# Fully-Homomorphic Encryption

Chethan Kamath

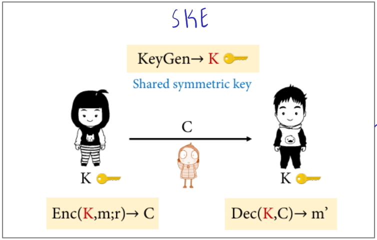
IITB TRUST LAB

DIGITAL : SECURE : RESPONSIBLE

ACM Summer School 2024, 7/Jun/2024

# Recall from Yesterday's Sessions

## SYMMETRIC KEY UNIVERSE



### SKE

KeyGen→ K

Shared symmetric key

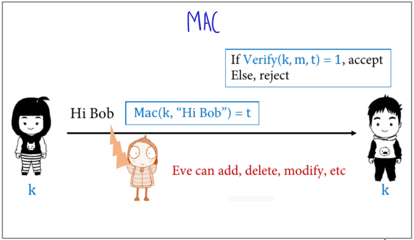
C

K

K

Enc(K,m;r)→ C

Dec(K,C)→ m'



### MAC

If Verify(k, m, t) = 1, accept
Else, reject

Hi Bob

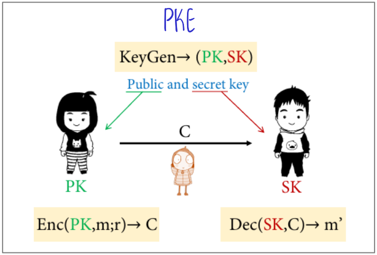
Mac(k, "Hi Bob") = t

Eve can add, delete, modify, etc

k

k

## PUBLIC KEY UNIVERSE



### PKE

KeyGen→ (PK,SK)

Public and secret key

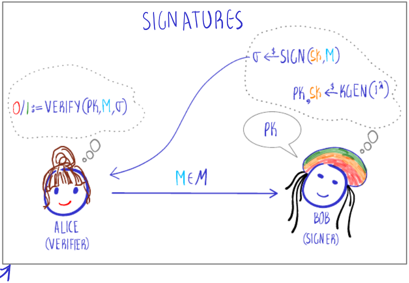
C

PK

SK

Enc(PK,m;r)→ C

Dec(SK,C)→ m'

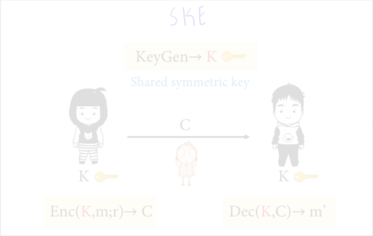


### SIGNATURES

$\sigma \xleftarrow{\$} \mathrm{SIGN}(SK, M)$
$PK, SK \xleftarrow{\$} \mathrm{KGEN}(1^{\lambda})$
$0/1 := \mathrm{VERIFY}(PK, M, \sigma)$
PK
$M \in \mathcal{M}$
ALICE
(VERIFIER)
BOB
(SIGNER)

# Recall from Yesterday's Sessions

## SYMMETRIC KEY UNIVERSE



SKE

KeyGen→ K

Shared symmetric key

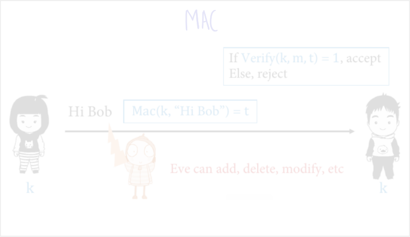
C

K

K

Enc(K,m;r)→ C

Dec(K,C)→ m'



MAC

If Verify(k, m, t) = 1, accept
Else, reject

Hi Bob

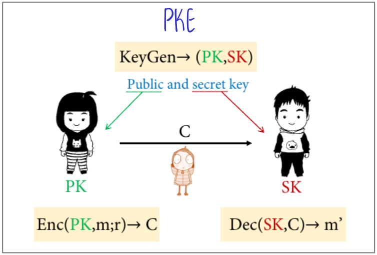
Mac(k, "Hi Bob") = t

Eve can add, delete, modify, etc

k

k

## PUBLIC KEY UNIVERSE



PKE

KeyGen→ (PK,SK)

Public and secret key

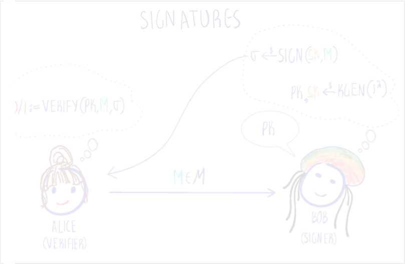
C

PK

SK

Enc(PK,m;r)→ C

Dec(SK,C)→ m'



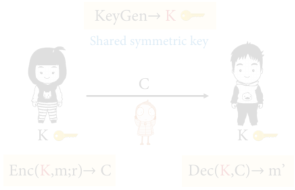
SIGNATURES

$\sigma \xleftarrow{\$} \mathrm{SIGN}(SK,M)$
$PK, SK \xleftarrow{\$} \mathrm{KGEN}(1^\lambda)$
$0/1 := \mathrm{VERIFY}(PK,M,\sigma)$
PK
M
ALICE (VERIFIER)
BOB (SIGNER)

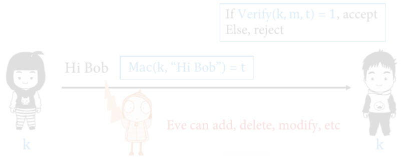
# Recall from Yesterday's Sessions

## SYMMETRIC KEY UNIVERSE

### SKE

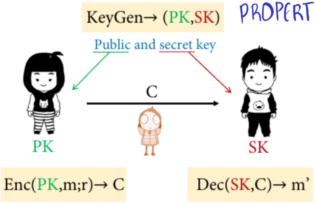


### MAC

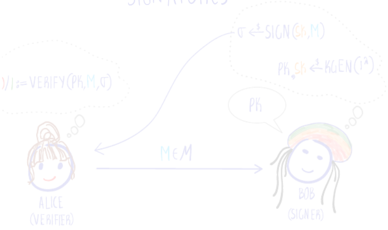

If Verify(k, m, t) = 1, accept
Else, reject
Hi Bob
Mac(k, "Hi Bob") = t
Eve can add, delete, modify, etc
k
k

## PUBLIC KEY UNIVERSE

### PKE W/ HOMOMORPHIC PROPERTIES



### SIGNATURES



# Plan for this Session

Homomorphic Encryption

# Plan for this Session

Homomorphic Encryption

Fully-Homomorphic Encryption (FHE)

# Plan for this Session

Homomorphic Encryption

Fully-Homomorphic Encryption (FHE)

Learning with Errors (LWE)

# Plan for this Session

Homomorphic Encryption

Fully-Homomorphic Encryption (FHE)

Learning with Errors (LWE)

Gentry-Sahai-Waters FHE from LWE

# Plan for this Session

Homomorphic Encryption

Fully-Homomorphic Encryption (FHE)

Learning with Errors (LWE)

Gentry-Sahai-Waters FHE from LWE

Wrapping Up

# Plan for this Session

Homomorphic Encryption

Fully-Homomorphic Encryption (FHE)

Learning with Errors (LWE)

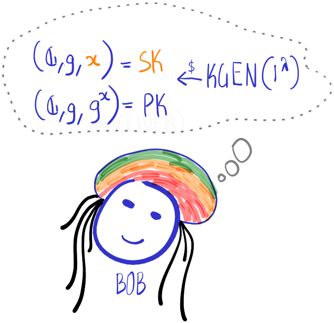
Gentry-Sahai-Waters FHE from LWE

Wrapping Up

# Example 1: Elgamal Encryption

- What happens when we multiply ciphertexts?
- Is it possible to compute sum of plaintexts?

