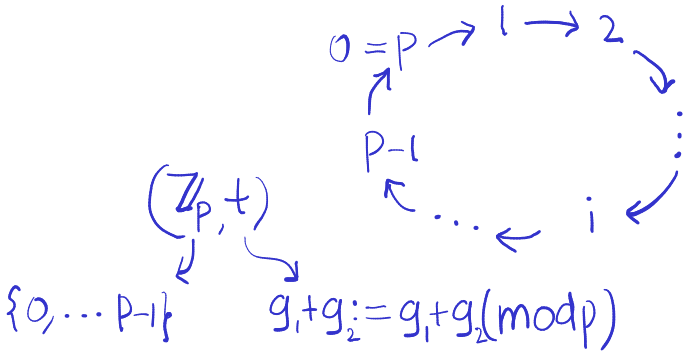
*The DDH assumption holds in cyclic group $\mathbb{G}$ w.r.to $\mathsf{S}$ if for all PPT distinguishers $\mathsf{D}$, the following is negligible:*

$$\left|\Pr_{\substack{(\mathbb{G},\ell,g)\leftarrow\mathsf{S}(1^n)\\ a,b\leftarrow\mathbb{Z}_\ell}}[\mathsf{D}(g^a,g^b,g^{ab})=0]-\Pr_{\substack{(\mathbb{G},\ell,g)\leftarrow\mathsf{S}(1^n)\\ a,b,r\leftarrow\mathbb{Z}_\ell}}[\mathsf{D}(g^a,g^b,g^{r})=0]\right|$$

"real/pseudorandom" "random"

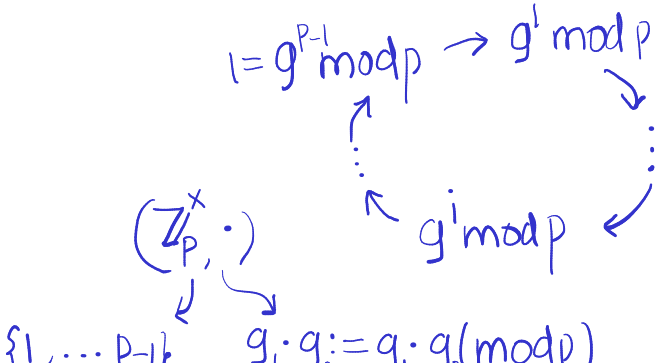
# Recall Group Theory from Last Lecture...

## Addition modulo prime $p$



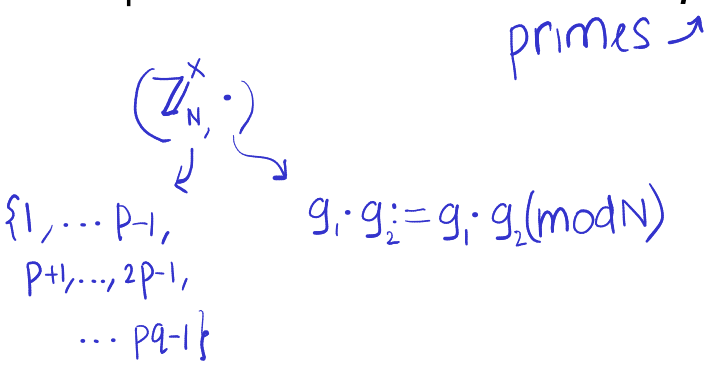

- order $p$
- cyclic

## Multiplication modulo prime $p$



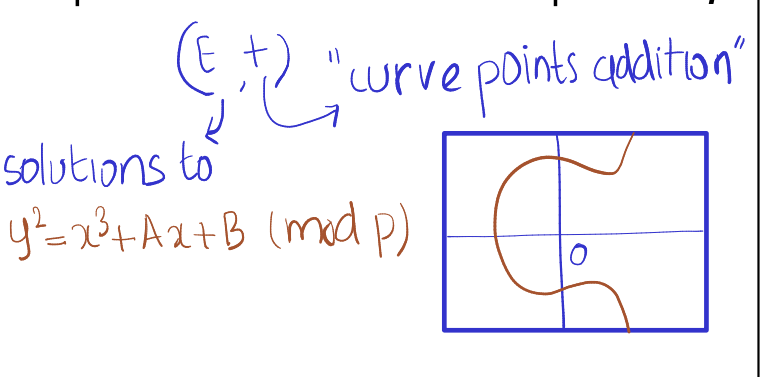

- order $p-1$
- cyclic

## Multiplication modulo $N = pq$




- order $(p-1)(q-1)$
- not cyclic

## Elliptic curves modulo prime $p$

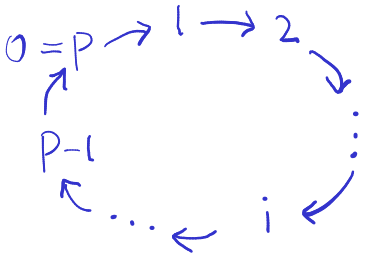



- $|p+1 - \text{order}| \leq 2\sqrt{p}$
- cyclic

# Recall Group Theory from Last Lecture...

DLog easy

## Addition modulo prime $p$



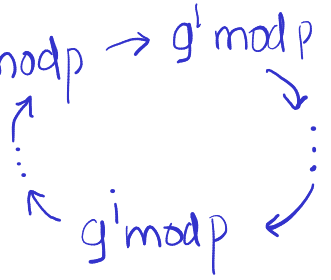

$(\mathbb{Z}_p, +)$

$\{0, \dots p-1\}$ $\quad g_1 + g_2 := g_1 + g_2 \pmod p$

- order $p$
- cyclic

DLog hard, but DDH easy

## Multiplication modulo prime $p$




$(\mathbb{Z}_p^\times, \cdot)$

$\{1, \dots p-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod p$

- order $p-1$
- cyclic

## Multiplication modulo $N = pq$

primes

$(\mathbb{Z}_N^\times, \cdot)$

$\{1, \dots p-1, p+1, \dots, 2p-1, \dots pq-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod N$

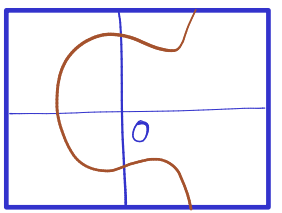
- order $(p-1)(q-1)$
- not cyclic

DDH hard in "subgroup"

## Elliptic curves modulo prime $p$

$(E, +)$ "curve points addition"

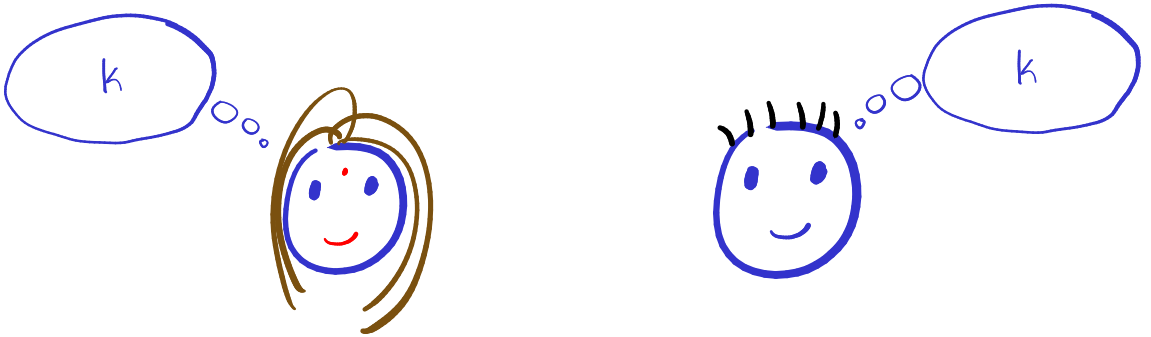
solutions to $y^2 = x^3 + Ax + B \pmod p$




- $|p+1 - \text{order}| \leq 2\sqrt{p}$
- cyclic

DLog very hard, DDH hard

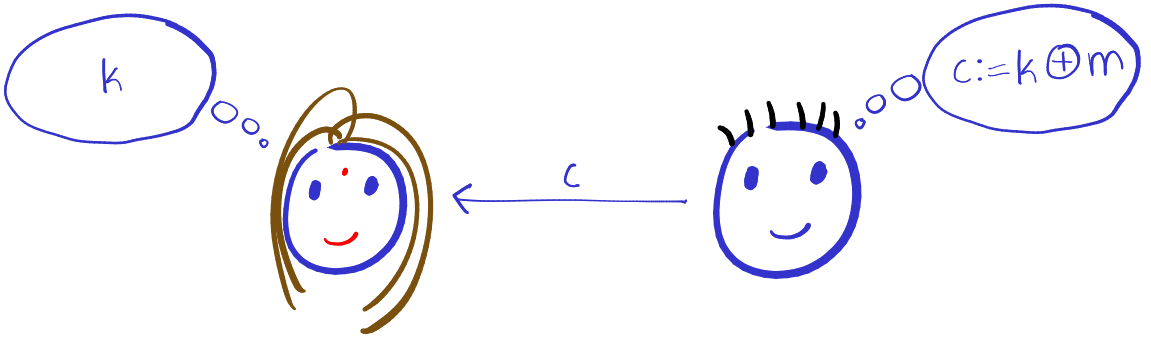
# One-Time Pad Can be Generalised Over Groups




**Pseudocode 1 (OTP over $(\{0,1\}^n, \oplus)$ with message space $\{0,1\}^n$)**

- *Key generation* $\mathsf{Gen}$*: output* $k \leftarrow \{0,1\}^n$
- *Encryption* $\mathsf{Enc}(k, m)$*: output* $c := k \oplus m$
- *Decryption* $\mathsf{Dec}(k, c)$*: output* $m := k \oplus c$

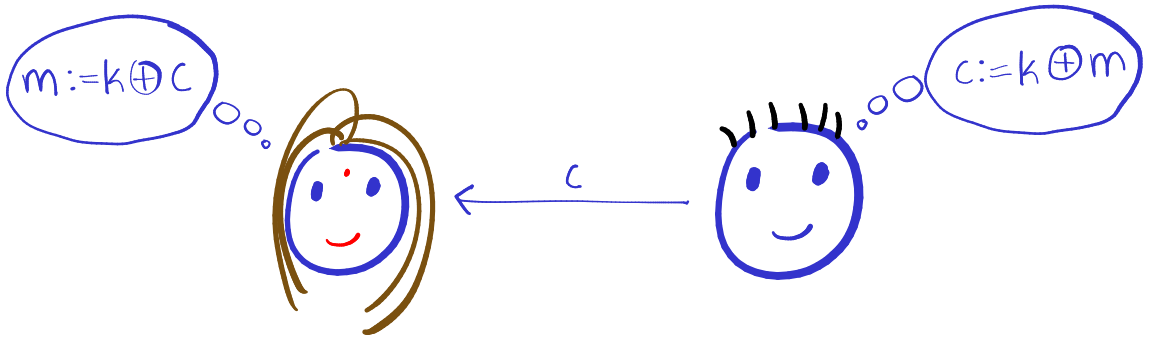
# One-Time Pad Can be Generalised Over Groups




**Pseudocode 1 (OTP over $(\{0,1\}^n, \oplus)$ with message space $\{0,1\}^n$)**

- *Key generation* $\mathsf{Gen}$*: output* $k \leftarrow \{0,1\}^n$
- *Encryption* $\mathsf{Enc}(k, m)$*: output* $c := k \oplus m$
- *Decryption* $\mathsf{Dec}(k, c)$*: output* $m := k \oplus c$

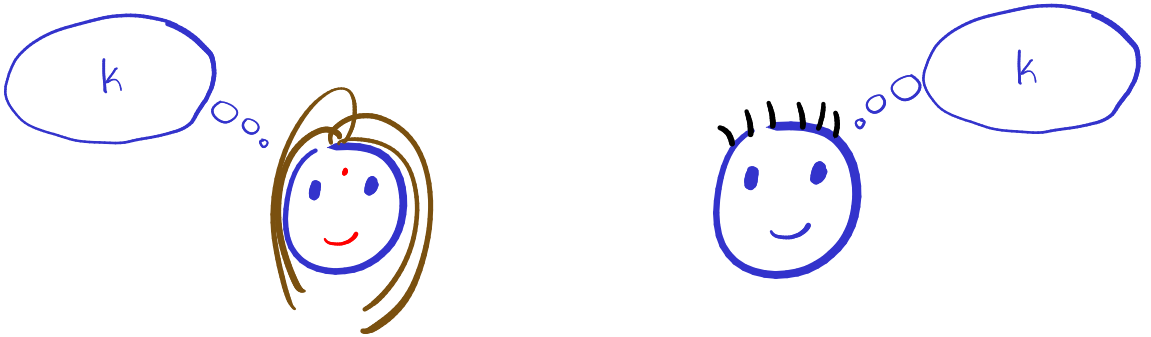
# One-Time Pad Can be Generalised Over Groups




**Pseudocode 1 (OTP over $(\{0,1\}^n, \oplus)$ with message space $\{0,1\}^n$)**

- *Key generation* $\mathsf{Gen}$*: output* $k \leftarrow \{0,1\}^n$
- *Encryption* $\mathsf{Enc}(k, m)$*: output* $c := k \oplus m$
- *Decryption* $\mathsf{Dec}(k, c)$*: output* $m := k \oplus c$

# One-Time Pad Can be Generalised Over Groups



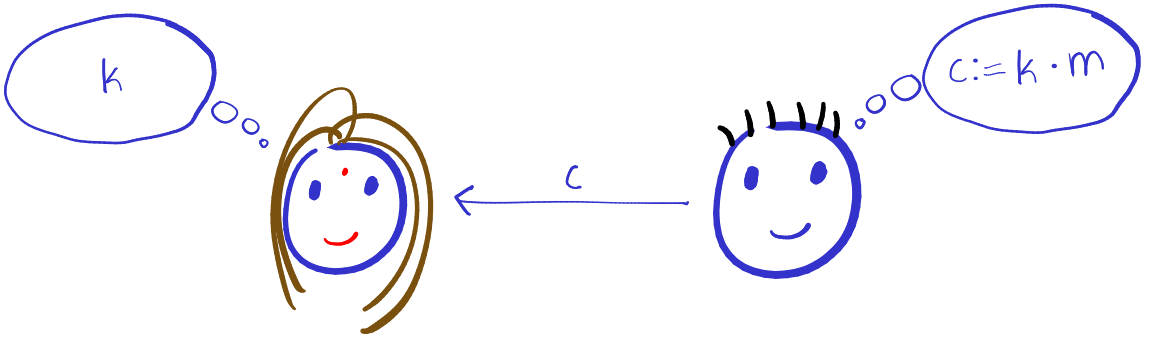

## Pseudocode 1 (OTP over $(\{0,1\}^n, \oplus)$ with message space $\{0,1\}^n$)

- *Key generation* $\mathsf{Gen}$*: output* $k \leftarrow \{0,1\}^n$
- *Encryption* $\mathsf{Enc}(k, m)$*: output* $c := k \oplus m$
- *Decryption* $\mathsf{Dec}(k, c)$*: output* $m := k \oplus c$

## Pseudocode 2 (OTP over group $\mathbb{G} := (\mathcal{G}, \cdot)$ with message space $\mathcal{G}$)

- *Key generation* $\mathsf{Gen}$*: output* $k \leftarrow \mathcal{G}$

# One-Time Pad Can be Generalised Over Groups



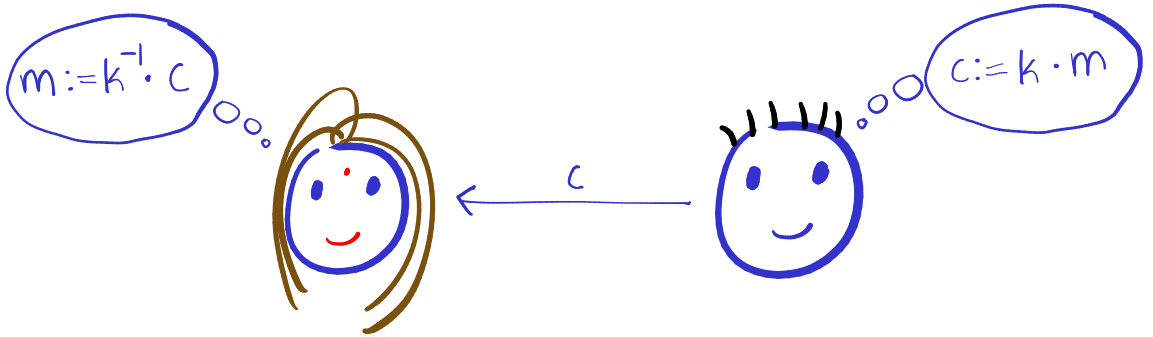

**Pseudocode 1 (OTP over $(\{0,1\}^n, \oplus)$ with message space $\{0,1\}^n$)**

- *Key generation* $\mathsf{Gen}$*: output* $k \leftarrow \{0,1\}^n$
- *Encryption* $\mathsf{Enc}(k, m)$*: output* $c := k \oplus m$
- *Decryption* $\mathsf{Dec}(k, c)$*: output* $m := k \oplus c$

**Pseudocode 2 (OTP over group $\mathbb{G} := (\mathcal{G}, \cdot)$ with message space $\mathcal{G}$)**

- *Key generation* $\mathsf{Gen}$*: output* $k \leftarrow \mathcal{G}$
- *Encryption* $\mathsf{Enc}(k, m)$*: output* $c := k \cdot m$

# One-Time Pad Can be Generalised Over Groups



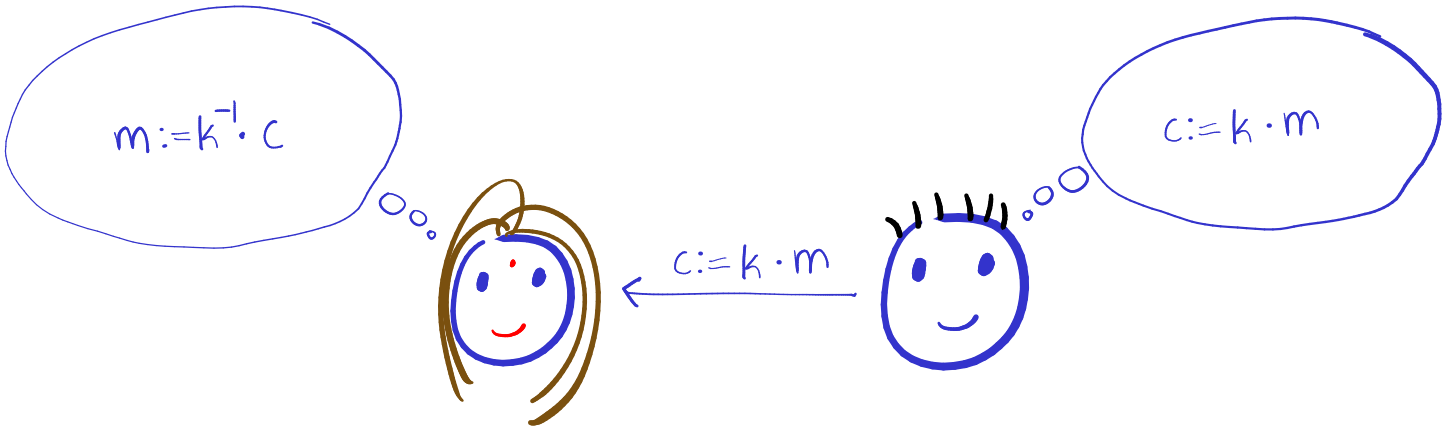

## Pseudocode 1 (OTP over $(\{0,1\}^n, \oplus)$ with message space $\{0,1\}^n$)

- *Key generation* $\mathsf{Gen}$*: output* $k \leftarrow \{0,1\}^n$
- *Encryption* $\mathsf{Enc}(k, m)$*: output* $c := k \oplus m$
- *Decryption* $\mathsf{Dec}(k, c)$*: output* $m := k \oplus c$