# Example 1: Elgamal Encryption

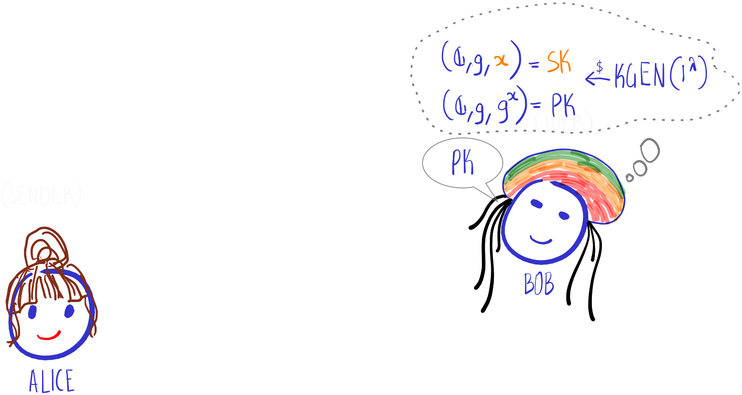


$$(\mathbb{G}, g, x) = SK \xleftarrow{\$} KGEN(1^\lambda)$$

$$(\mathbb{G}, g, g^x) = PK$$

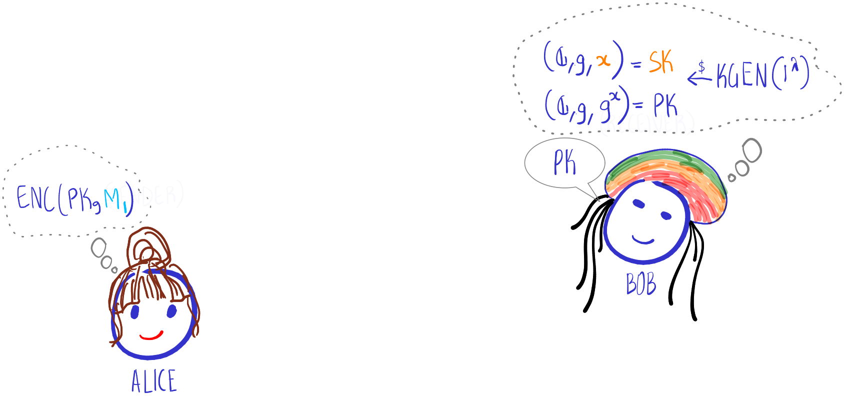
- What happens when we multiply ciphertexts?
- Is it possible to compute sum of plaintexts?

# Example 1: Elgamal Encryption


$(\mathbb{G}, g, x) = SK \xleftarrow{\$} KGEN(1^\lambda)$
$(\mathbb{G}, g, g^x) = PK$
PK
(RECEIVER)
BOB
(SENDER)
ALICE

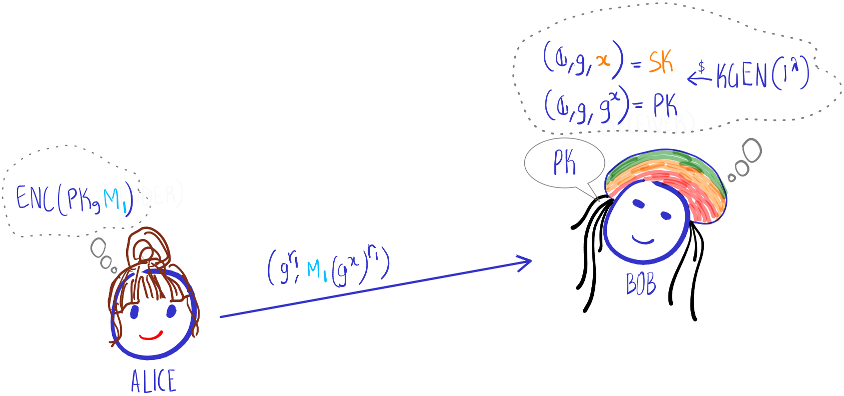
- What happens when we multiply ciphertexts?
- Is it possible to compute sum of plaintexts?

# Example 1: Elgamal Encryption


$(\mathbb{G}, g, x) = SK \overset{\$}{\leftarrow} KGEN(1^\lambda)$
$(\mathbb{G}, g, g^x) = PK$
(RECEIVER)
PK
BOB
$ENC(PK, M_1)$
(SENDER)
ALICE

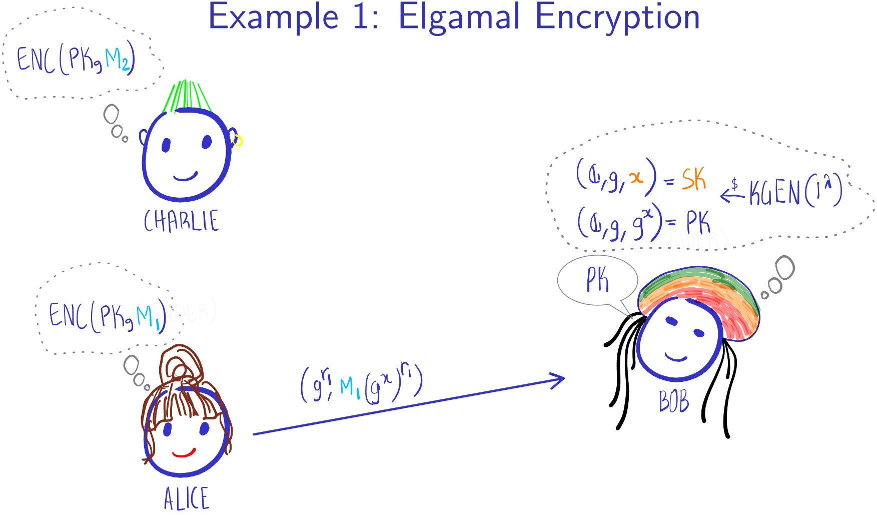
- What happens when we multiply ciphertexts?
- Is it possible to compute sum of plaintexts?

# Example 1: Elgamal Encryption


$(\mathbb{G}, g, x) = SK \xleftarrow{\$} KGEN(1^\lambda)$
$(\mathbb{G}, g, g^x) = PK$
PK
$ENC(PK, M_1)$
$(g^{r_1}, M_1 (g^x)^{r_1})$
ALICE
BOB

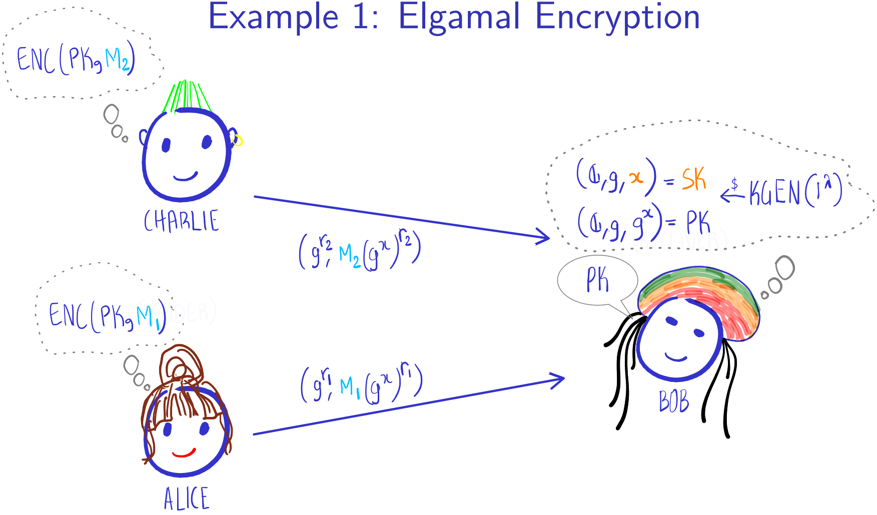
- What happens when we multiply ciphertexts?
- Is it possible to compute sum of plaintexts?

# Example 1: Elgamal Encryption


$\mathrm{ENC}(PK_g, M_2)$
CHARLIE
$(\mathbb{G}, g, x) = SK \xleftarrow{\$} \mathrm{KGEN}(1^\lambda)$
$(\mathbb{G}, g, g^x) = PK$
(RECEIVER)
PK
$\mathrm{ENC}(PK_g, M_1)$
(SENDER)
$(g^{r_1}, M_1 (g^x)^{r_1})$
BOB
ALICE

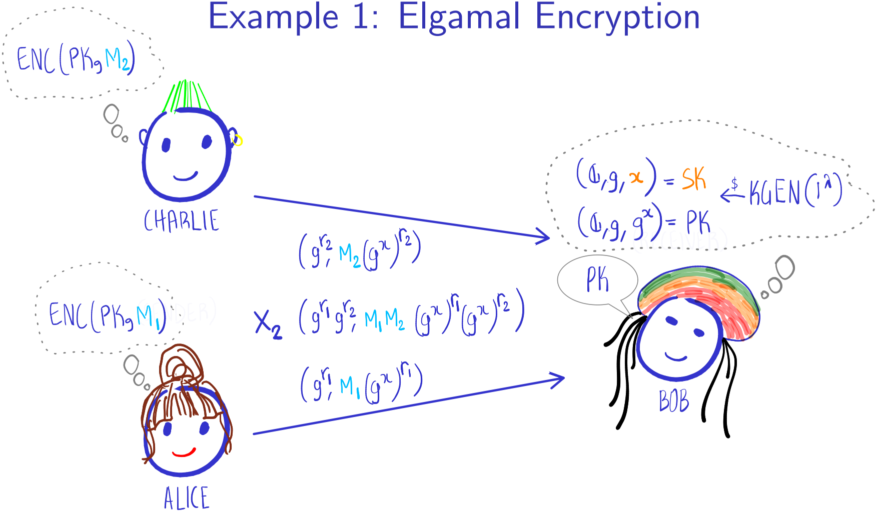
- What happens when we multiply ciphertexts?
- Is it possible to compute sum of plaintexts?

# Example 1: Elgamal Encryption


$\mathrm{ENC}(PK, M_2)$
CHARLIE
$(g^{r_2}, M_2 (g^x)^{r_2})$
$(\mathbb{G}, g, x) = SK \xleftarrow{\$} \mathrm{KGEN}(1^\lambda)$
$(\mathbb{G}, g, g^x) = PK$
PK
$\mathrm{ENC}(PK, M_1)$
$(g^{r_1}, M_1 (g^x)^{r_1})$
BOB
ALICE

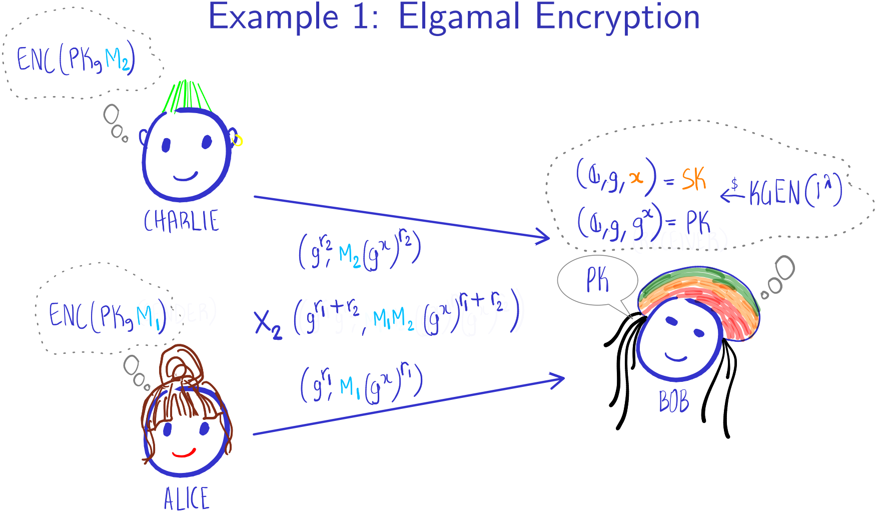
- What happens when we multiply ciphertexts?
- Is it possible to compute sum of plaintexts?

# Example 1: Elgamal Encryption


$ENC(PK, M_2)$
CHARLIE
$(g^{r_2}, M_2 (g^x)^{r_2})$
$X_2$ $(g^{r_1} g^{r_2}, M_1 M_2 (g^x)^{r_1} (g^x)^{r_2})$
$(g^{r_1}, M_1 (g^x)^{r_1})$
$ENC(PK, M_1)$
ALICE
$(\mathbb{G}, g, x) = SK \xleftarrow{\$} KGEN(1^\lambda)$
$(\mathbb{G}, g, g^x) = PK$
PK
BOB

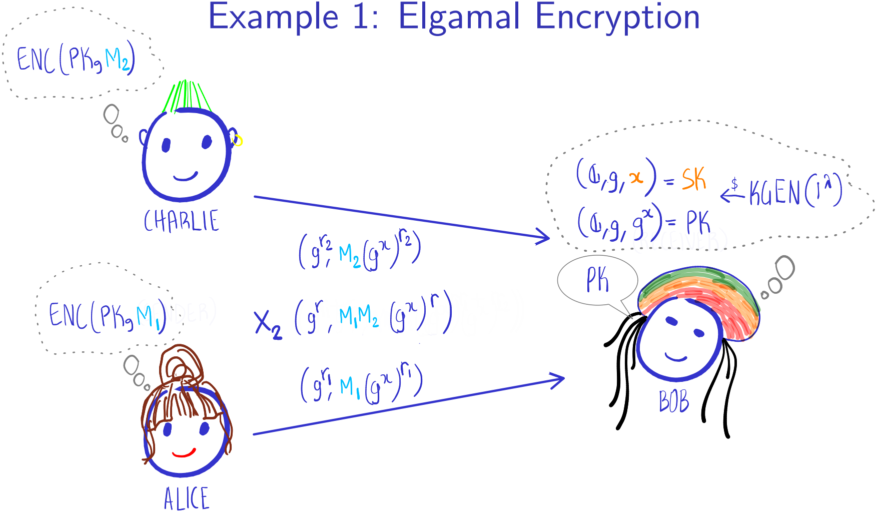
- What happens when we multiply ciphertexts?
- Is it possible to compute sum of plaintexts?

# Example 1: Elgamal Encryption


$ENC(PK_g, M_2)$
CHARLIE
$ENC(PK_g, M_1)$
ALICE
$(g^{r_2}, M_2(g^x)^{r_2})$
$X_2 \; (g^{r_1+r_2}, M_1M_2 (g^x)^{r_1+r_2})$
$(g^{r_1}, M_1(g^x)^{r_1})$
$(\mathbb{G}, g, x) = SK \xleftarrow{\$} KGEN(1^\lambda)$
$(\mathbb{G}, g, g^x) = PK$
PK
BOB

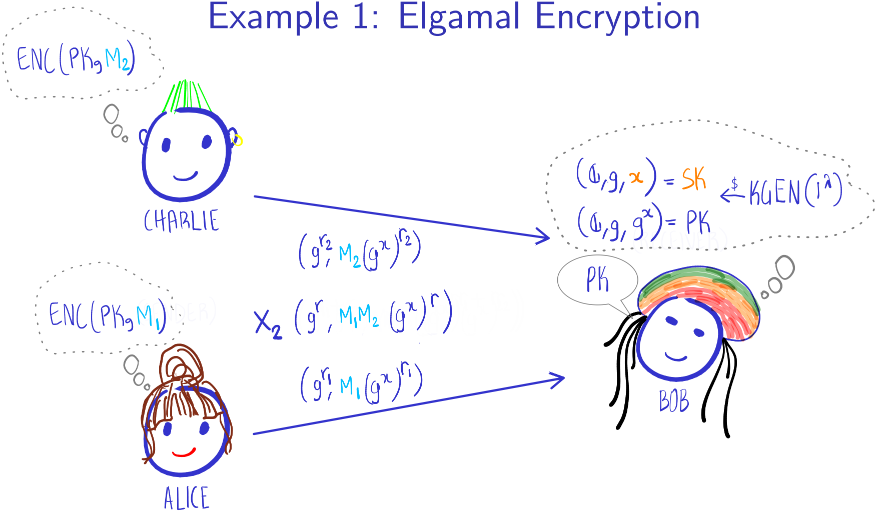
- What happens when we multiply ciphertexts?
- Is it possible to compute sum of plaintexts?

# Example 1: Elgamal Encryption


ENC(PK, $M_2$)
CHARLIE
$(\mathbb{G}, g, x) = SK \xleftarrow{\$} KGEN(1^\lambda)$
$(\mathbb{G}, g, g^x) = PK$
PK
BOB
$(g^{r_2}, M_2 (g^x)^{r_2})$
$\times_2$ $(g^r, M_1 M_2 (g^x)^r)$
$(g^{r_1}, M_1 (g^x)^{r_1})$
ENC(PK, $M_1$)
ALICE

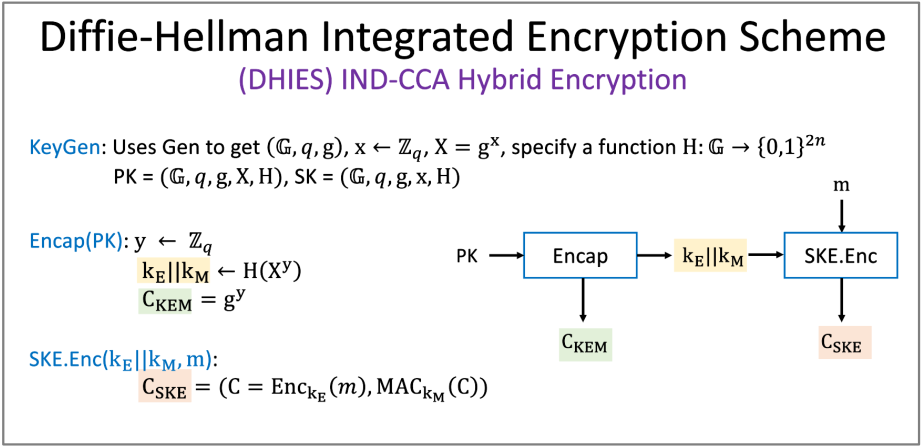
- What happens when we multiply ciphertexts?
- Is it possible to compute sum of plaintexts?