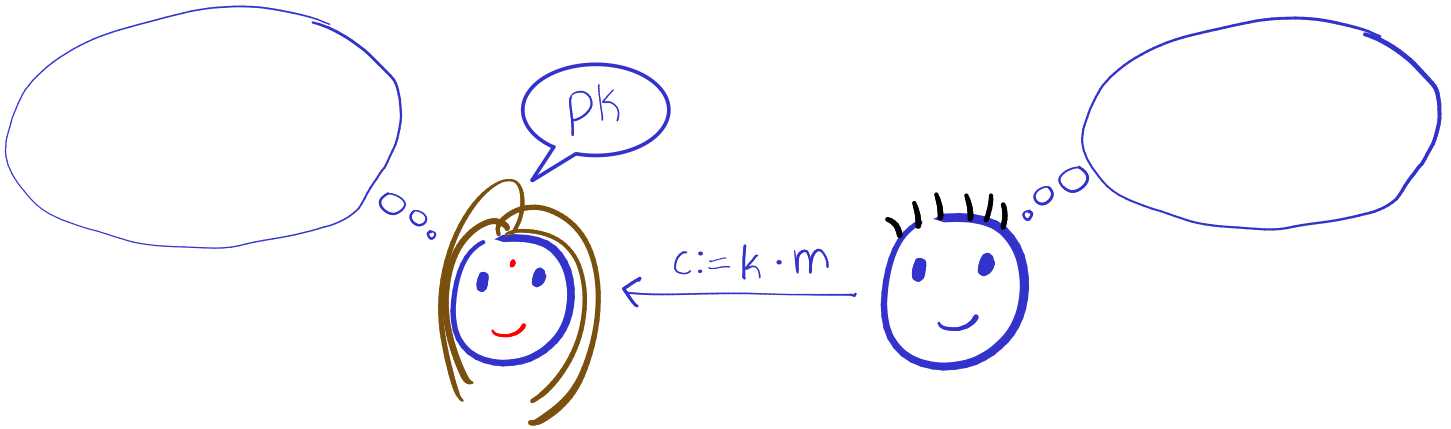
## Pseudocode 2 (OTP over group $\mathbb{G} := (\mathcal{G}, \cdot)$ with message space $\mathcal{G}$)

- *Key generation* $\mathsf{Gen}$*: output* $k \leftarrow \mathcal{G}$
- *Encryption* $\mathsf{Enc}(k, m)$*: output* $c := k \cdot m$
- *Decryption* $\mathsf{Dec}(k, c)$*: output* $m := k^{-1} \cdot c$

# Let's Build on Group-Based OTP to Construct a PKE

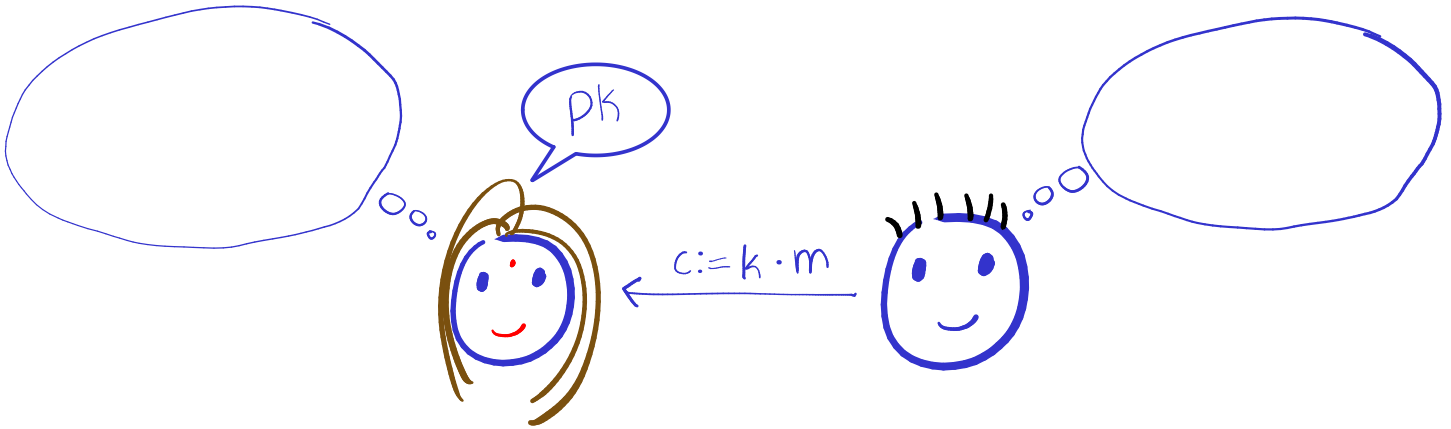


# Let's Build on Group-Based OTP to Construct a PKE



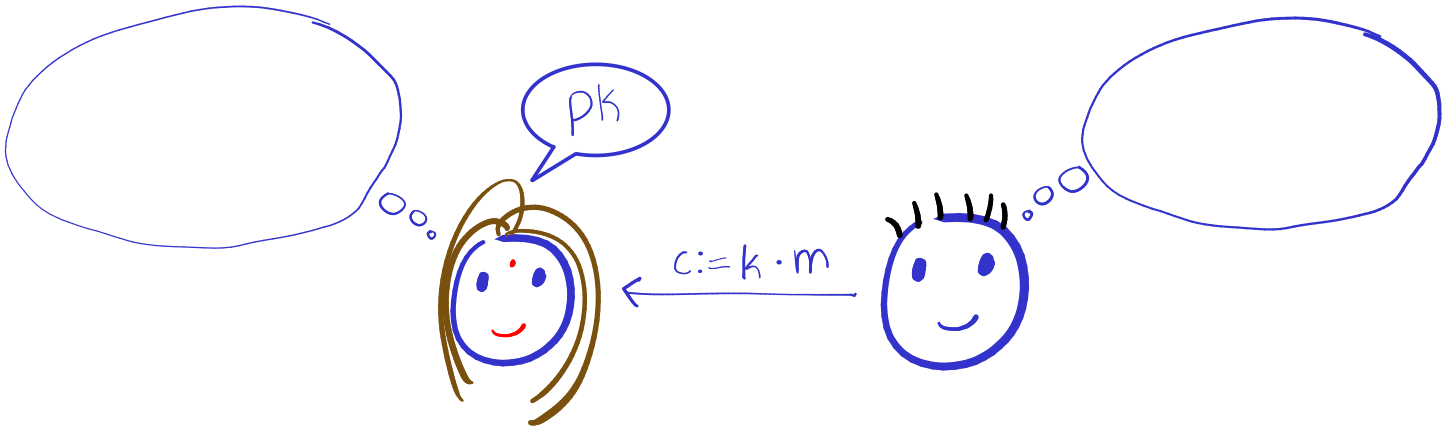

- Our ciphertexts will be of form $c := k \cdot m$

# Let's Build on Group-Based OTP to Construct a PKE




- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP $k$
  2. Eve mustn't be able to generate this $k$ from $pk$ and ciphertext $c$

# Let's Build on Group-Based OTP to Construct a PKE

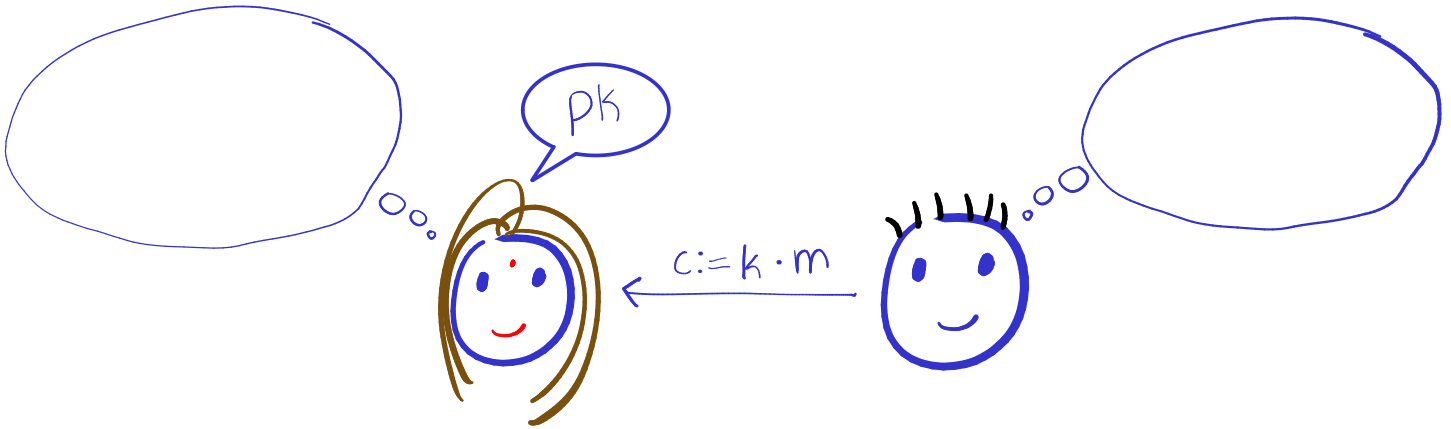



- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP $k$
  2. Eve mustn't be able to generate this $k$ from $pk$ and ciphertext $c$

Any ideas on

1. What can the public key $pk$ be?
2. How to generate $k$?

# Let's Build on Group-Based OTP to Construct a PKE



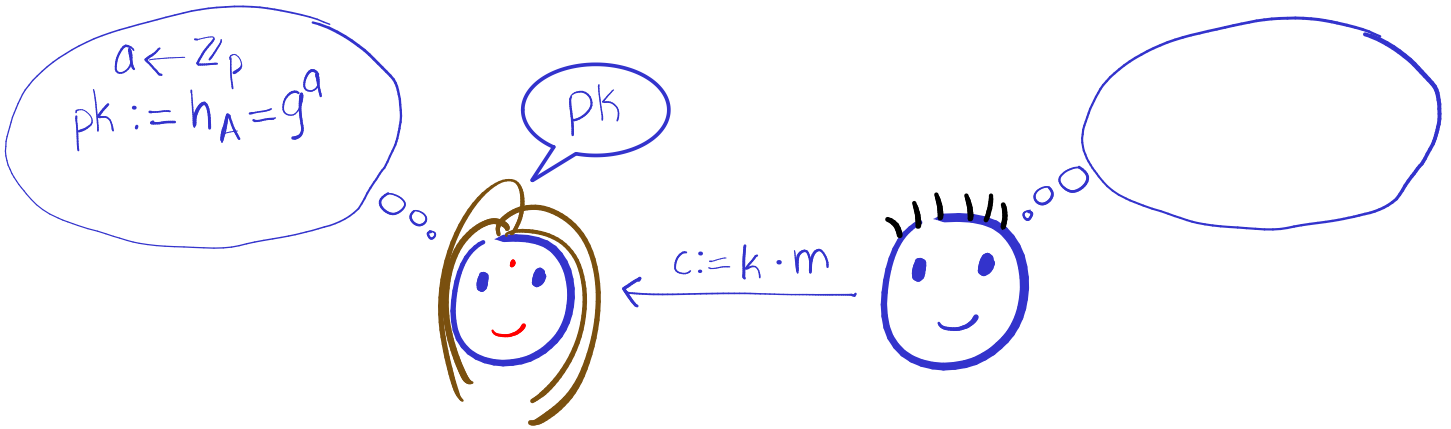

- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP $k$
  2. Eve mustn't be able to generate this $k$ from $pk$ and ciphertext $c$

Any ideas on

1. What can the public key $pk$ be?
2. How to generate $k$?

Hint: we have already exploited this "structure" in DHKE

# Let's Build on Group-Based OTP to Construct a PKE



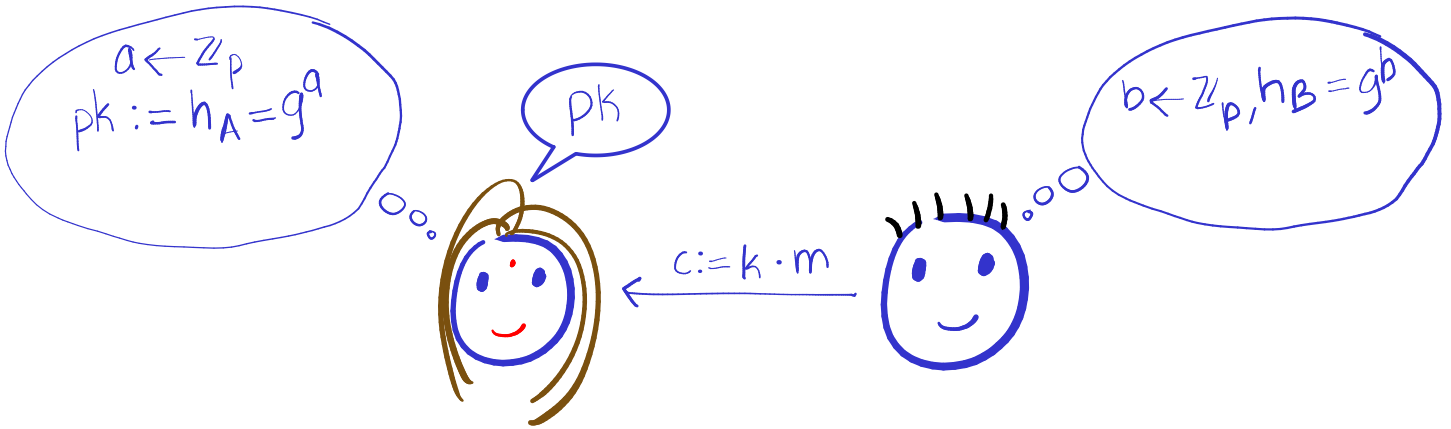

- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP $k$
  2. Eve mustn't be able to generate this $k$ from $pk$ and ciphertext $c$

Any ideas on

1. What can the public key $pk$ be?
2. How to generate $k$?

Hint: we have already exploited this "structure" in DHKE

# Let's Build on Group-Based OTP to Construct a PKE



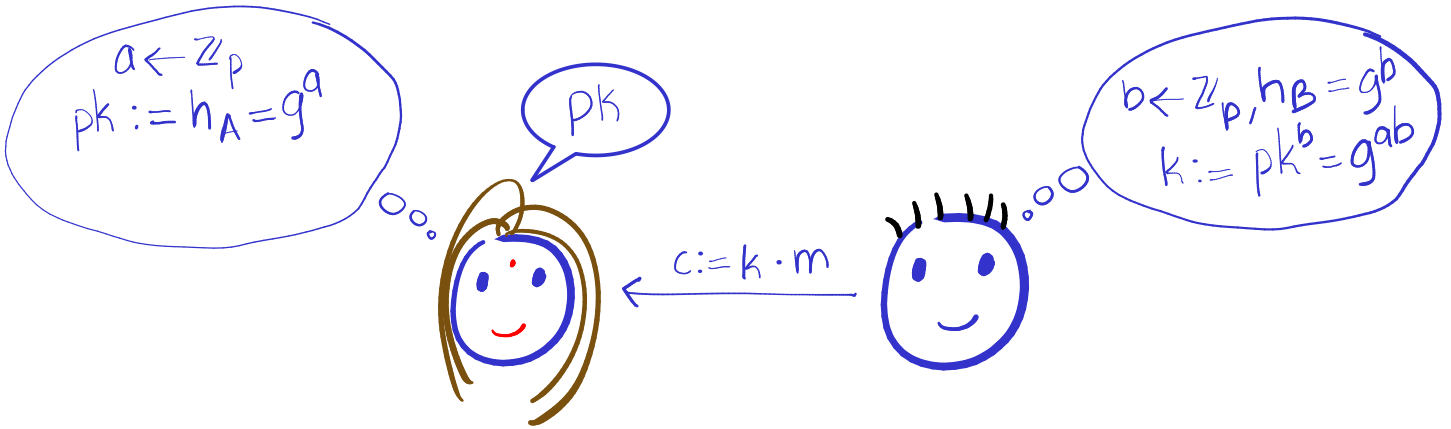

- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP $k$
  2. Eve mustn't be able to generate this $k$ from $pk$ and ciphertext $c$

Any ideas on

1. What can the public key $pk$ be?
2. How to generate $k$?

Hint: we have already exploited this "structure" in DHKE

# Let's Build on Group-Based OTP to Construct a PKE



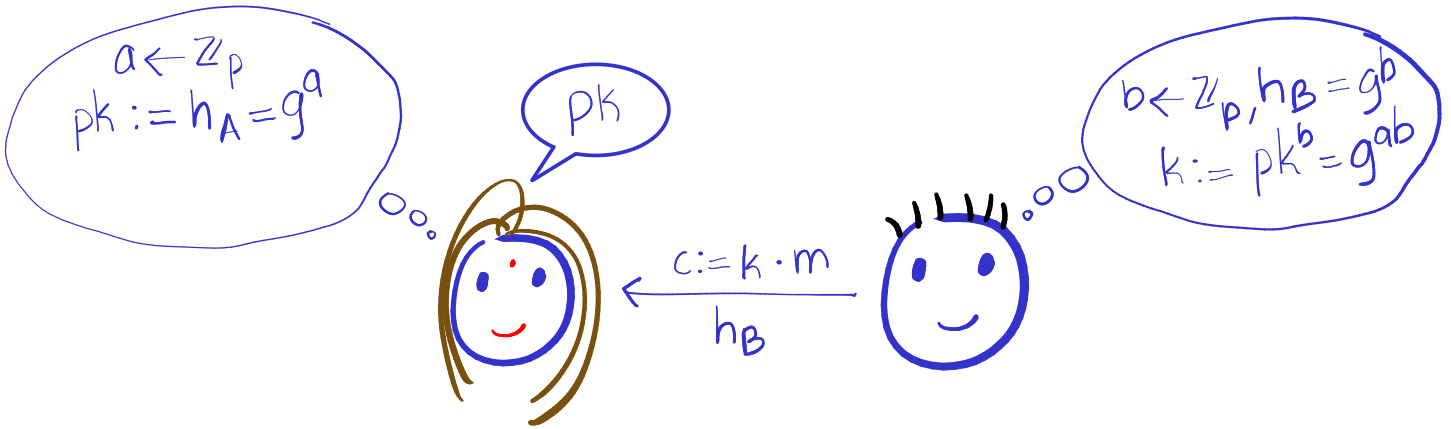

- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP $k$
  2. Eve mustn't be able to generate this $k$ from $pk$ and ciphertext $c$