# Example 1: Elgamal Encryption


$\mathrm{ENC}(PK, M_2)$
CHARLIE
$(g^{r_2}, M_2 (g^x)^{r_2})$
$\mathrm{ENC}(PK, M_1)$
$X_2 \ (g^r, M_1 M_2 (g^x)^r)$
$(g^{r_1}, M_1 (g^x)^{r_1})$
ALICE
$(\mathbb{G}, g, x) = SK \xleftarrow{\$} \mathrm{KGEN}(1^\lambda)$
$(\mathbb{G}, g, g^x) = PK$
PK
BOB

- What happens when we multiply ciphertexts?
- Is it possible to compute sum of plaintexts?

# What about DHIES?



## Diffie-Hellman Integrated Encryption Scheme

### (DHIES) IND-CCA Hybrid Encryption

KeyGen: Uses Gen to get $(\mathbb{G}, q, \mathrm{g})$, $\mathrm{x} \leftarrow \mathbb{Z}_q$, $\mathrm{X} = \mathrm{g}^{\mathrm{x}}$, specify a function $\mathrm{H}: \mathbb{G} \rightarrow \{0,1\}^{2n}$

PK = $(\mathbb{G}, q, \mathrm{g}, \mathrm{X}, \mathrm{H})$, SK = $(\mathbb{G}, q, \mathrm{g}, \mathrm{x}, \mathrm{H})$

Encap(PK): $\mathrm{y} \leftarrow \mathbb{Z}_q$

$\mathrm{k_E}||\mathrm{k_M} \leftarrow \mathrm{H}(\mathrm{X}^{\mathrm{y}})$

$\mathrm{C_{KEM}} = \mathrm{g}^{\mathrm{y}}$

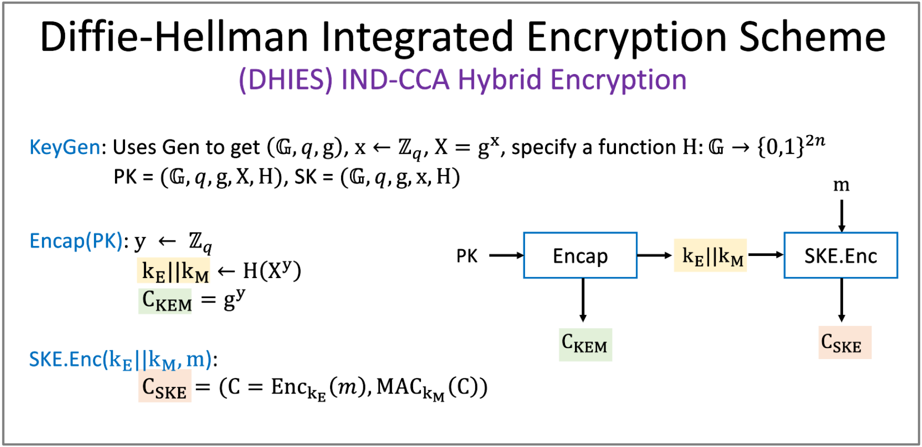
SKE.Enc($\mathrm{k_E}||\mathrm{k_M}, \mathrm{m}$):

$\mathrm{C_{SKE}} = (\mathrm{C} = \mathrm{Enc}_{\mathrm{k_E}}(m), \mathrm{MAC}_{\mathrm{k_M}}(\mathrm{C}))$

Exercise 1

*What happens when we (say) XOR ciphertexts?*

# What about DHIES?



## Diffie-Hellman Integrated Encryption Scheme

(DHIES) IND-CCA Hybrid Encryption

KeyGen: Uses Gen to get $(\mathbb{G}, q, g)$, $x \leftarrow \mathbb{Z}_q$, $X = g^x$, specify a function $H: \mathbb{G} \rightarrow \{0,1\}^{2n}$

PK = $(\mathbb{G}, q, g, X, H)$, SK = $(\mathbb{G}, q, g, x, H)$

Encap(PK): $y \leftarrow \mathbb{Z}_q$

$k_E || k_M \leftarrow H(X^y)$

$C_{KEM} = g^y$

SKE.Enc($k_E || k_M$, m):

$C_{SKE} = (C = \mathrm{Enc}_{k_E}(m), \mathrm{MAC}_{k_M}(C))$

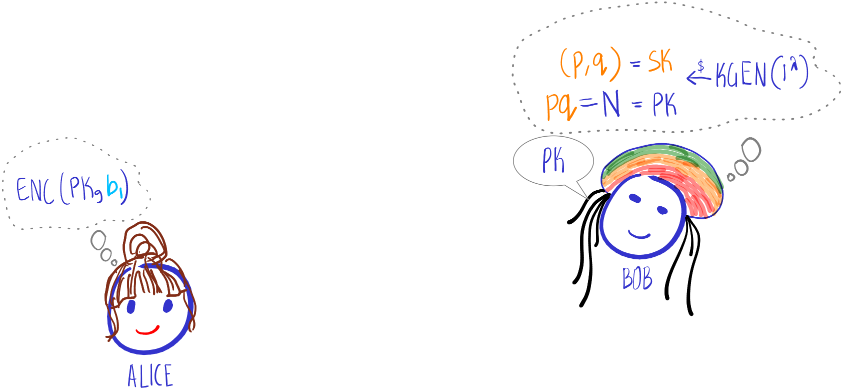
**Exercise 1**

*What happens when we (say) XOR ciphertexts?*

# Example 2: Goldwasser-Micali Bit Encryption

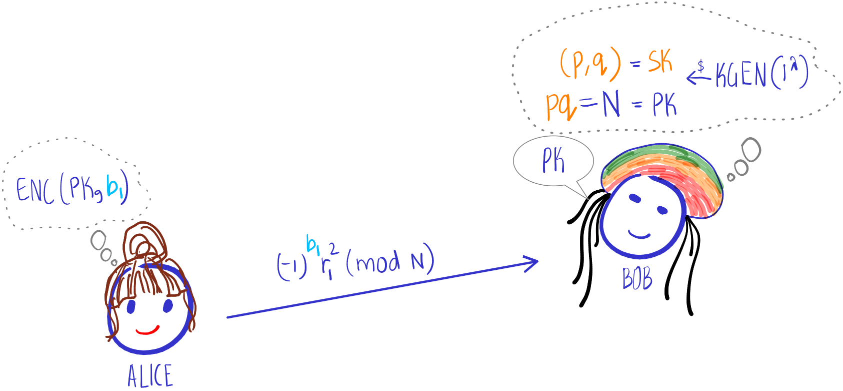
- What happens when we multiply ciphertexts?
- Is it possible compute product of plaintexts (modulo 2)?

# Example 2: Goldwasser-Micali Bit Encryption



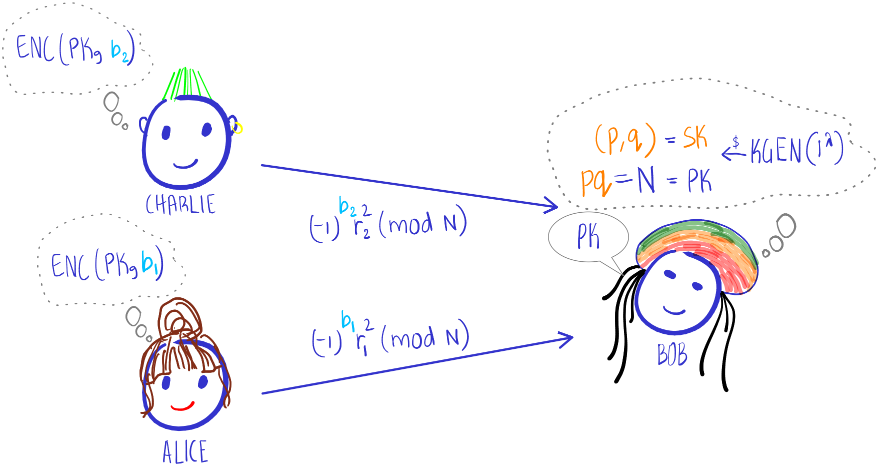
- What happens when we multiply ciphertexts?
- Is it possible compute product of plaintexts (modulo 2)?