Any ideas on

1. What can the public key $pk$ be?
2. How to generate $k$?

Hint: we have already exploited this "structure" in DHKE

# Let's Build on Group-Based OTP to Construct a PKE



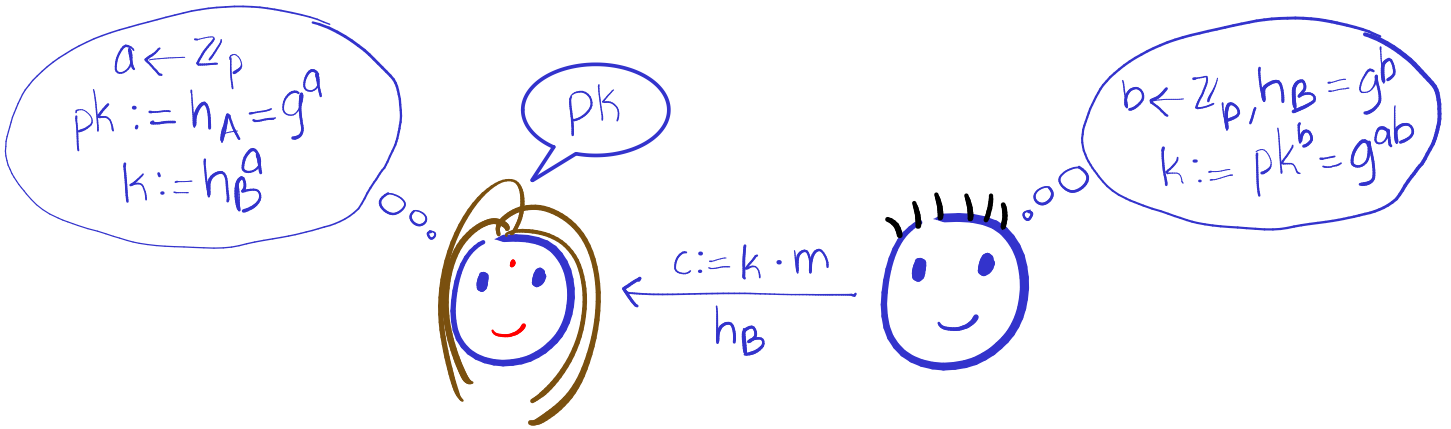

- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP $k$
  2. Eve mustn't be able to generate this $k$ from $pk$ and ciphertext $c$

Any ideas on

1. What can the public key $pk$ be?
2. How to generate $k$?

Hint: we have already exploited this "structure" in DHKE

# Let's Build on Group-Based OTP to Construct a PKE




- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP $k$
  2. Eve mustn't be able to generate this $k$ from $pk$ and ciphertext $c$

Any ideas on

1. What can the public key $pk$ be?
2. How to generate $k$?

Hint: we have already exploited this "structure" in DHKE

# ElGamal PKE over Group $\mathbb{G}$...

## Pseudocode 3 (ElGamal PKE over group $\mathbb{G} = (\mathcal{G}, \cdot)$)

- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample group* $(\mathbb{G}, p, g) \leftarrow \mathsf{S}(1^n)$
  2. *Sample random index* $a \leftarrow \mathbb{Z}_p$
  3. *Output* $(pk := g^a, sk := a)$

# ElGamal PKE over Group $\mathbb{G}$...

## Pseudocode 3 (ElGamal PKE over group $\mathbb{G} = (\mathcal{G}, \cdot)$)

- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample group* $(\mathbb{G}, p, g) \leftarrow \mathsf{S}(1^n)$
  2. *Sample random index* $a \leftarrow \mathbb{Z}_p$
  3. *Output* $(pk := g^a, sk := a)$
- *Encryption* $\mathsf{Enc}(pk, m)$*:*
  1. *Sample random index* $b \leftarrow \mathbb{Z}_p$*, and set* $k := pk^b$
  2. *Output* $c := (c_1, c_2) := (k \cdot m, g^b)$

# ElGamal PKE over Group $\mathbb{G}$...

## Pseudocode 3 (ElGamal PKE over group $\mathbb{G} = (\mathcal{G}, \cdot)$)

- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample group* $(\mathbb{G}, p, g) \leftarrow \mathsf{S}(1^n)$
  2. *Sample random index* $a \leftarrow \mathbb{Z}_p$
  3. *Output* $(pk := g^a, sk := a)$
- *Encryption* $\mathsf{Enc}(pk, m)$*:*
  1. *Sample random index* $b \leftarrow \mathbb{Z}_p$*, and set* $k := pk^b$
  2. *Output* $c := (c_1, c_2) := (k \cdot m, g^b)$
- *Decryption* $\mathsf{Dec}(sk, c =: (c_1, c_2))$*: output* $m := (c_2^{sk})^{-1} \cdot c_1$

# ElGamal PKE over Group $\mathbb{G}$...

## Pseudocode 3 (ElGamal PKE over group $\mathbb{G} = (\mathcal{G}, \cdot)$)

- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample group* $(\mathbb{G}, p, g) \leftarrow \mathsf{S}(1^n)$
  2. *Sample random index* $a \leftarrow \mathbb{Z}_p$
  3. *Output* $(pk := g^a, sk := a)$
- *Encryption* $\mathsf{Enc}(pk, m)$*:*
  1. *Sample random index* $b \leftarrow \mathbb{Z}_p$*, and set* $k := pk^b$
  2. *Output* $c := (c_1, c_2) := (k \cdot m, g^b)$
- *Decryption* $\mathsf{Dec}(sk, c =: (c_1, c_2))$*: output* $m := (c_2^{sk})^{-1} \cdot c_1$
- *Correctness of decryption:*

# ElGamal PKE over Group $\mathbb{G}$...

## Pseudocode 3 (ElGamal PKE over group $\mathbb{G} = (\mathcal{G}, \cdot)$)

- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample group* $(\mathbb{G}, p, g) \leftarrow \mathsf{S}(1^n)$
  2. *Sample random index* $a \leftarrow \mathbb{Z}_p$
  3. *Output* $(pk := g^a, sk := a)$
- *Encryption* $\mathsf{Enc}(pk, m)$*:*
  1. *Sample random index* $b \leftarrow \mathbb{Z}_p$*, and set* $k := pk^b = (g^a)^b = g^{ab}$
  2. *Output* $c := (c_1, c_2) := (k \cdot m, g^b)$
- *Decryption* $\mathsf{Dec}(sk, c =: (c_1, c_2))$*: output* $m := (c_2^{sk})^{-1} \cdot c_1$
- *Correctness of decryption:*

# ElGamal PKE over Group $\mathbb{G}$...

## Pseudocode 3 (ElGamal PKE over group $\mathbb{G} = (\mathcal{G}, \cdot)$)

- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample group* $(\mathbb{G}, p, g) \leftarrow \mathsf{S}(1^n)$
  2. *Sample random index* $a \leftarrow \mathbb{Z}_p$
  3. *Output* $(pk := g^a, sk := a)$
- *Encryption* $\mathsf{Enc}(pk, m)$*:*
  1. *Sample random index* $b \leftarrow \mathbb{Z}_p$*, and set* $k := pk^b = (g^a)^b = g^{ab}$
  2. *Output* $c := (c_1, c_2) := (k \cdot m, g^b)$
- *Decryption* $\mathsf{Dec}(sk, c =: (c_1, c_2))$*: output* $m := (c_2^{sk})^{-1} \cdot c_1$
- *Correctness of decryption:*

$(g^b)^a = g^{ab}$ $\qquad$ $g^{ab} \cdot m$

# ElGamal PKE over Group $\mathbb{G}$...

## Pseudocode 3 (ElGamal PKE over group $\mathbb{G} = (\mathcal{G}, \cdot)$)

- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample group* $(\mathbb{G}, p, g) \leftarrow \mathsf{S}(1^n)$
  2. *Sample random index* $a \leftarrow \mathbb{Z}_p$
  3. *Output* $(pk := g^a, sk := a)$
- *Encryption* $\mathsf{Enc}(pk, m)$*:*
  1. *Sample random index* $b \leftarrow \mathbb{Z}_p$*, and set* $k := pk^b = (g^a)^b = g^{ab}$
  2. *Output* $c := (c_1, c_2) := (k \cdot m, g^b)$
- *Decryption* $\mathsf{Dec}(sk, c =: (c_1, c_2))$*: output* $m := (c_2^{sk})^{-1} \cdot c_1$
- *Correctness of decryption:*

$(g^b)^a = g^{ab}$ $\qquad$ $g^{ab} \cdot m$

# ElGamal PKE is CPA-secret

**Theorem 1 (DDH → CPA-PKE)**

*ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to $\mathsf{S}$.*

**Proof sketch. Hybrid argument.**

Four hybrids $H_0, H_0', H_1', H_1$

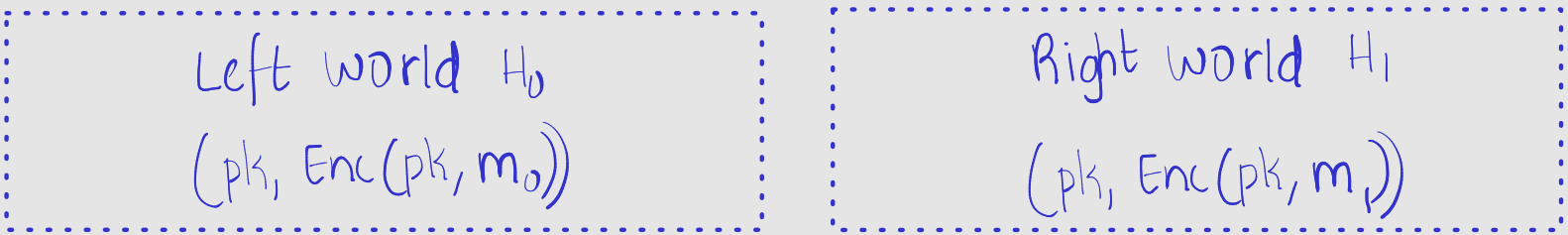
# ElGamal PKE is CPA-secret

**Theorem 1 (DDH → CPA-PKE)**

*ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to $\mathsf{S}$.*

**Proof sketch. Hybrid argument.**

Four hybrids $H_0, H_0', H_1', H_1$



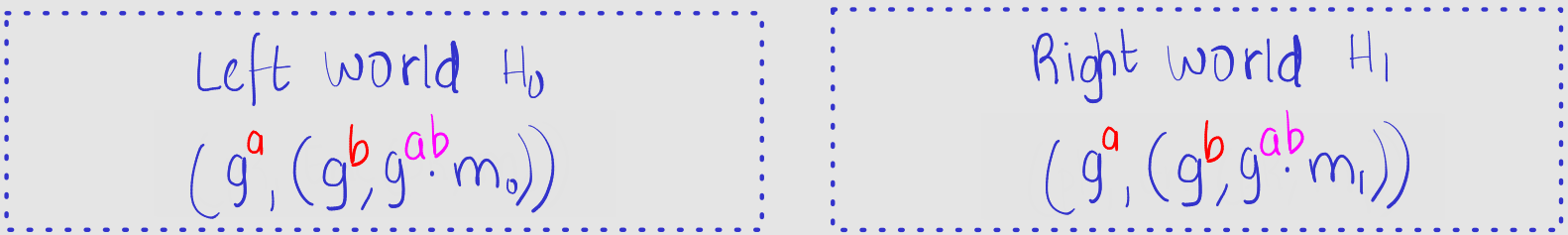
# ElGamal PKE is CPA-secret

**Theorem 1 (DDH → CPA-PKE)**

*ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to $\mathsf{S}$.*

**Proof sketch. Hybrid argument.**

Four hybrids $H_0, H_0', H_1', H_1$



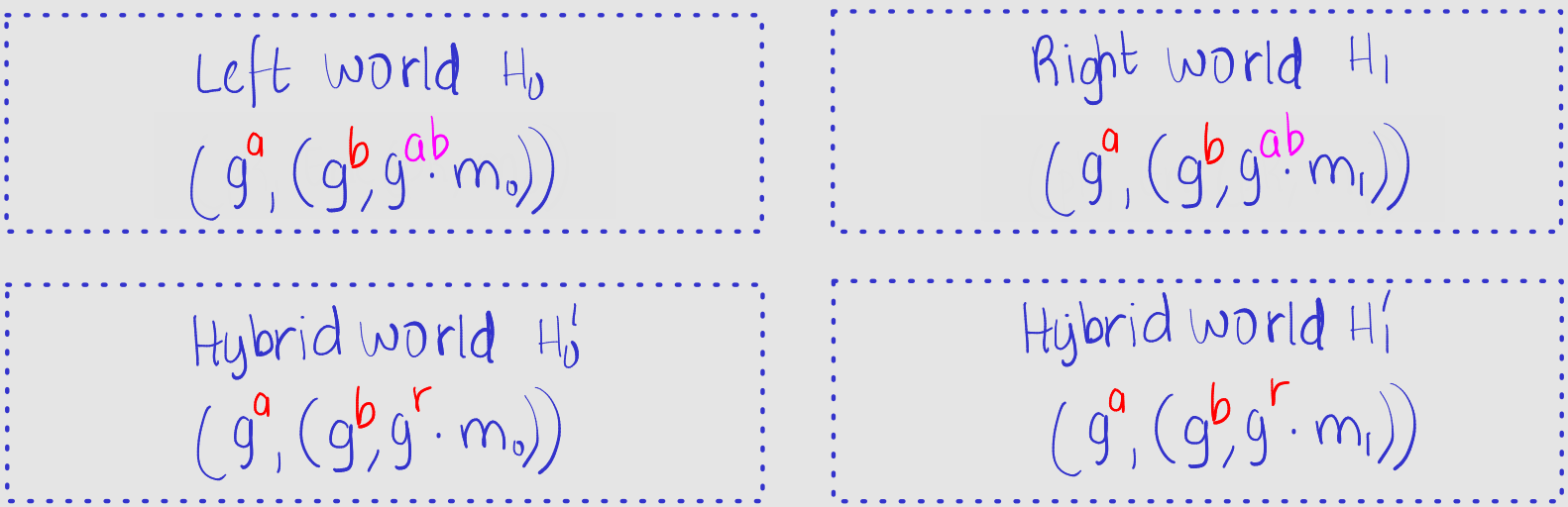
# ElGamal PKE is CPA-secret

**Theorem 1 (DDH → CPA-PKE)**

*ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to $\mathsf{S}$.*

**Proof sketch. Hybrid argument.**

Four hybrids $H_0, H_0', H_1', H_1$



# ElGamal PKE is CPA-secret

**Theorem 1 (DDH → CPA-PKE)**

*ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to $\mathsf{S}$.*

**Proof sketch. Hybrid argument.**

Four hybrids $H_0, H_0', H_1', H_1$

Left world $H_0$

$(g^a, (g^b, g^{ab} \cdot m_0))$

Right world $H_1$

$(g^a, (g^b, g^{ab} \cdot m_1))$

Hybrid world $H_0'$

$(g^a, (g^b, g^r \cdot m_0))$

Hybrid world $H_1'$

$(g^a, (g^b, g^r \cdot m_1))$

? Why is $H_0$ indistinguishable from $H_0'$ ?