# Example 2: Goldwasser-Micali Bit Encryption



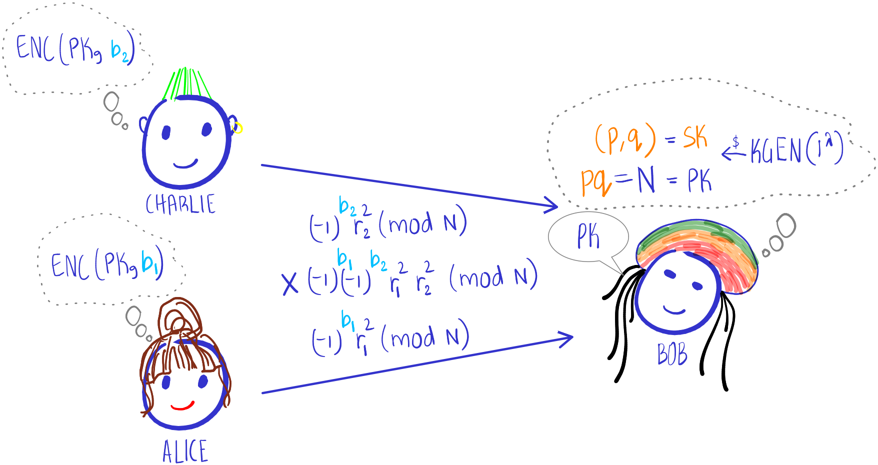
- What happens when we multiply ciphertexts?
- Is it possible compute product of plaintexts (modulo 2)?

# Example 2: Goldwasser-Micali Bit Encryption


$ENC(PK_g, b_2)$
CHARLIE
$ENC(PK_g, b_1)$
ALICE
$(p,q) = SK \overset{\$}{\leftarrow} KGEN(1^\lambda)$
$pq = N = PK$
PK
BOB
$(-1)^{b_2} r_2^2 \pmod N$
$(-1)^{b_1} r_1^2 \pmod N$

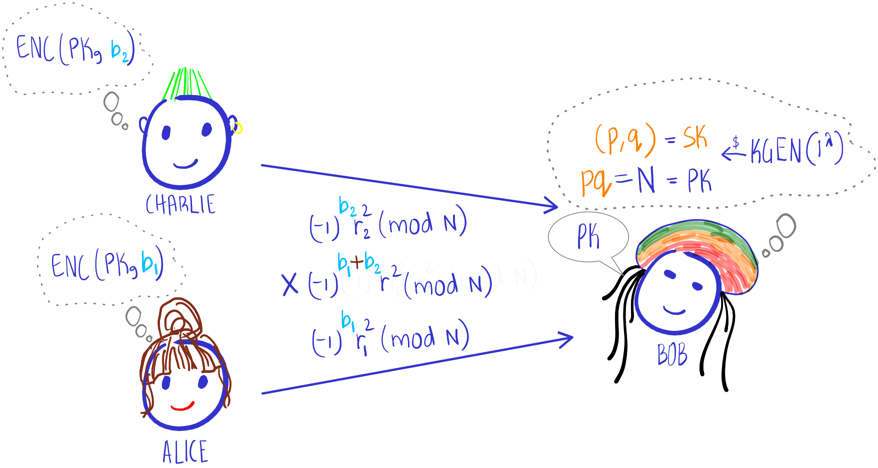
- What happens when we multiply ciphertexts?
- Is it possible compute product of plaintexts (modulo 2)?

# Example 2: Goldwasser-Micali Bit Encryption


$\mathrm{ENC}(PK_g\, b_2)$
CHARLIE
$\mathrm{ENC}(PK_g\, b_1)$
ALICE
$(-1)^{b_2} r_2^2 \pmod N$
$\times\ (-1)^{b_1}(-1)^{b_2} r_1^2 r_2^2 \pmod N$
$(-1)^{b_1} r_1^2 \pmod N$
$(p,q) = SK \xleftarrow{\$} KGEN(1^\lambda)$
$pq = N = PK$
PK
BOB

- What happens when we multiply ciphertexts?
- Is it possible compute product of plaintexts (modulo 2)?

# Example 2: Goldwasser-Micali Bit Encryption


$\mathrm{ENC}(PK_B, b_2)$
CHARLIE
$\mathrm{ENC}(PK_B, b_1)$
ALICE
$(p,q) = SK \xleftarrow{\$} \mathrm{KGEN}(1^\lambda)$
$pq = N = PK$
PK
BOB
$(-1)^{b_2} r_2^2 \pmod N$
$\times\ (-1)^{b_1+b_2} r^2 \pmod N$
$(-1)^{b_1} r_1^2 \pmod N$

- What happens when we multiply ciphertexts?
- Is it possible compute product of plaintexts (modulo 2)?

# Plan for this Session

Homomorphic Encryption

Fully-Homomorphic Encryption (FHE)

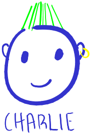
Learning with Errors (LWE)

Gentry-Sahai-Waters FHE from LWE

Wrapping Up

# Defining Homomorphic Encryption

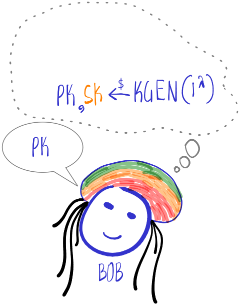
- Public-key encryption with additional *evaluation* algorithm
  - Four-tuple of algorithms: $(\texttt{KGEN}, \texttt{ENC}, \texttt{DEC}, \texttt{EVAL})$



- FHE supports evaluation of *arbitrary* functions $F$
- Levelled FHE supports function of depth $L$

# Defining Homomorphic Encryption

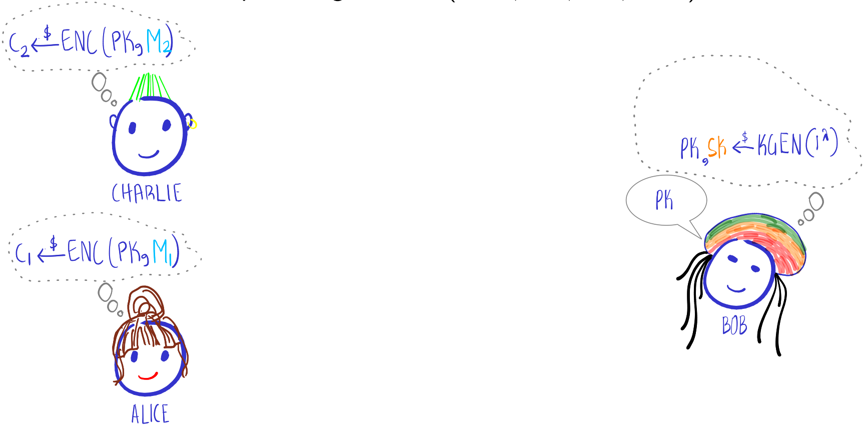
- Public-key encryption with additional *evaluation* algorithm
  - Four-tuple of algorithms: $(\mathtt{KGEN}, \mathtt{ENC}, \mathtt{DEC}, \mathtt{EVAL})$



- FHE supports evaluation of *arbitrary* functions $F$
- Levelled FHE supports function of depth $L$

# Defining Homomorphic Encryption

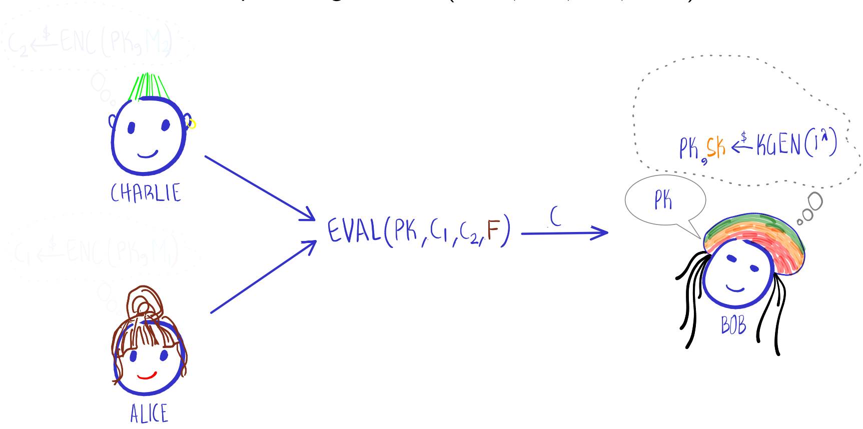
- Public-key encryption with additional *evaluation* algorithm
  - Four-tuple of algorithms: $(\mathtt{KGEN}, \mathtt{ENC}, \mathtt{DEC}, \mathtt{EVAL})$