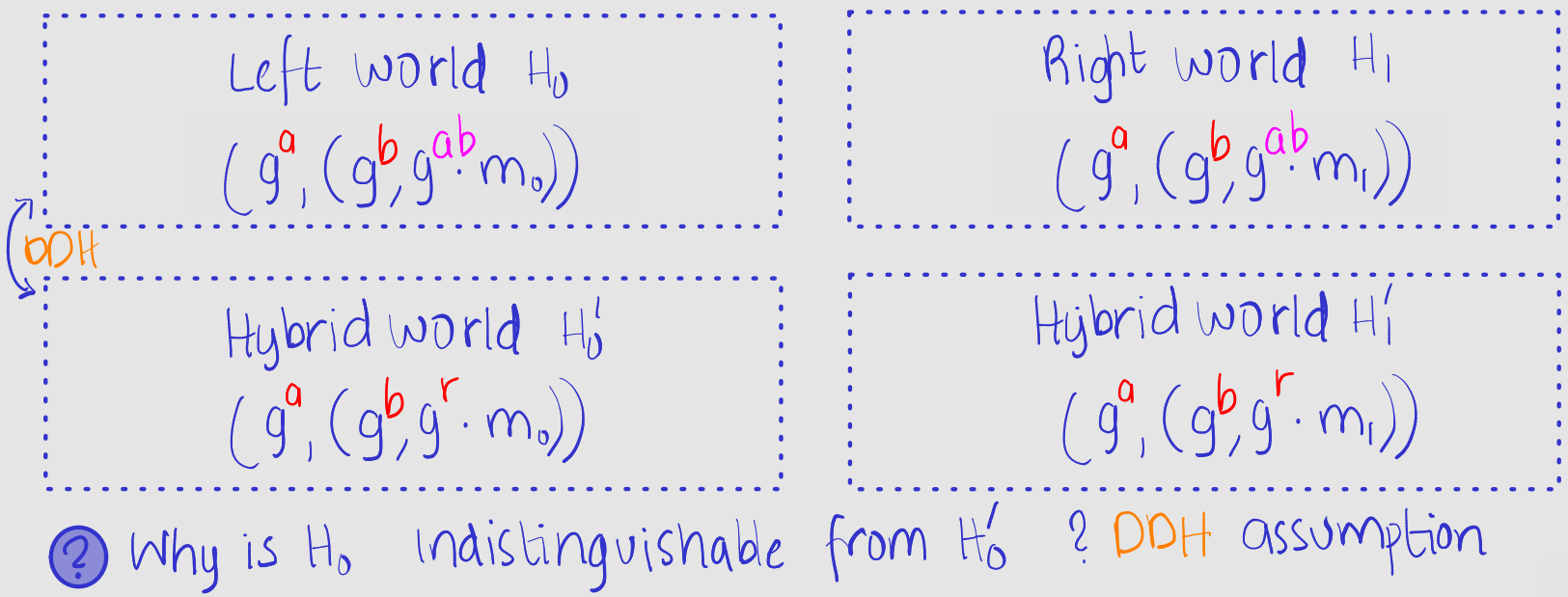
# ElGamal PKE is CPA-secret

**Theorem 1 (DDH → CPA-PKE)**

*ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to $\mathsf{S}$.*

**Proof sketch. Hybrid argument.**

Four hybrids $H_0, H_0', H_1', H_1$




? Why is $H_0$ indistinguishable from $H_0'$ ? DDH assumption

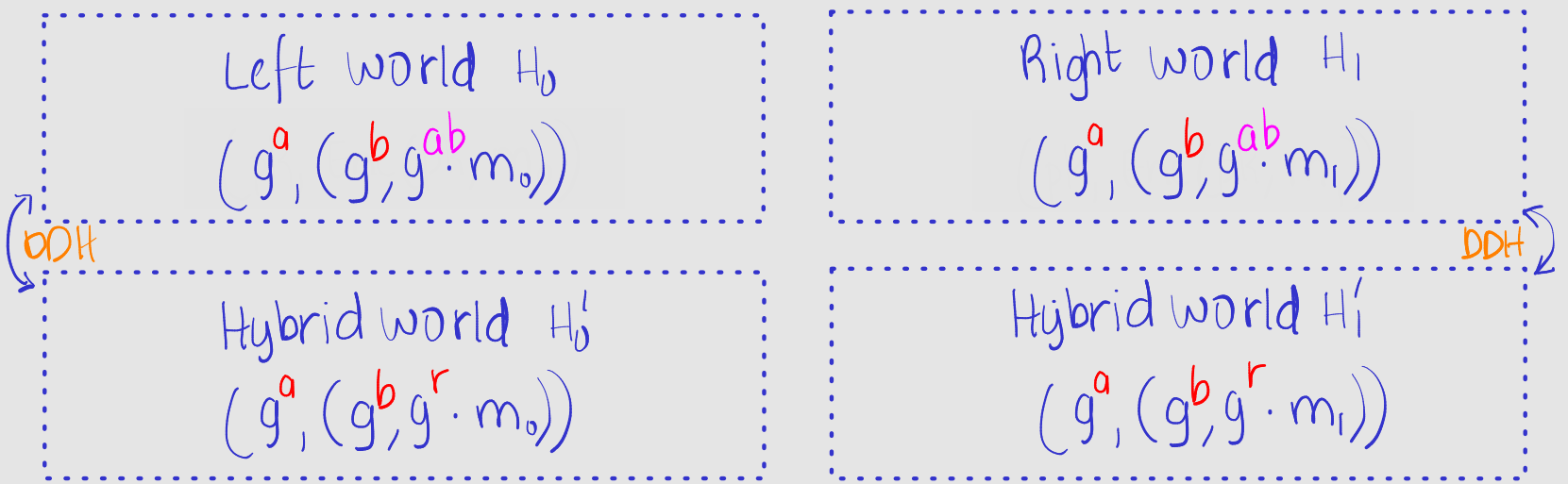
# ElGamal PKE is CPA-secret

**Theorem 1 (DDH → CPA-PKE)**

*ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to $\mathsf{S}$.*

**Proof sketch. Hybrid argument.**

Four hybrids $H_0, H_0', H_1', H_1$




? Why is $H_0/H_1$ indistinguishable from $H_0'/H_1'$? DDH assumption

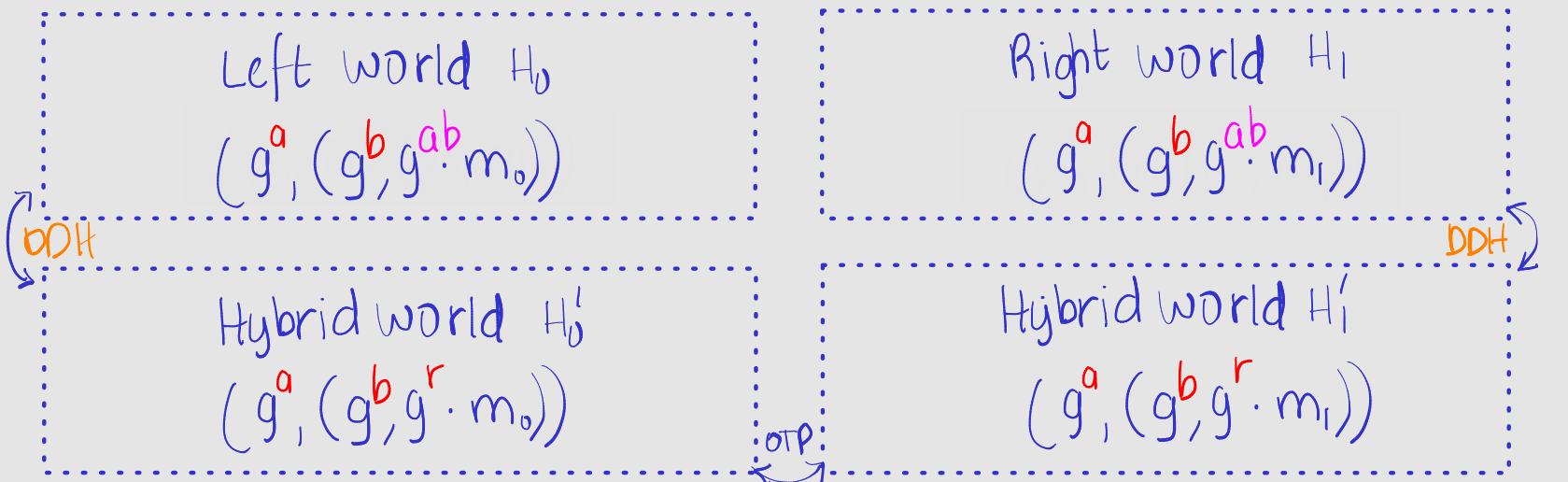
# ElGamal PKE is CPA-secret

**Theorem 1 (DDH → CPA-PKE)**

*ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to $\mathsf{S}$.*

**Proof sketch. Hybrid argument.**

Four hybrids $H_0, H_0', H_1', H_1$




? Why is $H_0/H_1$ indistinguishable from $H_0'/H_1'$? DDH assumption

? Why is $H_0'$ indistinguishable from $H_1'$? OTP over group

# Is it a Coincidence that ElGamal is Similar to DHKE? No!

**Claim 1 (Two-message KE $\rightarrow$ CPA-PKE)**

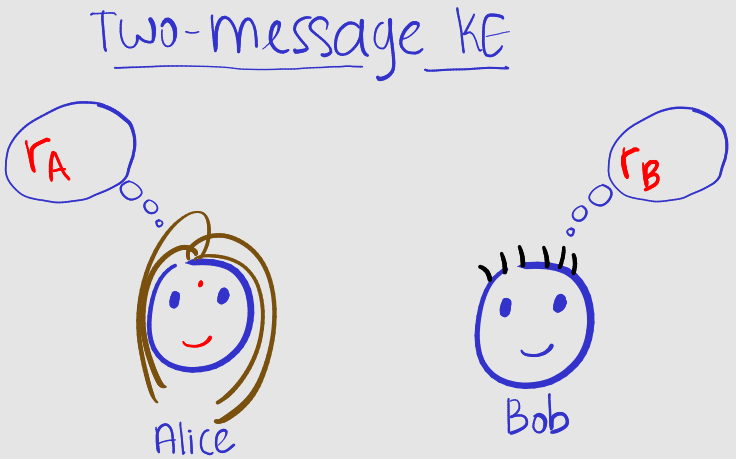
*If two-message key exchange protocol $\Pi$ exists then so does PKE.*

# Is it a Coincidence that ElGamal is Similar to DHKE? No!

## Claim 1 (Two-message KE → CPA-PKE)

*If two-message key exchange protocol* $\Pi$ *exists then so does PKE.*

## Construction 1

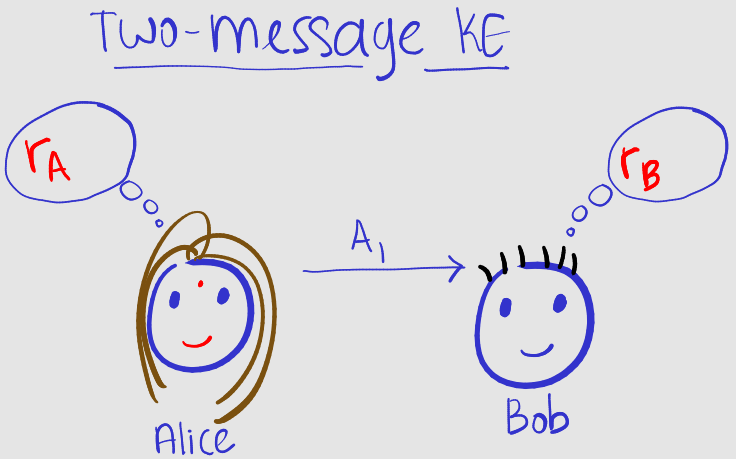


# Is it a Coincidence that ElGamal is Similar to DHKE? No!

## Claim 1 (Two-message KE → CPA-PKE)

*If two-message key exchange protocol* $\Pi$ *exists then so does PKE.*

## Construction 1

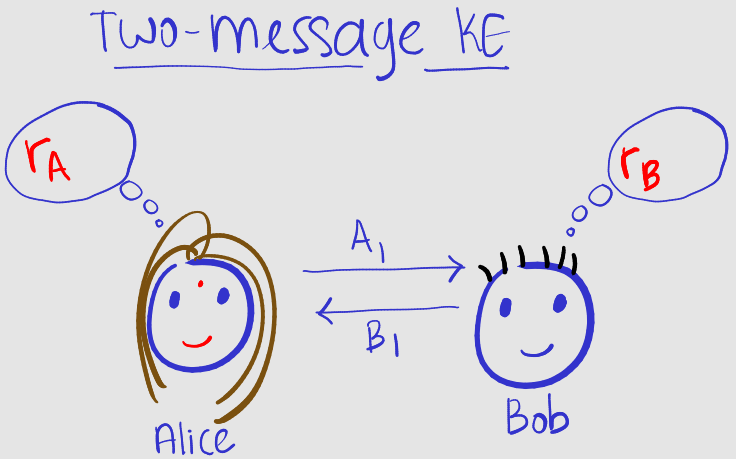


# Is it a Coincidence that ElGamal is Similar to DHKE? No!

## Claim 1 (Two-message KE → CPA-PKE)

*If two-message key exchange protocol* $\Pi$ *exists then so does PKE.*

## Construction 1

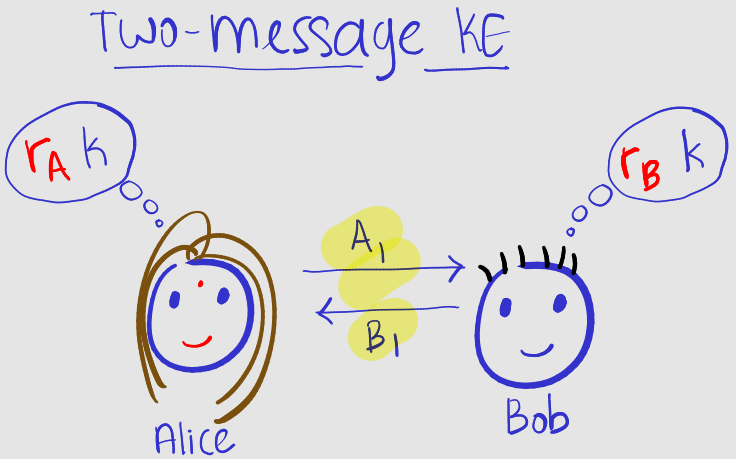


# Is it a Coincidence that ElGamal is Similar to DHKE? No!

## Claim 1 (Two-message KE → CPA-PKE)

*If two-message key exchange protocol* $\Pi$ *exists then so does PKE.*

## Construction 1

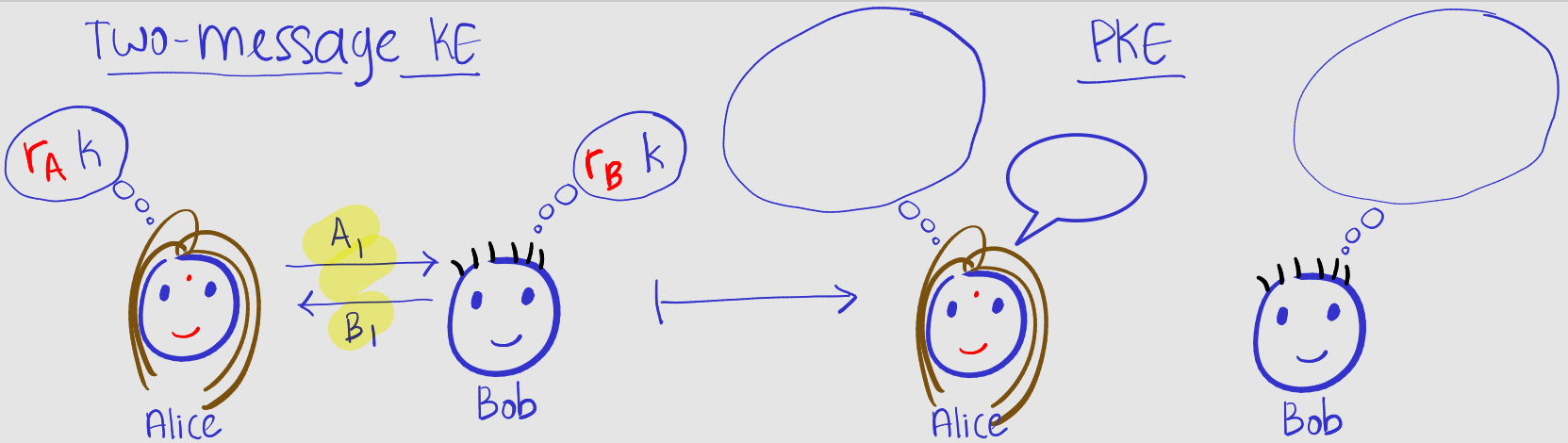


# Is it a Coincidence that ElGamal is Similar to DHKE? No!

## Claim 1 (Two-message KE → CPA-PKE)

*If two-message key exchange protocol* $\Pi$ *exists then so does PKE.*

## Construction 1

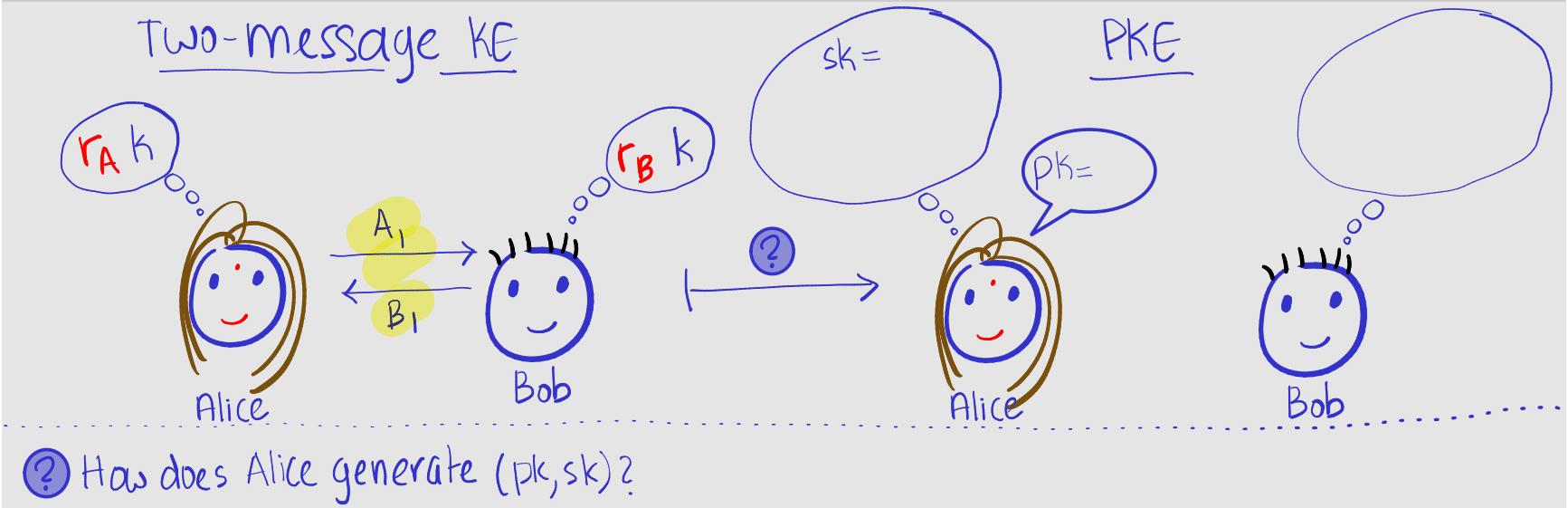


# Is it a Coincidence that ElGamal is Similar to DHKE? No!

## Claim 1 (Two-message KE → CPA-PKE)

*If two-message key exchange protocol* $\Pi$ *exists then so does PKE.*

## Construction 1

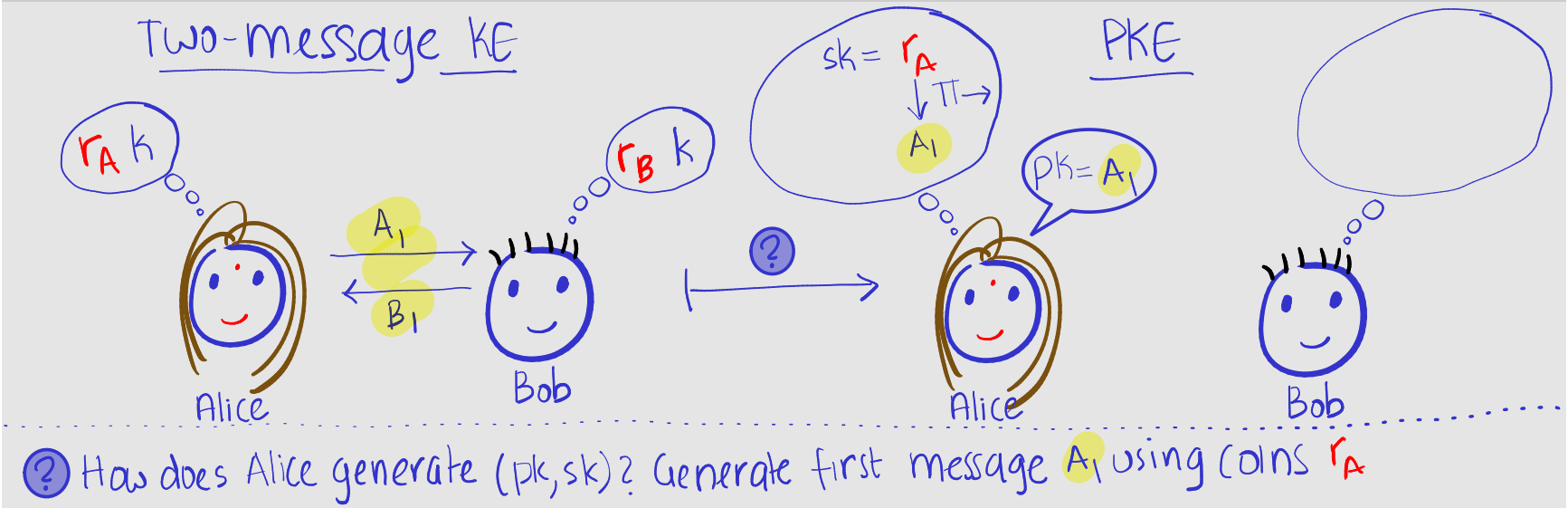


# Is it a Coincidence that ElGamal is Similar to DHKE? No!

## Claim 1 (Two-message KE → CPA-PKE)

*If two-message key exchange protocol* $\Pi$ *exists then so does PKE.*

## Construction 1

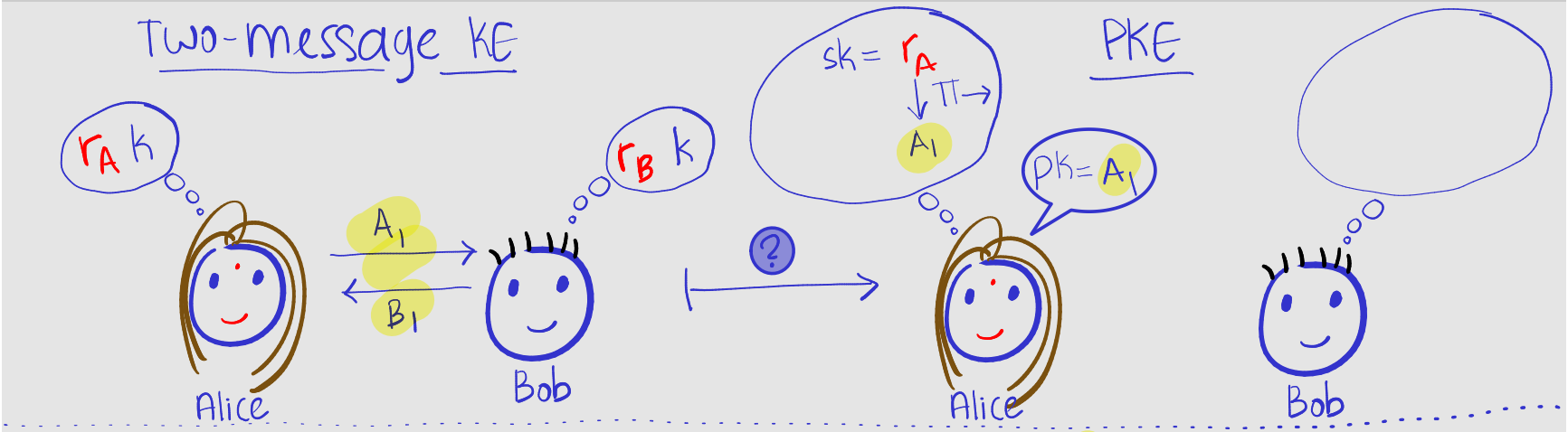


# Is it a Coincidence that ElGamal is Similar to DHKE? No!

## Claim 1 (Two-message KE → CPA-PKE)

*If two-message key exchange protocol* $\Pi$ *exists then so does PKE.*

## Construction 1




? How does Alice generate (pk, sk)? Generate first message $A_1$ using coins $r_A$

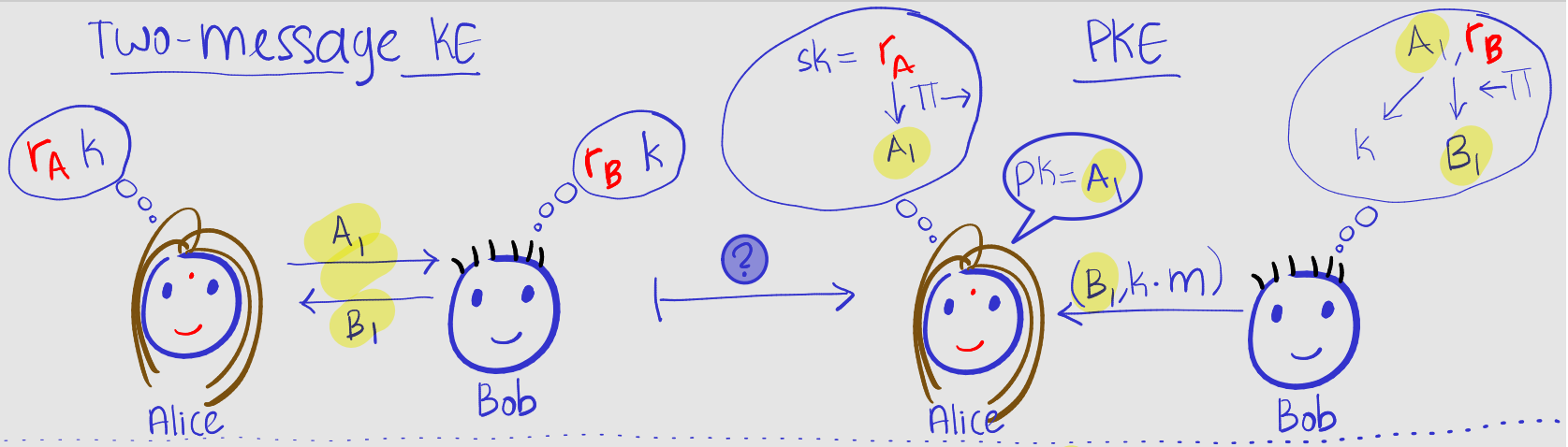
? How does Bob encrypt?

# Is it a Coincidence that ElGamal is Similar to DHKE? No!

## Claim 1 (Two-message KE → CPA-PKE)

*If two-message key exchange protocol* $\Pi$ *exists then so does PKE.*

## Construction 1




? How does Alice generate (pk, sk)? Generate first message $A_1$ using coins $r_A$

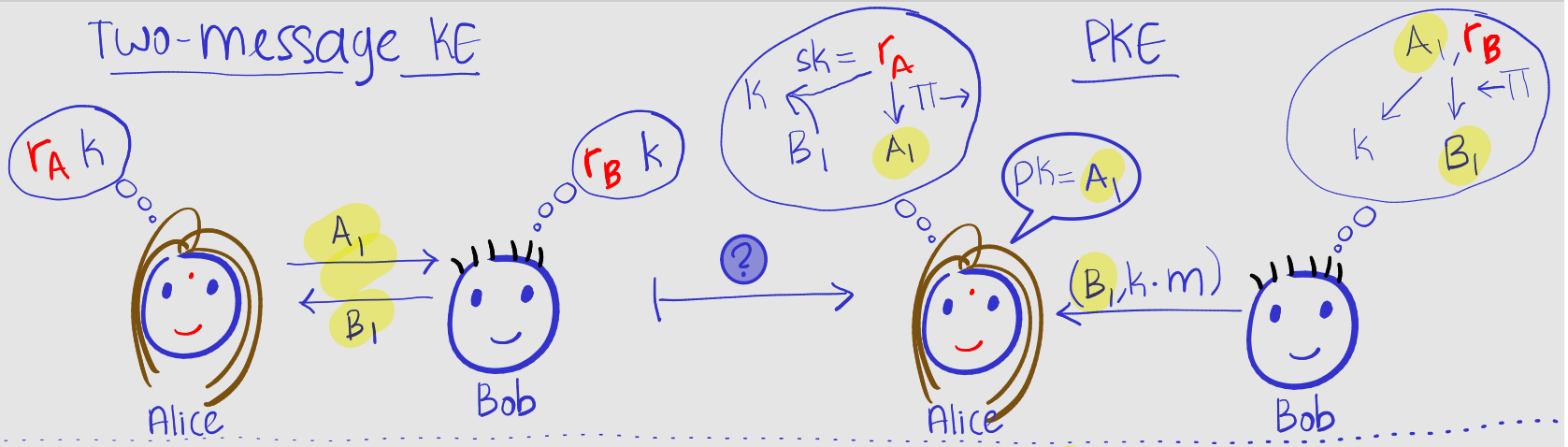
? How does Bob encrypt? Generate 2nd message $B_1$ and shared key k using coins $r_B$ and $A_1$. Use k as OTP; send $B_1$ to help decryption.

# Is it a Coincidence that ElGamal is Similar to DHKE? No!

## Claim 1 (Two-message KE → CPA-PKE)

*If two-message key exchange protocol* $\Pi$ *exists then so does PKE.*

## Construction 1




? How does Alice generate (pk, sk)? Generate first message $A_1$ using coins $r_A$

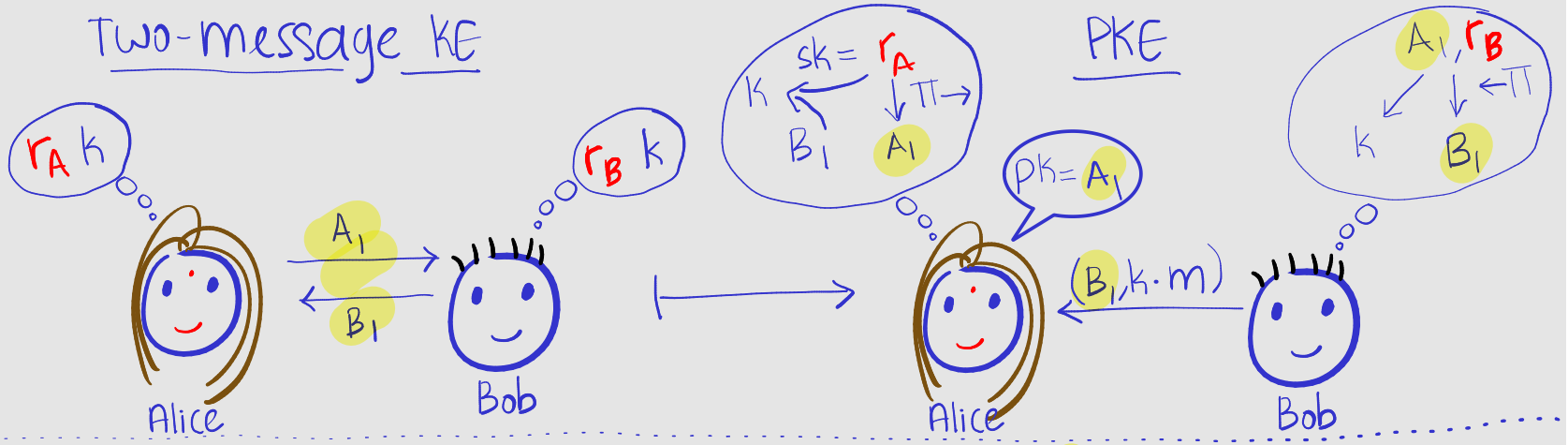
? How does Bob encrypt? Generate 2nd message $B_1$ and shared key $k$ using coins $r_B$ and $A_1$. Use $k$ as OTP; send $B_1$ to help decryption.

# Is it a Coincidence that ElGamal is Similar to DHKE? No!

## Claim 1 (Two-message KE → CPA-PKE)

*If two-message key exchange protocol Π exists then so does PKE.*

## Construction 1




? How does Alice generate (pk, sk)? Generate first message $A_1$ using coins $r_A$

? How does Bob encrypt? Generate 2nd message $B_1$ and shared key $k$ using coins $r_B$ and $A_1$. Use $k$ as OTP; send $B_1$ to help decryption.

## Exercise 1 (Converse to Claim 1: two-message KE ← CPA-PKE)

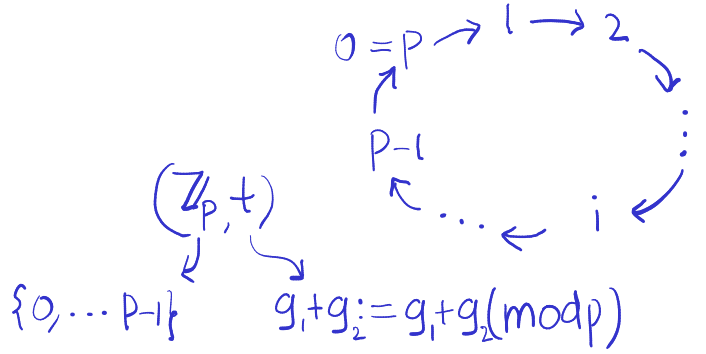
*If PKE exists then so does two-message key exchange.*

# Plan for this Lecture

1. Public-Key Encryption (PKE)
2. ElGamal PKE ⟵ DDH
3. Goldwasser-Micali PKE ⟵ QR

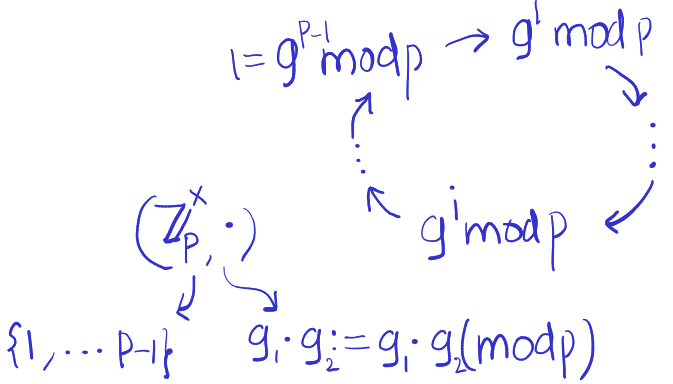
# But First, Some Number Theory

## Addition modulo prime $p$




- order $p$
- cyclic

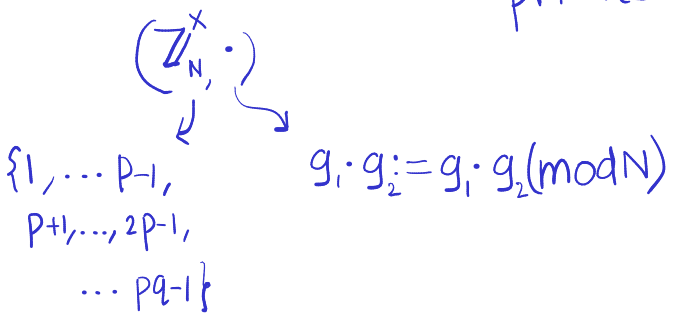
## Multiplication modulo prime $p$




- order $p-1$
- cyclic

## Multiplication modulo $N = pq$

primes



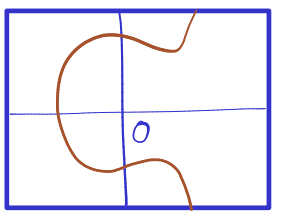

- order $(p-1)(q-1)$
- not cyclic

## Elliptic curves modulo prime $p$

$(E, +)$ "curve points addition"

solutions to

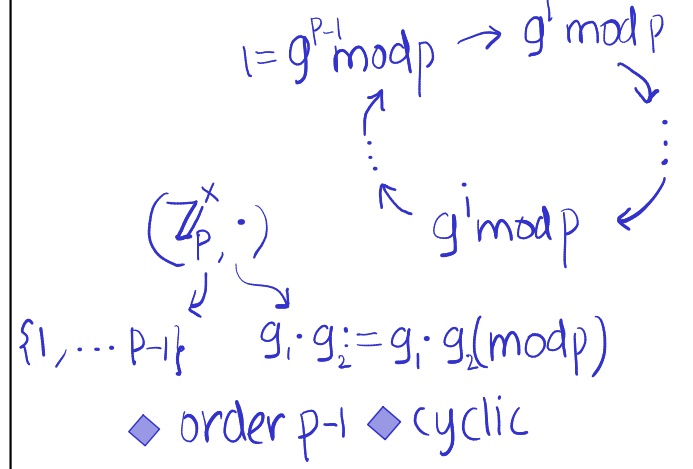
$y^2 = x^3 + Ax + B \pmod p$




- $|p+1 - \text{order}| \leq 2\sqrt{p}$
- cyclic

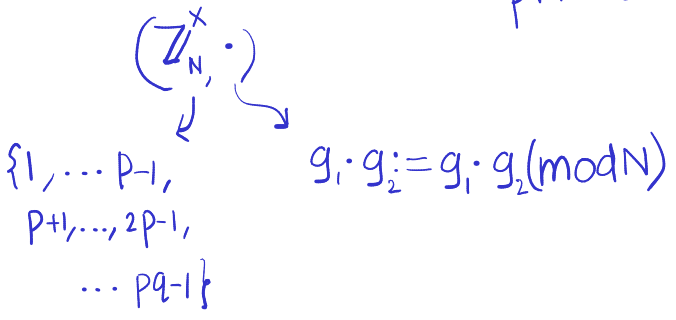
# But First, Some Number Theory

**Multiplication modulo prime $p$** $= 2p'+1$ ($p'$ prime)




- order $p-1$
- cyclic

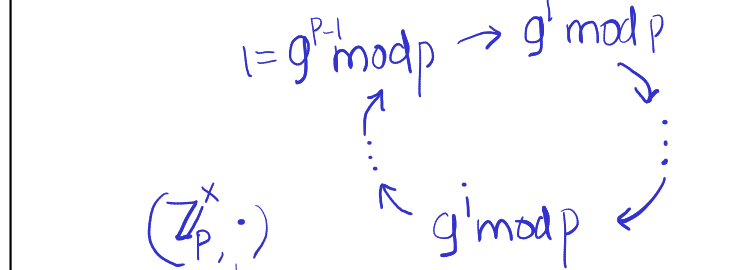
**Multiplication modulo $N = pq$** $= (2p'+1)\cdot(2q'+1)$ ($p', q'$ primes)