$C_2 \xleftarrow{\$} \mathrm{ENC}(PK, M_2)$
CHARLIE
$C_1 \xleftarrow{\$} \mathrm{ENC}(PK, M_1)$
ALICE
$PK, SK \xleftarrow{\$} \mathrm{KGEN}(1^\lambda)$
PK
BOB

- FHE supports evaluation of *arbitrary* functions $F$
- Levelled FHE supports function of depth $L$

# Defining Homomorphic Encryption

- Public-key encryption with additional *evaluation* algorithm
  - Four-tuple of algorithms: $(\mathtt{KGEN}, \mathtt{ENC}, \mathtt{DEC}, \mathtt{EVAL})$

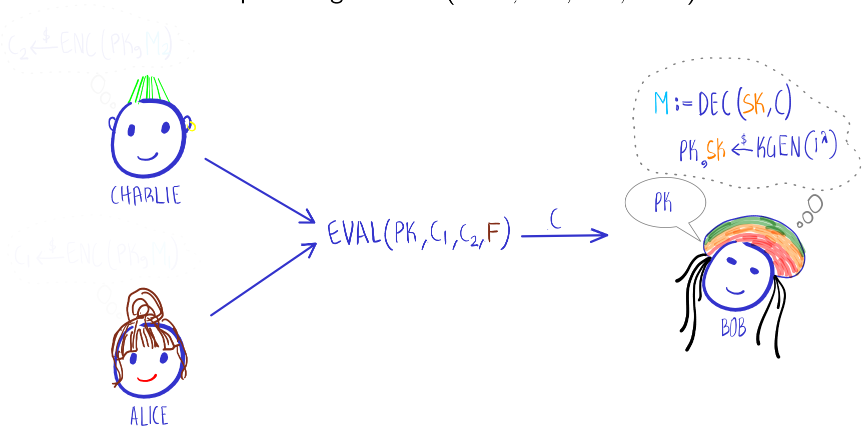


$C_2 \xleftarrow{\$} ENC(PK, M_2)$
CHARLIE
$C_1 \xleftarrow{\$} ENC(PK, M_1)$
ALICE
$EVAL(PK, C_1, C_2, F)$
$C$
$PK, SK \xleftarrow{\$} KGEN(1^\lambda)$
PK
BOB

- FHE supports evaluation of *arbitrary* functions $F$
- Levelled FHE supports function of depth $L$

# Defining Homomorphic Encryption

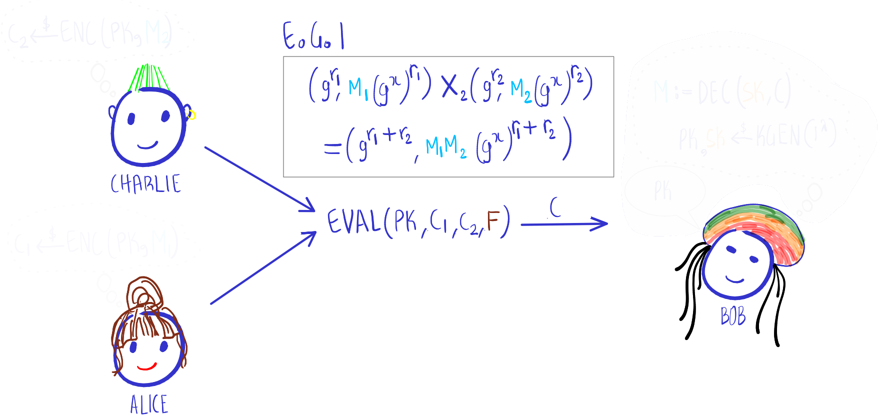
- Public-key encryption with additional *evaluation* algorithm
  - Four-tuple of algorithms: (KGEN, ENC, DEC, EVAL)



- FHE supports evaluation of *arbitrary* functions $F$
- Levelled FHE supports function of depth $L$

# Defining Homomorphic Encryption

- Public-key encryption with additional *evaluation* algorithm
  - Four-tuple of algorithms: $(\mathtt{KGEN}, \mathtt{ENC}, \mathtt{DEC}, \mathtt{EVAL})$



EoGol

$$\left(g^{r_1}, M_1 (g^x)^{r_1}\right) \times_2 \left(g^{r_2}, M_2 (g^x)^{r_2}\right)$$
$$= \left(g^{r_1+r_2}, M_1 M_2 (g^x)^{r_1+r_2}\right)$$

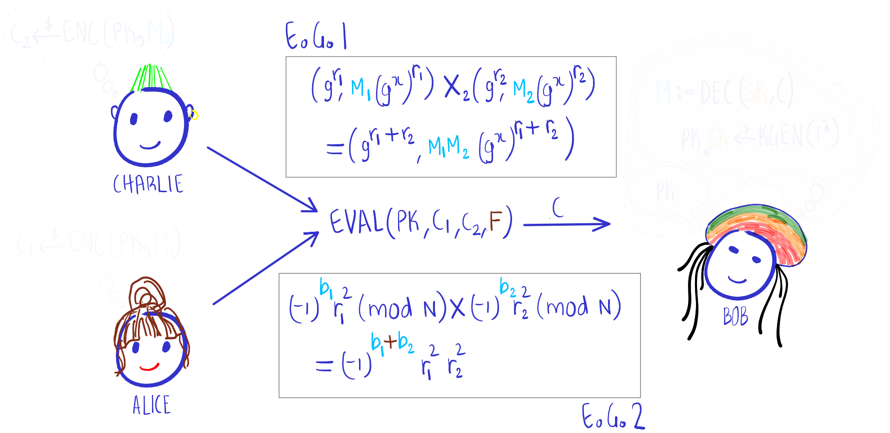
$\mathrm{EVAL}(PK, C_1, C_2, F) \xrightarrow{C}$

$C_2 \xleftarrow{\$} \mathrm{ENC}(PK, M_2)$
CHARLIE
$C_1 \xleftarrow{\$} \mathrm{ENC}(PK, M_1)$
ALICE
$M := \mathrm{DEC}(SK, C)$
$PK, SK \xleftarrow{\$} \mathrm{KGEN}(1^\lambda)$
PK
BOB

- FHE supports evaluation of *arbitrary* functions $F$
- Levelled FHE supports function of depth $L$

# Defining Homomorphic Encryption

- Public-key encryption with additional *evaluation* algorithm
    - Four-tuple of algorithms: $(\texttt{KGEN}, \texttt{ENC}, \texttt{DEC}, \texttt{EVAL})$



$C_2 \xleftarrow{\$} \mathrm{ENC}(PK, M_2)$
CHARLIE
$C_1 \xleftarrow{\$} \mathrm{ENC}(PK, M_1)$
ALICE
$M := \mathrm{DEC}(SK, C)$
$PK, SK \xleftarrow{\$} \mathrm{KGEN}(1^\lambda)$
PK
BOB