- order $(p-1)(q-1)$
- not cyclic

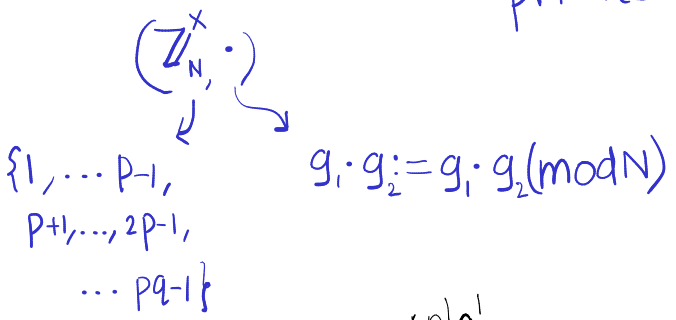
# But First, Some Number Theory

**Multiplication modulo prime $p$** $= 2p' + 1$ ($p'$ prime)



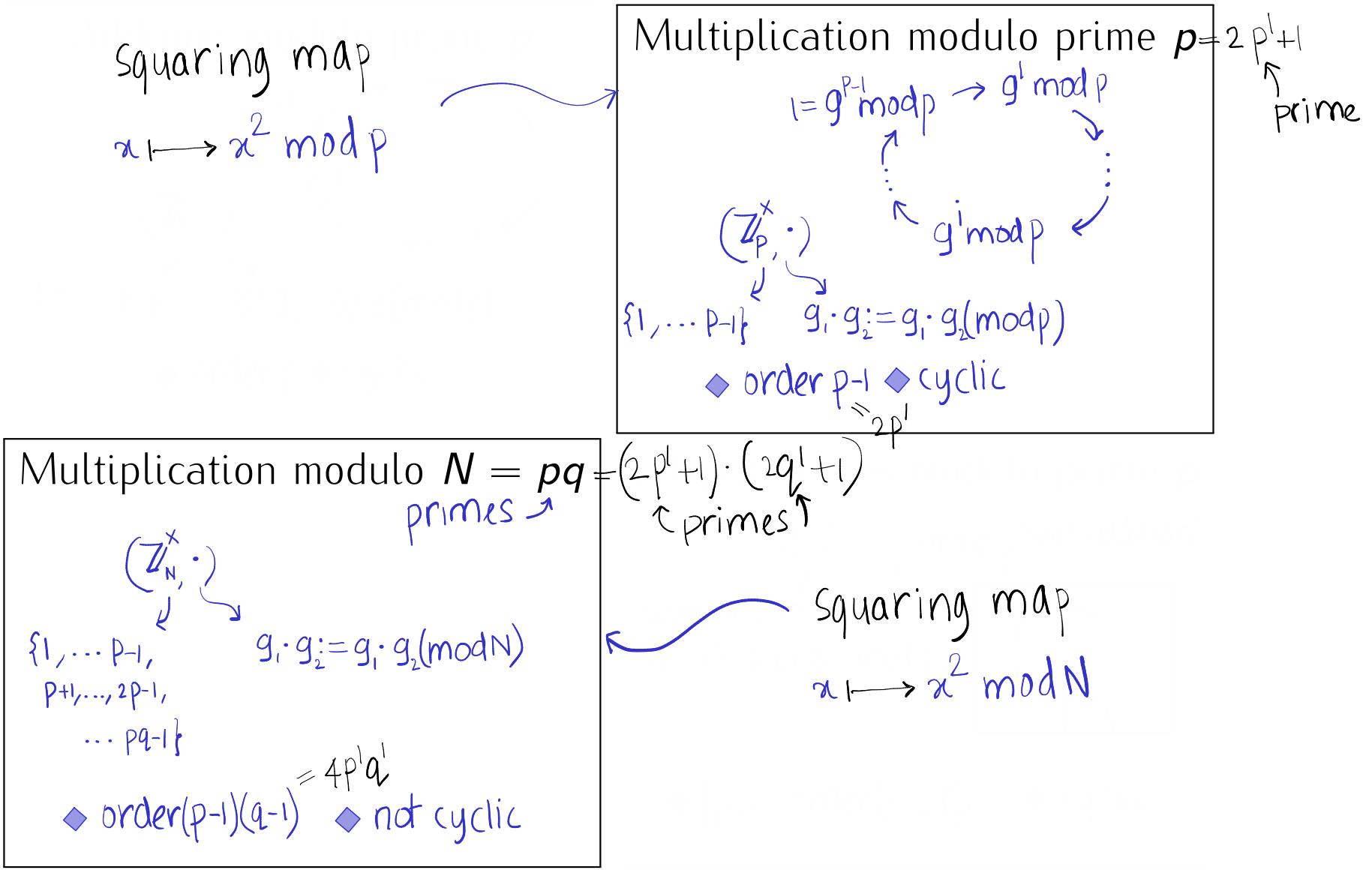

- order $p-1$ $= 2p'$
- cyclic

**Multiplication modulo $N = pq$** $= (2p'+1)\cdot(2q'+1)$ (primes)




- order $(p-1)(q-1)$ $= 4p'q'$
- not cyclic

# But First, Some Number Theory



# But First, Some Number Theory

Squaring map

$x \mapsto x^2 \bmod p$

Multiplication modulo prime $p = 2p'+1$ ($p'$ prime)

$1 = g^{p-1} \bmod p \to g^1 \bmod p$

$g^i \bmod p$

$(\mathbb{Z}_p^*, \cdot)$

$\{1, \dots p-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod p$

◆ order $p-1 = 2p'$ ◆ cyclic

Multiplication modulo $N = pq = (2p'+1)\cdot(2q'+1)$ (primes)

$(\mathbb{Z}_N^*, \cdot)$

$\{1, \dots p-1, p+1, \dots, 2p-1, \dots pq-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod N$

◆ order $(p-1)(q-1) = 4p'q'$ ◆ not cyclic

Squaring map

$x \mapsto x^2 \bmod N$

Recall: this is a OWF, inverting as hard as factoring $N$

# But First, Some Number Theory

Squaring map

$x \longmapsto x^2 \bmod p$

Multiplication modulo prime $p = 2p'+1$ ($p'$ prime)

$1 = g^{p-1} \bmod p \to g^1 \bmod p$

$g^i \bmod p$

$(\mathbb{Z}_p^*, \cdot)$

$\{1, \dots p-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod p$

◆ order $p-1 = 2p'$ ◆ cyclic

Multiplication modulo $N = pq = (2p'+1)\cdot(2q'+1)$ (primes)

$(\mathbb{Z}_N^*, \cdot)$

$\{1, \dots p-1, p+1, \dots, 2p-1, \dots pq-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod N$

◆ order $(p-1)(q-1) = 4p'q'$ ◆ not cyclic

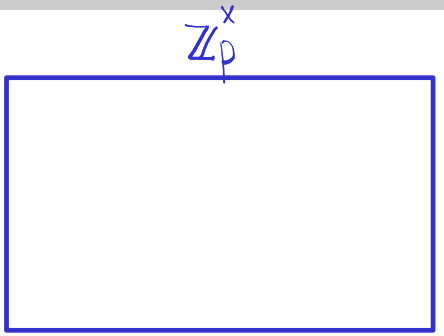
Squaring map

$x \longmapsto x^2 \bmod N$

Recall: this is a OWF, inverting as hard as factoring $N$

Looking ahead: our hard problem will be of form "is $y$ a square?"

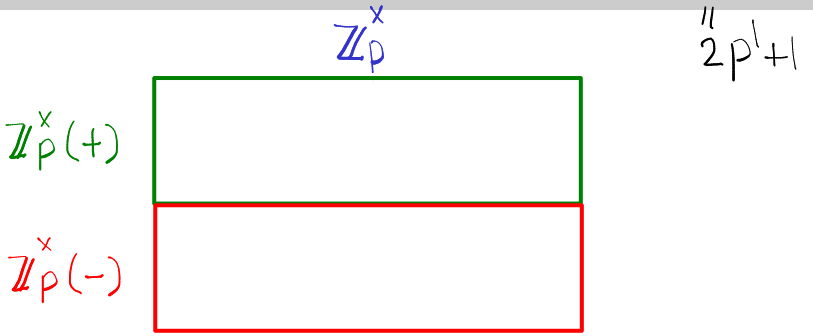
# Squaring Map in $(\mathbb{Z}_p^\times, \cdot)$




- 2-1 map since 2 is not invertible modulo $2p'$

  $\Rightarrow$ Only 1/2 the elements $\mathbb{Z}_p^\times(+) \subset \mathbb{Z}_p^\times$ have square roots

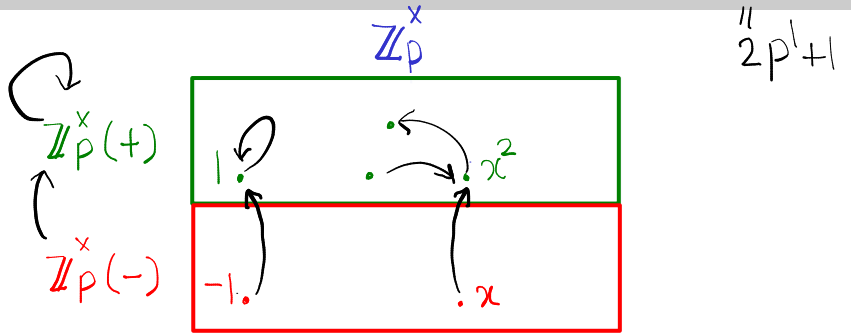
# Squaring Map in $(\mathbb{Z}_p^\times, \cdot)$




- 2-1 map since 2 is not invertible modulo $2p'$

  $\Rightarrow$ Only 1/2 the elements $\mathbb{Z}_p^\times(+) \subset \mathbb{Z}_p^\times$ have square roots

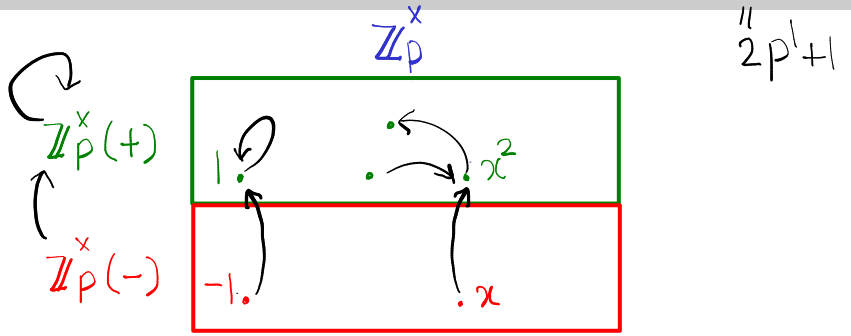
# Squaring Map in $(\mathbb{Z}_p^\times, \cdot)$




- 2-1 map since 2 is not invertible modulo $2p'$

  $\Rightarrow$ Only 1/2 the elements $\mathbb{Z}_p^\times(+) \subset \mathbb{Z}_p^\times$ have square roots

# Squaring Map in $(\mathbb{Z}_p^\times, \cdot)$

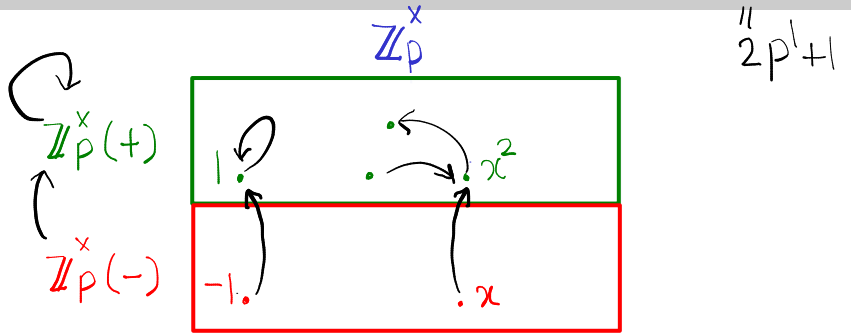



- 2-1 map since 2 is not invertible modulo $2p'$

  $\Rightarrow$ Only 1/2 the elements $\mathbb{Z}_p^\times(+) \subset \mathbb{Z}_p^\times$ have square roots

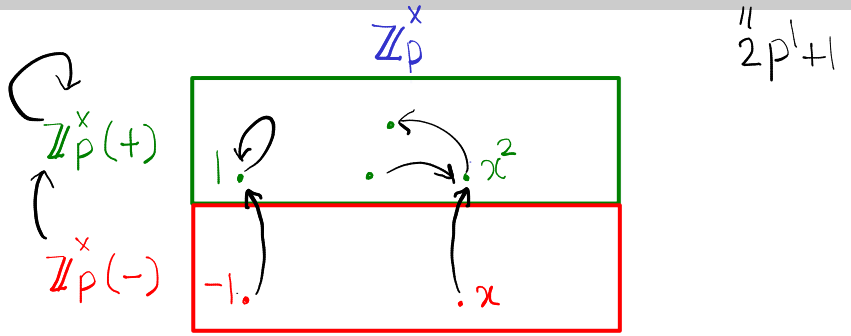
- Is it possible to *test* if $y \in \mathbb{Z}_p^\times(+)$?

# Squaring Map in $(\mathbb{Z}_p^\times, \cdot)$



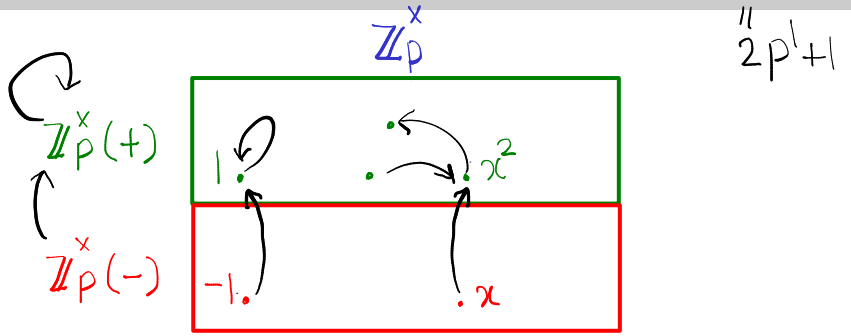

- 2-1 map since 2 is not invertible modulo $2p'$
  - $\Rightarrow$ Only 1/2 the elements $\mathbb{Z}_p^\times(+) \subset \mathbb{Z}_p^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_p^\times(+)$? Yes:
  - Compute discrete log $x$ of $y$ w.r.to *some* generator $g$
  - $y \in \mathbb{Z}_p^\times(+)$ if $x$ is *even*

# Squaring Map in $(\mathbb{Z}_p^\times, \cdot)$




- 2-1 map since 2 is not invertible modulo $2p'$
  - $\Rightarrow$ Only 1/2 the elements $\mathbb{Z}_p^\times(+) \subset \mathbb{Z}_p^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_p^\times(+)$? Yes:
  - Compute discrete log $x$ of $y$ w.r.to *some* generator $g$
  - $y \in \mathbb{Z}_p^\times(+)$ if $x$ is *even*
- Is it possible to *efficiently* test if $y \in \mathbb{Z}_p^\times(+)$? Yes:
  - Compute Legendre symbol $y^{(p-1)/2} \in \{\pm 1\}$
  - $y \in \mathbb{Z}_p^\times(+)$ iff its value is $+1$

# Squaring Map in $(\mathbb{Z}_p^\times, \cdot)$

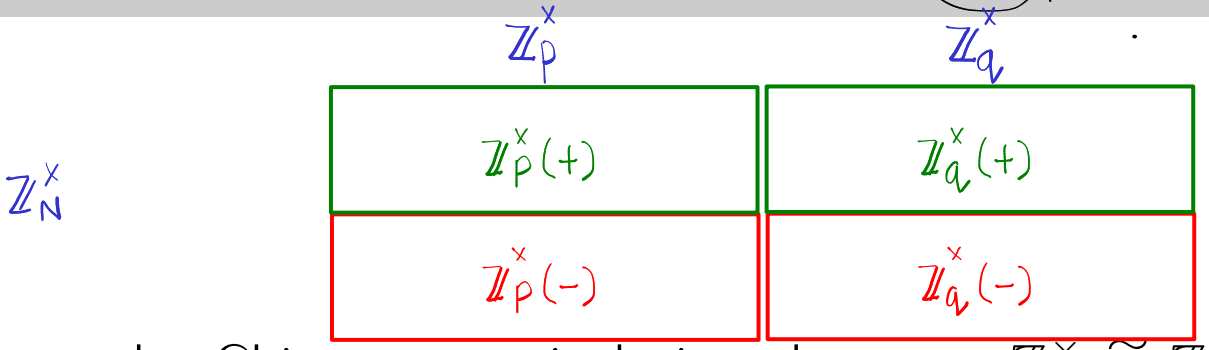



- 2-1 map since 2 is not invertible modulo $2p'$
  - $\Rightarrow$ Only 1/2 the elements $\mathbb{Z}_p^\times(+) \subset \mathbb{Z}_p^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_p^\times(+)$? Yes:
  - Compute discrete log $x$ of $y$ w.r.to *some* generator $g$
  - $y \in \mathbb{Z}_p^\times(+)$ if $x$ is *even*
- Is it possible to *efficiently* test if $y \in \mathbb{Z}_p^\times(+)$? Yes:
  - Compute Legendre symbol $y^{(p-1)/2} \in \{\pm 1\}$
  - $y \in \mathbb{Z}_p^\times(+)$ iff its value is $+1$

**Exercise 2 (Hint: Legendre symbol is multiplicative)**

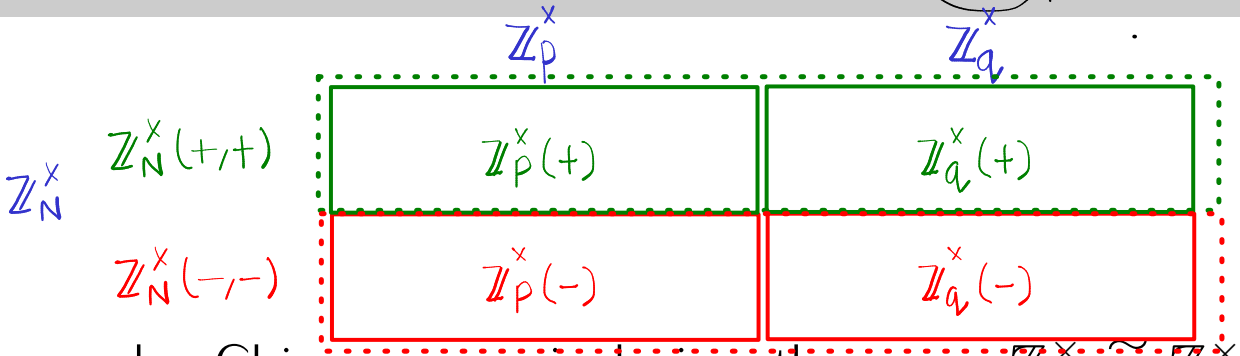
*Show that DDH assumption doesn't hold in* $(\mathbb{Z}_p^\times, \cdot)$

# Squaring Map in $(\mathbb{Z}_N^\times, \cdot)$



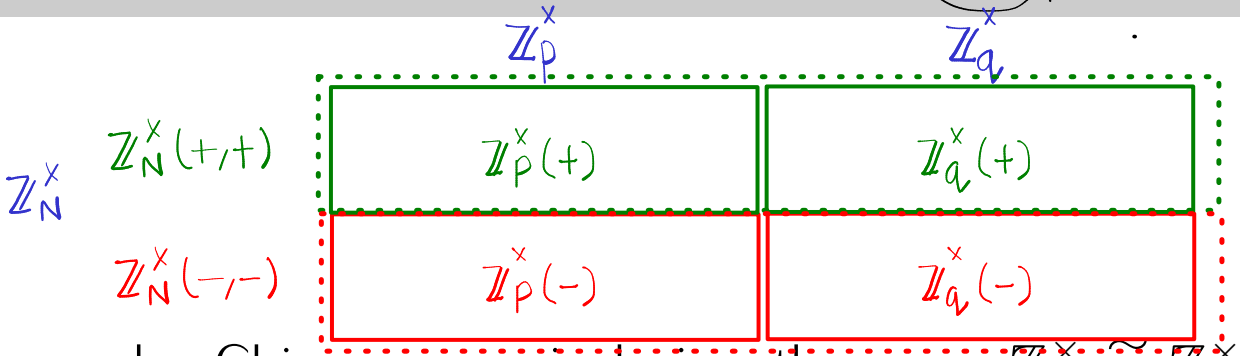

- 4–1 map by Chinese remaindering theorem: $\mathbb{Z}_N^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$
  $\Rightarrow$ Only 1/4 the elements $\mathbb{Z}_N^\times(+,+) \subset \mathbb{Z}_N^\times$ have square roots

# Squaring Map in $(\mathbb{Z}_N^\times, \cdot)$



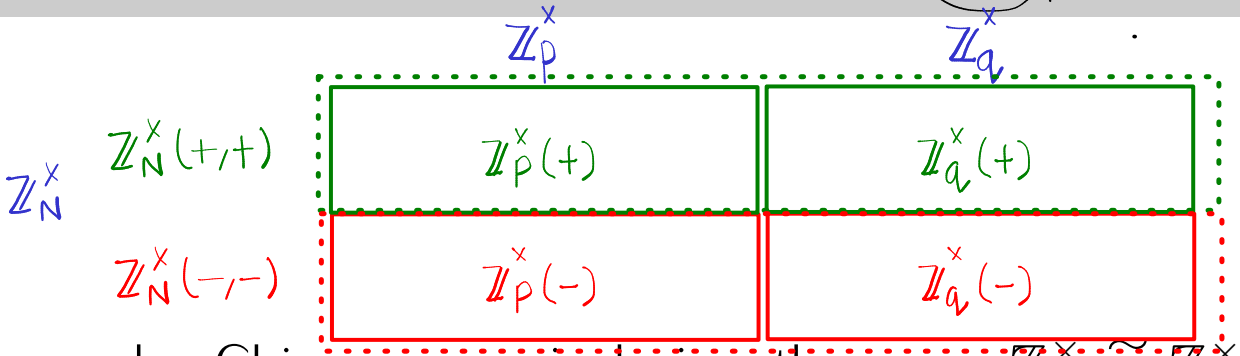

- 4–1 map by Chinese remaindering theorem: $\mathbb{Z}_N^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$
  $\Rightarrow$ Only 1/4 the elements $\mathbb{Z}_N^\times(+,+) \subset \mathbb{Z}_N^\times$ have square roots

# Squaring Map in $(\mathbb{Z}_N^\times, \cdot)$



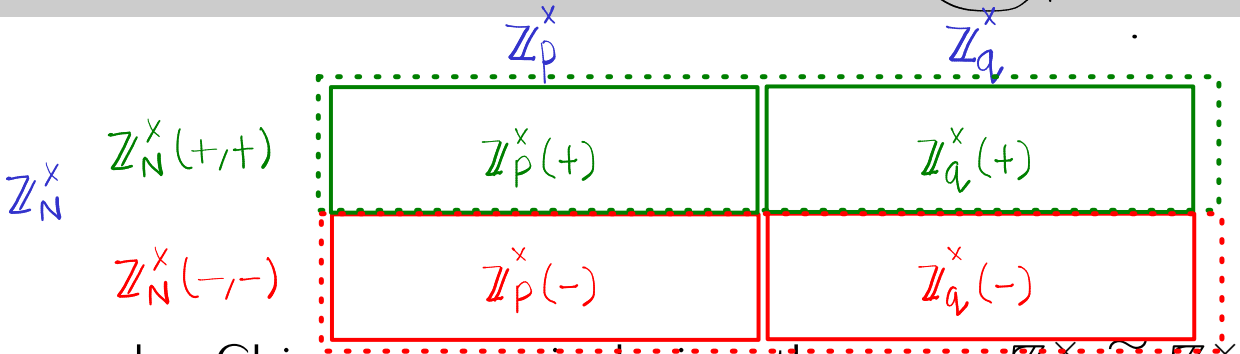

- 4–1 map by Chinese remaindering theorem: $\mathbb{Z}_N^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$
  $\Rightarrow$ Only 1/4 the elements $\mathbb{Z}_N^\times(+,+) \subset \mathbb{Z}_N^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_N^\times(+,+)$?

# Squaring Map in $(\mathbb{Z}_N^\times, \cdot)$



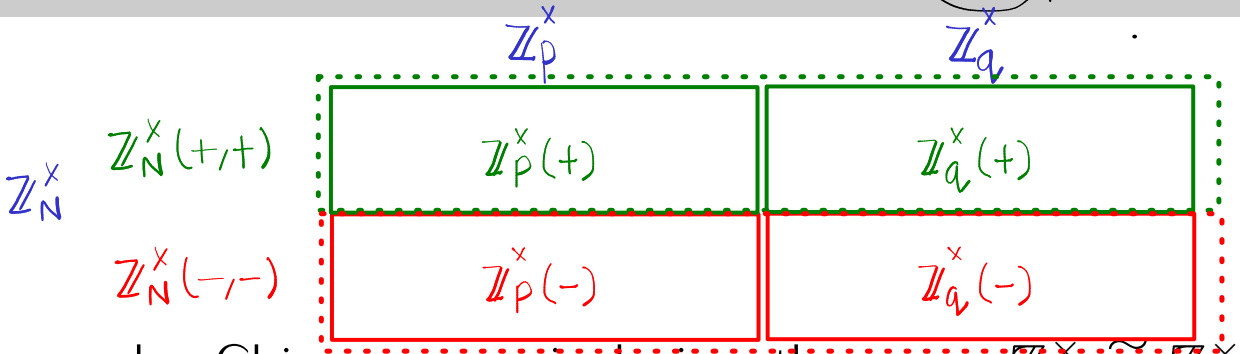

- 4–1 map by Chinese remaindering theorem: $\mathbb{Z}_N^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$
  - $\Rightarrow$ Only 1/4 the elements $\mathbb{Z}_N^\times(+,+) \subset \mathbb{Z}_N^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_N^\times(+,+)$? Yes:
  - Test if $y \in \mathbb{Z}_p^\times(+)$ *and* $y \in \mathbb{Z}_q^\times(+)$

# Squaring Map in $(\mathbb{Z}_N^\times, \cdot)$



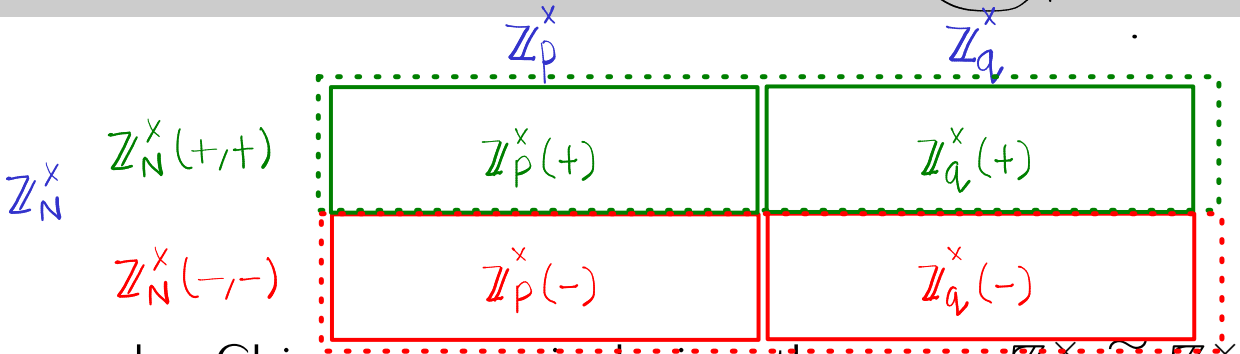

- 4–1 map by Chinese remaindering theorem: $\mathbb{Z}_N^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$
  - $\Rightarrow$ Only 1/4 the elements $\mathbb{Z}_N^\times(+,+) \subset \mathbb{Z}_N^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_N^\times(+,+)$? Yes:
  - Test if $y \in \mathbb{Z}_p^\times(+)$ *and* $y \in \mathbb{Z}_q^\times(+)$
- Is it possible to *efficiently test* if $y \in \mathbb{Z}_N^\times(+,+)$?

# Squaring Map in $(\mathbb{Z}_N^\times, \cdot)$




- 4–1 map by Chinese remaindering theorem: $\mathbb{Z}_N^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$
  - $\Rightarrow$ Only 1/4 the elements $\mathbb{Z}_N^\times(+,+) \subset \mathbb{Z}_N^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_N^\times(+,+)$? Yes:
  - Test if $y \in \mathbb{Z}_p^\times(+)$ *and* $y \in \mathbb{Z}_q^\times(+)$
- Is it possible to *efficiently test* if $y \in \mathbb{Z}_N^\times(+,+)$? Unclear
  - Can efficiently distinguish $\mathbb{Z}_N^\times(+,+) \cup \mathbb{Z}_N^\times(-,-)$ from $\mathbb{Z}_N^\times(-,+) \cup \mathbb{Z}_N^\times(+,-)$: compute Jacobi symbol

# Squaring Map in $(\mathbb{Z}_N^\times, \cdot)$




- 4–1 map by Chinese remaindering theorem: $\mathbb{Z}_N^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$
  - $\Rightarrow$ Only 1/4 the elements $\mathbb{Z}_N^\times(+,+) \subset \mathbb{Z}_N^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_N^\times(+,+)$? Yes:
  - Test if $y \in \mathbb{Z}_p^\times(+)$ *and* $y \in \mathbb{Z}_q^\times(+)$
- Is it possible to *efficiently test* if $y \in \mathbb{Z}_N^\times(+,+)$? Unclear
  - Can efficiently distinguish $\mathbb{Z}_N^\times(+,+) \cup \mathbb{Z}_N^\times(-,-)$ from $\mathbb{Z}_N^\times(-,+) \cup \mathbb{Z}_N^\times(+,-)$: compute Jacobi symbol

**Assumption 3 (Quadratic residuosity (QR) assumption in $(\mathbb{Z}_N^\times, \cdot)$)**

*The QR assumption w.r.to* $\mathsf{S}$ *holds if for all distinguishers* $\mathsf{D}$*, the following is negligible:*

$$\delta(n) := \Pr_{\substack{N \leftarrow \mathsf{S}(1^n) \\ y \leftarrow \mathbb{Z}_N^\times(+,+)}}[\mathsf{D}(N,y) = 0] - \Pr_{\substack{N \leftarrow \mathsf{S}(1^n) \\ y \leftarrow \mathbb{Z}_N^\times(-,-)}}[\mathsf{D}(N,y) = 0]$$

# Goldwasser-Micali Public-Key Bit Encryption

- Key idea: encode message in the "sign"
  - $0 \mapsto \mathbb{Z}_N^\times(+,+)$ and $1 \mapsto \mathbb{Z}_N^\times(-,-)$

# Goldwasser-Micali Public-Key Bit Encryption

- Key idea: encode message in the "sign"
  - $0 \mapsto \mathbb{Z}_N^\times(+,+)$ and $1 \mapsto \mathbb{Z}_N^\times(-,-)$
  - Exploit the fact that $-1 \in \mathbb{Z}_N^\times(-,-)$

## Pseudocode 4 (Goldwasser-Micali PKE over $(\mathbb{Z}_N^\times, \cdot)$)

- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample modulus with factors* $(N,(p,q)) \leftarrow \mathsf{S}(1^n)$
  2. *Output* $(pk := N, sk := (p,q))$

# Goldwasser-Micali Public-Key Bit Encryption

- Key idea: encode message in the "sign"
  - $0 \mapsto \mathbb{Z}_N^\times(+,+)$ and $1 \mapsto \mathbb{Z}_N^\times(-,-)$
  - Exploit the fact that $-1 \in \mathbb{Z}_N^\times(-,-)$

## Pseudocode 4 (Goldwasser-Micali PKE over $(\mathbb{Z}_N^\times, \cdot)$)

- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample modulus with factors* $(N,(p,q)) \leftarrow \mathsf{S}(1^n)$
  2. *Output* $(pk := N, sk := (p,q))$
- *Encryption* $\mathsf{Enc}(pk, m)$*:*
  1. *Sample random* $r \leftarrow \mathbb{Z}_N^\times$
  2. *Output* $c := (-1)^m \cdot r^2 \bmod N$

# Goldwasser-Micali Public-Key Bit Encryption

- Key idea: encode message in the "sign"
  - $0 \mapsto \mathbb{Z}_N^\times(+,+)$ and $1 \mapsto \mathbb{Z}_N^\times(-,-)$
  - Exploit the fact that $-1 \in \mathbb{Z}_N^\times(-,-)$

## Pseudocode 4 (Goldwasser-Micali PKE over $(\mathbb{Z}_N^\times, \cdot)$)

- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample modulus with factors* $(N,(p,q)) \leftarrow \mathsf{S}(1^n)$
  2. *Output* $(pk := N, sk := (p,q))$
- *Encryption* $\mathsf{Enc}(pk, m)$*:*
  1. *Sample random* $r \leftarrow \mathbb{Z}_N^\times$
  2. *Output* $c := (-1)^m \cdot r^2 \bmod N$
- *Decryption* $\mathsf{Dec}(sk, c)$*: output*

$$\begin{cases} 0 & \text{if } c \in \mathbb{Z}_N^\times(+,+) = \mathbb{Z}_p^\times(+) \cong \mathbb{Z}_q^\times(+) \\ 1 & \boxed{\text{otherwise}} \end{cases}$$

# Goldwasser-Micali Public-Key Bit Encryption

- Key idea: encode message in the "sign"
  - $0 \mapsto \mathbb{Z}_N^\times(+,+)$ and $1 \mapsto \mathbb{Z}_N^\times(-,-)$
  - Exploit the fact that $-1 \in \mathbb{Z}_N^\times(-,-)$

## Pseudocode 4 (Goldwasser-Micali PKE over $(\mathbb{Z}_N^\times, \cdot)$)

- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample modulus with factors* $(N,(p,q)) \leftarrow \mathsf{S}(1^n)$
  2. *Output* $(pk := N, sk := (p,q))$
- *Encryption* $\mathsf{Enc}(pk, m)$*:*
  1. *Sample random* $r \leftarrow \mathbb{Z}_N^\times$
  2. *Output* $c := (-1)^m \cdot r^2 \bmod N$
- *Decryption* $\mathsf{Dec}(sk, c)$*: output*

$$\begin{cases} 0 & \text{if } c \in \mathbb{Z}_N^\times(+,+) = \mathbb{Z}_p^\times(+) \cong \mathbb{Z}_q^\times(+) \\ 1 & \boxed{\text{otherwise}} \end{cases}$$

- *Correctness of decryption:*

# Goldwasser-Micali Public-Key Bit Encryption

- Key idea: encode message in the "sign"
  - $0 \mapsto \mathbb{Z}_N^\times(+,+)$ and $1 \mapsto \mathbb{Z}_N^\times(-,-)$
  - Exploit the fact that $-1 \in \mathbb{Z}_N^\times(-,-)$

## Pseudocode 4 (Goldwasser-Micali PKE over $(\mathbb{Z}_N^\times, \cdot)$)

- *Key generation* $\mathsf{Gen}(1^n)$*:*
  1. *Sample modulus with factors* $(N,(p,q)) \leftarrow \mathsf{S}(1^n)$
  2. *Output* $(pk := N, sk := (p,q))$
- *Encryption* $\mathsf{Enc}(pk, m)$*:*
  1. *Sample random* $r \leftarrow \mathbb{Z}_N^\times$
  2. *Output* $c := (-1)^m \cdot r^2 \bmod N$
- *Decryption* $\mathsf{Dec}(sk, c)$*: output*

$$\begin{cases} 0 & \text{if } c \in \mathbb{Z}_N^\times(+,+) = \mathbb{Z}_p^\times(+) \cong \mathbb{Z}_q^\times(+) \\ 1 & \text{otherwise} \end{cases}$$

- *Correctness of decryption:* since $r^2 \in \mathbb{Z}_N^\times(+,+)$, $c \in \mathbb{Z}_N^\times(+,+)$ iff $m = 0$

# Goldwasser-Micali PKE is CPA-secret

**Theorem 2 (QR → CPA-PKE)**

*Goldwasser-Micali PKE is CPA-secret under QR assumption.*

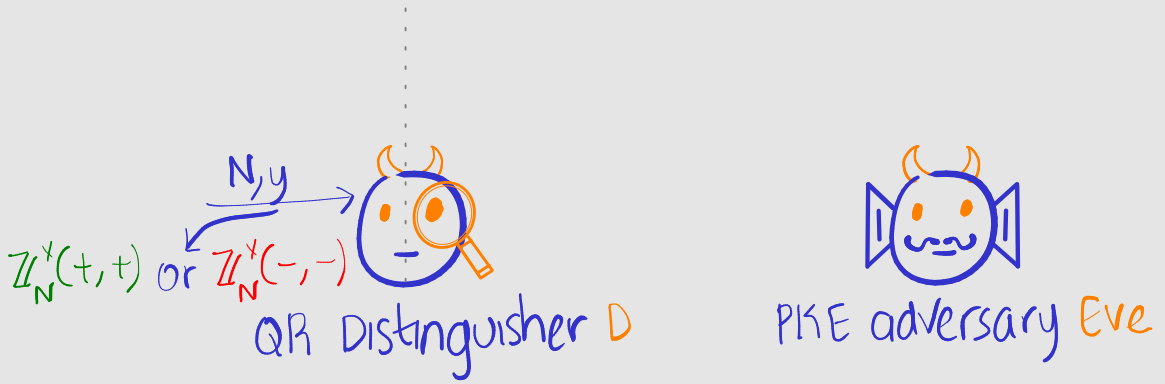
**Proof.** ∃D against QR ⇐ ∃Eve against PKE.