E.G.1

$$(g^{r_1}, M_1 (g^x)^{r_1}) \times_2 (g^{r_2}, M_2 (g^x)^{r_2})$$
$$= (g^{r_1+r_2}, M_1 M_2 (g^x)^{r_1+r_2})$$

$$\mathrm{EVAL}(PK, C_1, C_2, F) \xrightarrow{C}$$

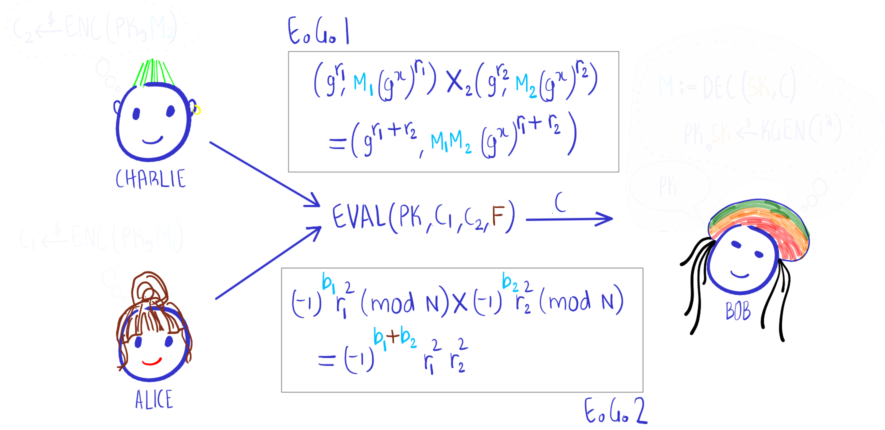
$$(-1)^{b_1} r_1^2 \pmod N \times (-1)^{b_2} r_2^2 \pmod N$$
$$= (-1)^{b_1+b_2} r_1^2 r_2^2$$

E.G.2

- FHE supports evaluation of *arbitrary* functions $F$
- Levelled FHE supports function of depth $L$

# Defining Homomorphic Encryption

- Public-key encryption with additional *evaluation* algorithm
  - Four-tuple of algorithms: $(\mathtt{KGEN}, \mathtt{ENC}, \mathtt{DEC}, \mathtt{EVAL})$



$C_2 \xleftarrow{\$} \mathrm{ENC}(PK, M_2)$
CHARLIE
$C_1 \xleftarrow{\$} \mathrm{ENC}(PK, M_1)$
ALICE
EVAL(PK, C_1, C_2, F)
C
$M := \mathrm{DEC}(SK, C)$
$PK, SK \xleftarrow{\$} \mathrm{KGEN}(1^\lambda)$
PK
BOB

E.G. 1

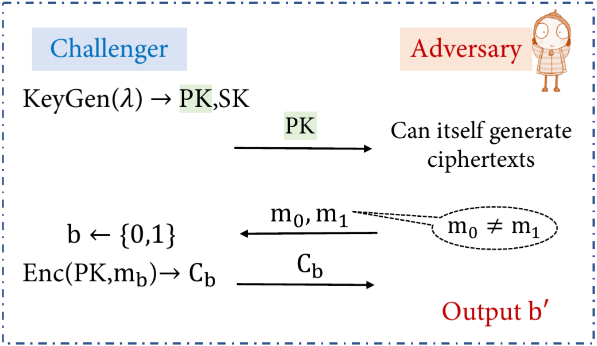
$$(g^{r_1}, M_1 (g^x)^{r_1}) \times_2 (g^{r_2}, M_2 (g^x)^{r_2}) = (g^{r_1+r_2}, M_1 M_2 (g^x)^{r_1+r_2})$$

$$(-1)^{b_1} r_1^2 (\mathrm{mod}\ N) \times (-1)^{b_2} r_2^2 (\mathrm{mod}\ N) = (-1)^{b_1+b_2} r_1^2 r_2^2$$

E.G. 2

- FHE supports evaluation of *arbitrary* functions $F$
- Levelled FHE supports function of depth $L$

# Security Model: IND-CPA for PKE



Challenger

Adversary

KeyGen($\lambda$) → PK,SK

PK

Can itself generate ciphertexts

$m_0, m_1$

$m_0 \neq m_1$

$b \leftarrow \{0,1\}$

Enc(PK,$m_b$)→ $C_b$

$C_b$

Output $b'$

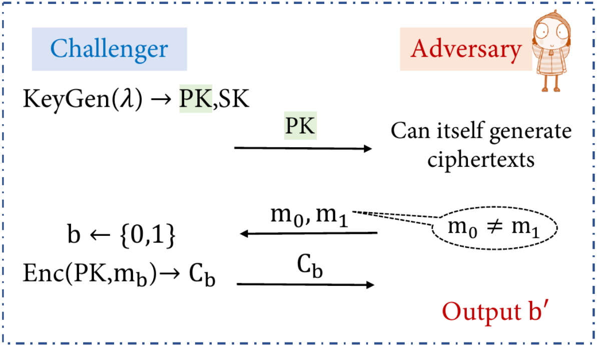
IND-CPA Security (Chosen Plaintext Attack)

$$\forall \text{ PPT adversaries, } \Pr[b' = b] \leq \frac{1}{2} + \mathrm{negl}(\lambda)$$

Exercise 2 (IND-CCA)

*Can FHE be IND-CCA secure?*

# Security Model: IND-CPA for PKE



Challenger

Adversary

$\mathrm{KeyGen}(\lambda) \rightarrow \mathrm{PK}, \mathrm{SK}$

$\xrightarrow{\mathrm{PK}}$

Can itself generate ciphertexts

$\xleftarrow{m_0, m_1}$

$m_0 \neq m_1$

$b \leftarrow \{0,1\}$

$\mathrm{Enc}(\mathrm{PK}, m_b) \rightarrow C_b$

$\xrightarrow{C_b}$

Output $b'$

IND-CPA Security (Chosen Plaintext Attack)

$$\forall \text{ PPT adversaries, } \Pr[b' = b] \leq \frac{1}{2} + \mathrm{negl}(\lambda)$$

## Exercise 2 (IND-CCA)

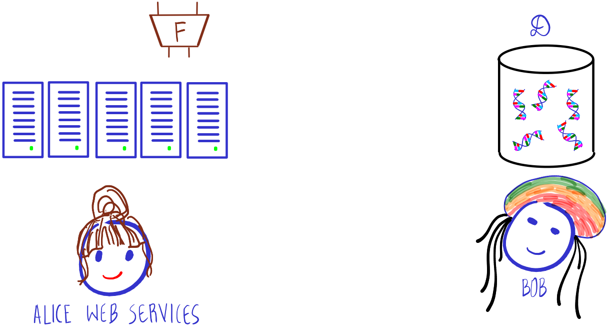
*Can FHE be IND-CCA secure?*

# What is FHE Useful for?

- Privacy-preserving outsourcing of computation

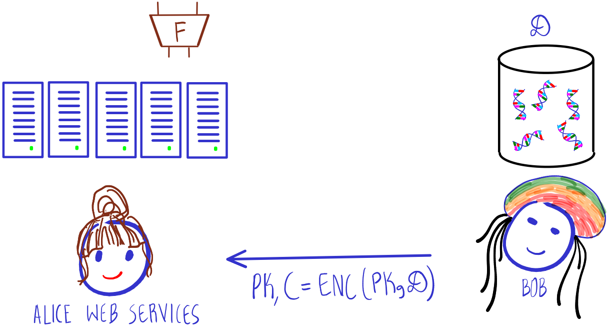
# What is FHE Useful for?

- Privacy-preserving outsourcing of computation



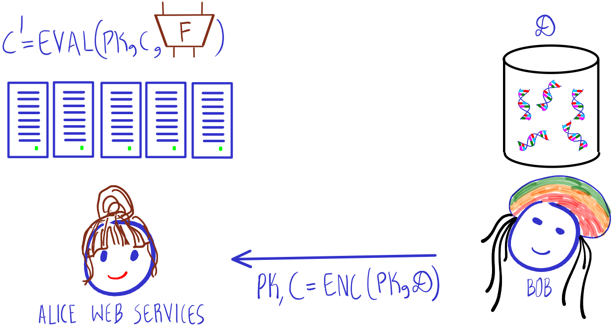
# What is FHE Useful for?

- Privacy-preserving outsourcing of computation



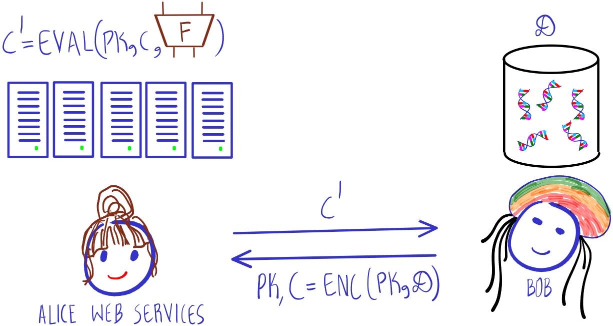
# What is FHE Useful for?