# Goldwasser-Micali PKE is CPA-secret

**Theorem 2 (QR → CPA-PKE)**

*Goldwasser-Micali PKE is CPA-secret under QR assumption.*

**Proof.** ∃D against QR ⇐ ∃Eve against PKE.

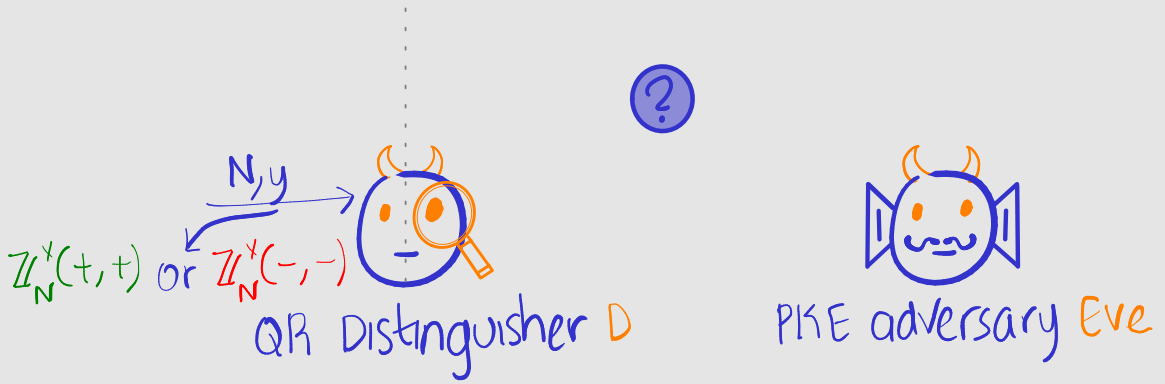


# Goldwasser-Micali PKE is CPA-secret

**Theorem 2 (QR → CPA-PKE)**

*Goldwasser-Micali PKE is CPA-secret under QR assumption.*

**Proof.** ∃D against QR ⇐ ∃Eve against PKE.

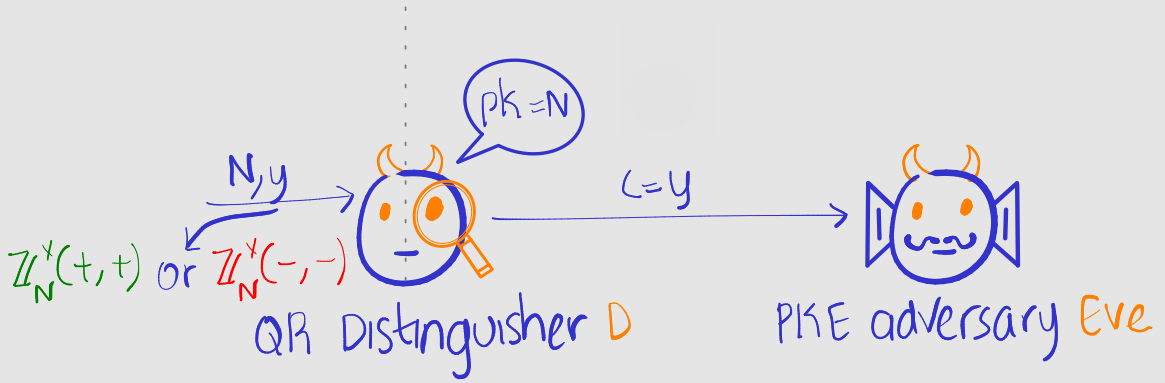


# Goldwasser-Micali PKE is CPA-secret

**Theorem 2 (QR → CPA-PKE)**

*Goldwasser-Micali PKE is CPA-secret under QR assumption.*

**Proof.** ∃D against QR ⇐ ∃Eve against PKE.

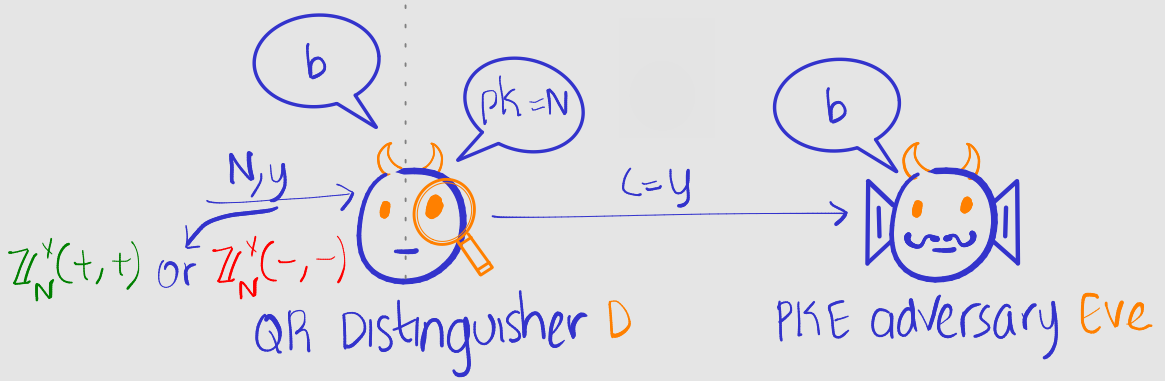


# Goldwasser-Micali PKE is CPA-secret

**Theorem 2 (QR → CPA-PKE)**

*Goldwasser-Micali PKE is CPA-secret under QR assumption.*

**Proof. ∃D against QR ⇐ ∃Eve against PKE.**

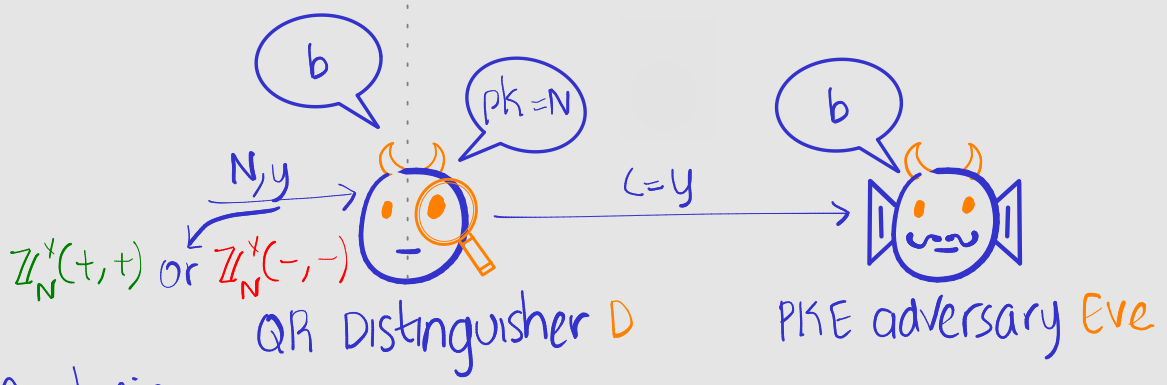


# Goldwasser-Micali PKE is CPA-secret

**Theorem 2 (QR → CPA-PKE)**

*Goldwasser-Micali PKE is CPA-secret under QR assumption.*

**Proof. ∃D against QR ⇐ ∃Eve against PKE.**




Analysis

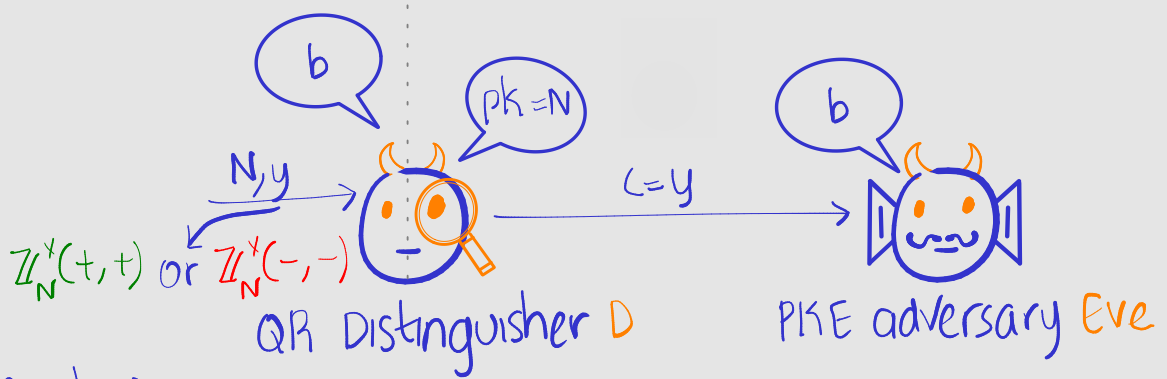
$$\left| \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(+,+)}} [D(N,y)=0] - \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(-,-)}} [D(N,y)=0] \right|$$

# Goldwasser-Micali PKE is CPA-secret

**Theorem 2 (QR → CPA-PKE)**

*Goldwasser-Micali PKE is CPA-secret under QR assumption.*

**Proof. ∃D against QR ⇐ ∃Eve against PKE.**




Analysis

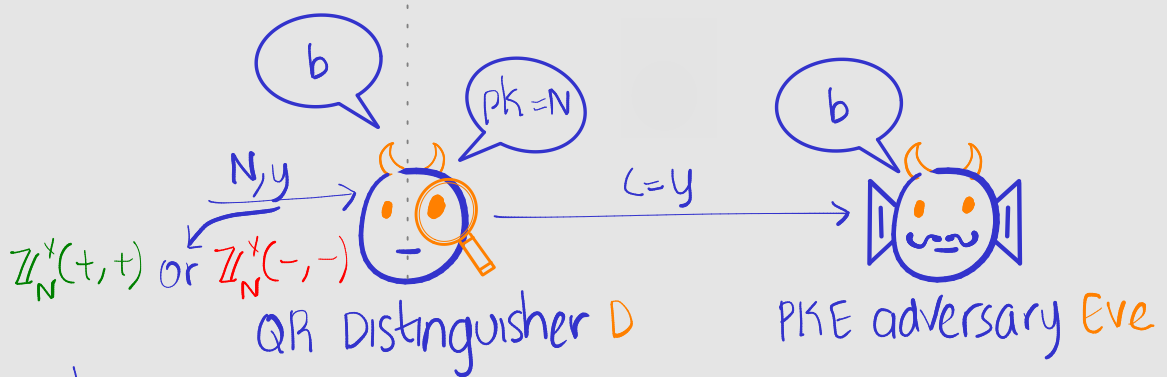
$$\left| \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^x(+,+)}} [D(N,y)=0] - \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^x(-,-)}} [D(N,y)=0] \right| = \left| \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^x(+,+)}} [Eve(N,y)=0] - \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^x(-,-)}} [Eve(N,y)=0] \right|$$

# Goldwasser-Micali PKE is CPA-secret

**Theorem 2 (QR → CPA-PKE)**

*Goldwasser-Micali PKE is CPA-secret under QR assumption.*

**Proof. ∃D against QR ⇐ ∃Eve against PKE.**




Analysis

$$\left| \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^x(+,+)}} [D(N,y)=0] - \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^x(-,-)}} [D(N,y)=0] \right| = \left| \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^x(+,+)}} [Eve(N,y)=0] - \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^x(-,-)}} [Eve(N,y)=0] \right|$$

$y \leftarrow \mathbb{Z}_N^x(+,+)$: distributed identically to $Enc(pk, 0)$

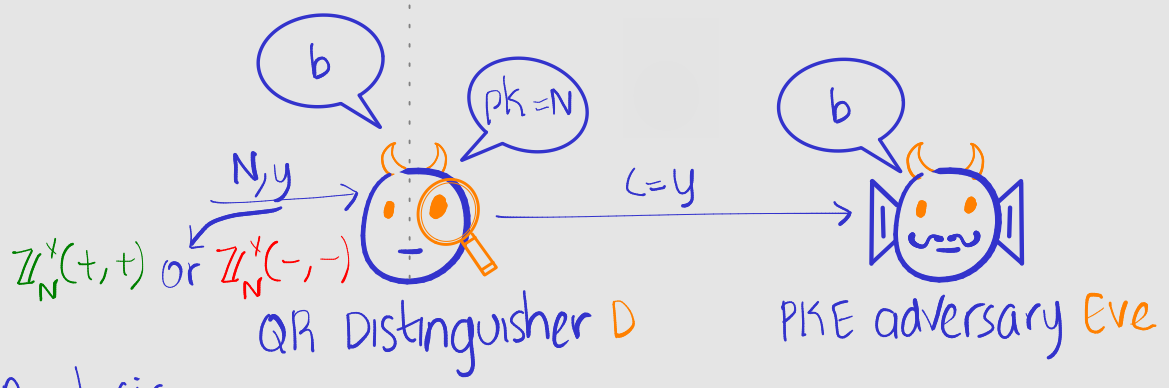
$y \leftarrow \mathbb{Z}_N^x(-,-)$: distributed identically to $Enc(pk, 1)$

# Goldwasser-Micali PKE is CPA-secret

**Theorem 2 (QR → CPA-PKE)**

*Goldwasser-Micali PKE is CPA-secret under QR assumption.*

**Proof. ∃D against QR ⇐ ∃Eve against PKE.**




Analysis

$$\left| \Pr_{\substack{N\leftarrow S(1^n)\\ y\leftarrow \mathbb{Z}_N^x(+,+)}} [D(N,y)=0] - \Pr_{\substack{N\leftarrow S(1^n)\\ y\leftarrow \mathbb{Z}_N^x(-,-)}} [D(N,y)=0] \right| = \left| \Pr_{\substack{N\leftarrow S(1^n)\\ y\leftarrow \mathbb{Z}_N^x(+,+)}} [Eve(N,y)=0] - \Pr_{\substack{N\leftarrow S(1^n)\\ y\leftarrow \mathbb{Z}_N^x(-,-)}} [Eve(N,y)=0] \right|$$

⇒ non-negligible

$y\leftarrow \mathbb{Z}_N^x(+,+)$: distributed identically to $Enc(pk,0)$

$y\leftarrow \mathbb{Z}_N^x(-,-)$: distributed identically to $Enc(pk,1)$ ∎

# What About Multiple Bits? Encrypt Bit by Bit

**Pseudocode 5 (Modified encryption algorithm)**

- *Encryption* $\mathsf{Enc}(pk, m =: (m_1, \ldots, m_\ell)$*:*
  1. *Sample random* $r_1, \ldots, r_\ell \leftarrow \mathbb{Z}_N^\times$
  2. *Output* $c_1, \ldots, c_\ell$*, where* $c_i := (-1)^{m_i} \cdot r_i^2 \bmod N$

# What About Multiple Bits? Encrypt Bit by Bit

**Pseudocode 5 (Modified encryption algorithm)**

- *Encryption* $\mathsf{Enc}(pk, m =: (m_1, \ldots, m_\ell)$*:*
  1. *Sample random* $r_1, \ldots, r_\ell \leftarrow \mathbb{Z}_N^\times$
  2. *Output* $c_1, \ldots, c_\ell$*, where* $c_i := (-1)^{m_i} \cdot r_i^2 \bmod N$

- Can be shown to be CPA-secret using *hybrid argument*
  - Loss in distinguishing advantage: $1/\ell$

# What About Multiple Bits? Encrypt Bit by Bit

**Pseudocode 5 (Modified encryption algorithm)**

- *Encryption* $\mathsf{Enc}(pk, m =: (m_1, \ldots, m_\ell)$*:*
  1. *Sample random* $r_1, \ldots, r_\ell \leftarrow \mathbb{Z}_N^\times$
  2. *Output* $c_1, \ldots, c_\ell$*, where* $c_i := (-1)^{m_i} \cdot r_i^2 \bmod N$

- Can be shown to be CPA-secret using *hybrid argument*
  - Loss in distinguishing advantage: $1/\ell$

- Can we do better? QR is random self-reducible
  1. Given instance of QR $\mapsto$ random instance of QR
  2. Solve given instance of QR $\Leftarrow$ solve random instance of QR

# What About Multiple Bits? Encrypt Bit by Bit

## Pseudocode 5 (Modified encryption algorithm)

- *Encryption* $\mathsf{Enc}(pk, m =: (m_1, \ldots, m_\ell)$*:*
  1. *Sample random* $r_1, \ldots, r_\ell \leftarrow \mathbb{Z}_N^\times$
  2. *Output* $c_1, \ldots, c_\ell$*, where* $c_i := (-1)^{m_i} \cdot r_i^2 \bmod N$

- Can be shown to be CPA-secret using *hybrid argument*
  - Loss in distinguishing advantage: $1/\ell$

- Can we do better? QR is random self-reducible
  1. Given instance of QR $\mapsto$ random instance of QR
  2. Solve given instance of QR $\Leftarrow$ solve random instance of QR

## Exercise 3

1. *Show that QR is random self-reducible*
2. *Can we exploit this to get a tight reduction for multiple bits?*

# To Recap Today's Lecture

- Task 4: Public-key encryption
  - Modelled setting and security (CPA secrecy)

# To Recap Today's Lecture

- Task 4: Public-key encryption
  - Modelled setting and security (CPA secrecy)
  - Saw two CPA-secret constructions, with proofs:
    - ElGamal PKE, based on DDH assumption
    - Goldwasser-Micali PKE, based on QR assumption

# To Recap Today's Lecture

- Task 4: Public-key encryption
  - Modelled setting and security (CPA secrecy)
  - Saw two CPA-secret constructions, with proofs:
    - ElGamal PKE, based on DDH assumption
    - Goldwasser-Micali PKE, based on QR assumption
  - Today's takeaway: two-message key-exchange $\leftrightarrow$ PKE

# To Recap Today's Lecture

- Task 4: Public-key encryption
  - Modelled setting and security (CPA secrecy)
  - Saw two CPA-secret constructions, with proofs:
    - ElGamal PKE, based on DDH assumption
    - Goldwasser-Micali PKE, based on QR assumption
  - Today's takeaway: two-message key-exchange $\leftrightarrow$ PKE
- Some open questions:
  1. CPA-PKE $\xrightarrow{?}$ CCA-PKE
     - Recall that CPA-SKE $\rightarrow$ CCA-SKE!
  2. DLog $\xrightarrow{?}$ CPA-PKE
     - We know CDH $\rightarrow$ CPA-PKE in the random-oracle model

# Next Lecture(s)

- This Friday (30/Aug): Quiz 1 crib session
  - Nivesh will discuss solutions before.

# Next Lecture(s)

- This Friday (30/Aug): Quiz 1 crib session
  - Nivesh will discuss solutions before.

- Next Tuesday (03/Sep): public-key encryption from lattices
  - Lattices and learning with errors (LWE) problem
  - Believed to be secure against quantum eavesdroppers!
  - Regev's PKE

# References

1. [KL14, Chapter 12] for details on this lecture.
2. Goldwasser–Micali PKE was presented in [GM84]. ElGamal PKE was presented in [ElG84].
3. Barak's exposition [Bar17] is an excellent source to understand the interplay of structure and hardness in PKE (and OWF)


Boaz Barak.

The complexity of public-key cryptography.

In *Tutorials on the Foundations of Cryptography*, pages 45–77. Springer International Publishing, 2017.


Taher ElGamal.

A public key cryptosystem and a signature scheme based on discrete logarithms.

In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18. Springer, Heidelberg, August 1984.


Shafi Goldwasser and Silvio Micali.

Probabilistic encryption.

*J. Comput. Syst. Sci.*, 28(2):270–299, 1984.


Jonathan Katz and Yehuda Lindell.

*Introduction to Modern Cryptography (3rd ed.)*.

Chapman and Hall/CRC, 2014.