- Privacy-preserving outsourcing of computation



# What is FHE Useful for?

- Privacy-preserving outsourcing of computation



# Plan for this Session

Homomorphic Encryption

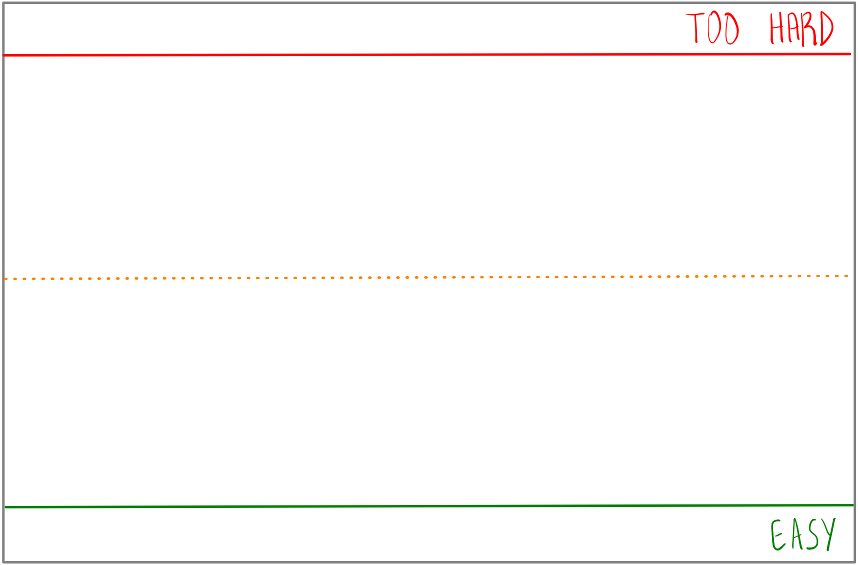
Fully-Homomorphic Encryption (FHE)

Learning with Errors (LWE)

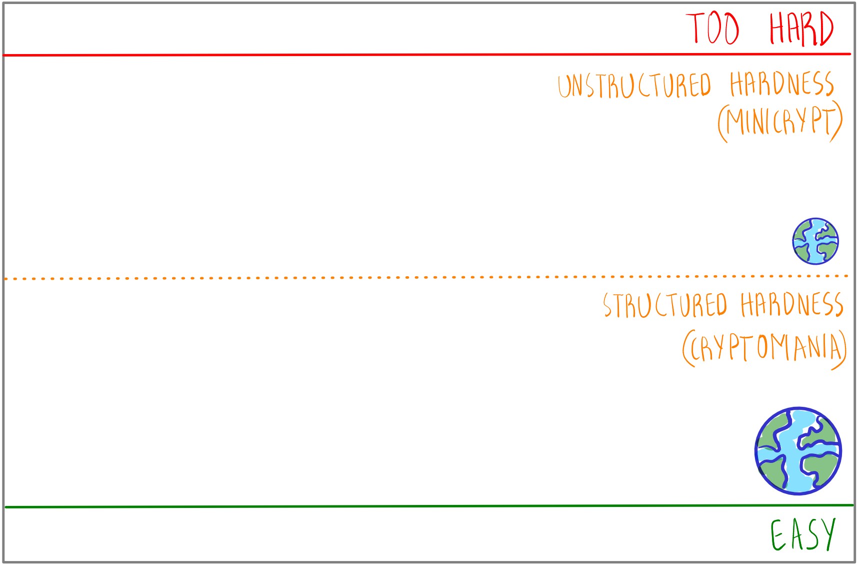
Gentry-Sahai-Waters FHE from LWE

Wrapping Up

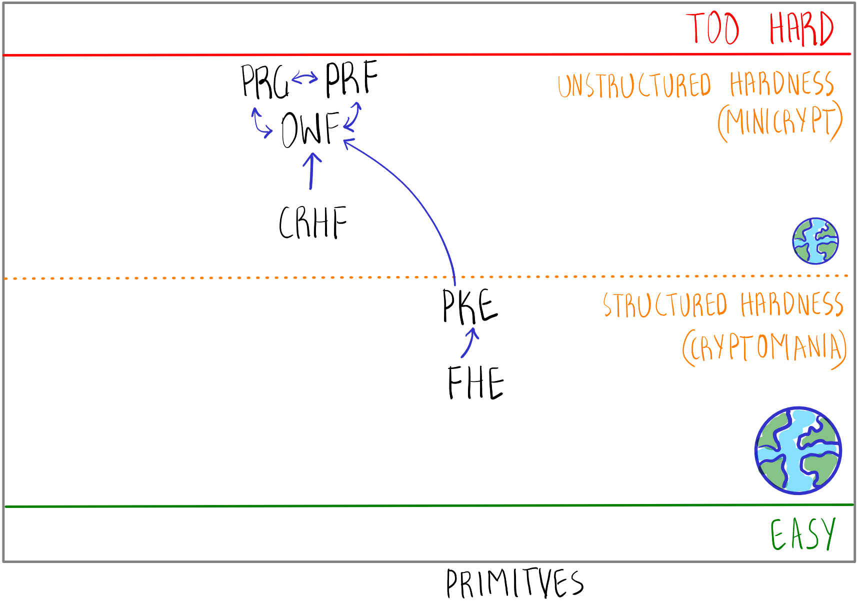
# Cryptography Landscape



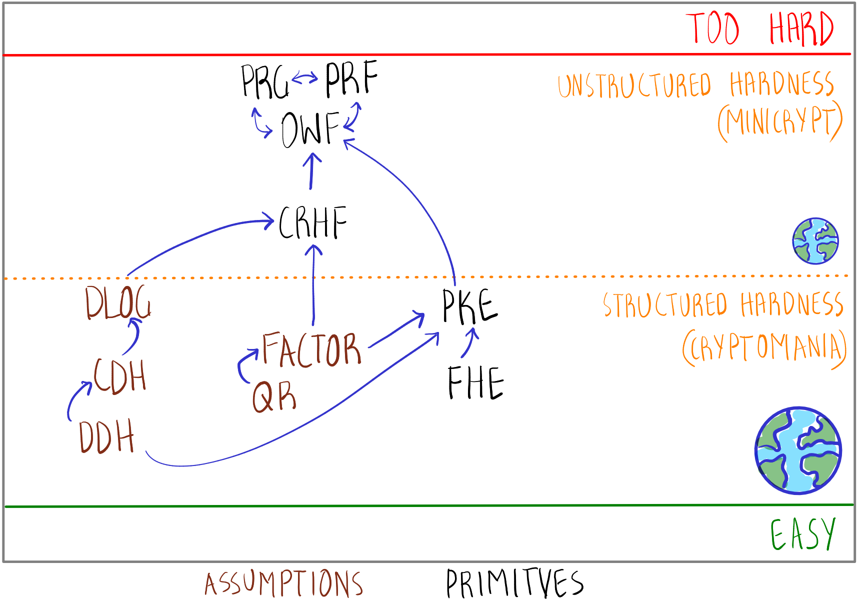
# Cryptography Landscape



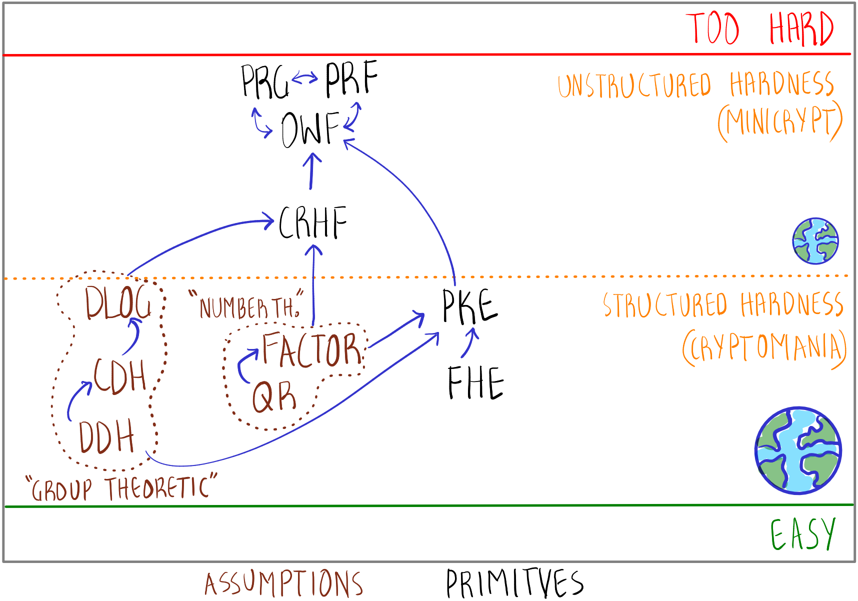
# Cryptography Landscape



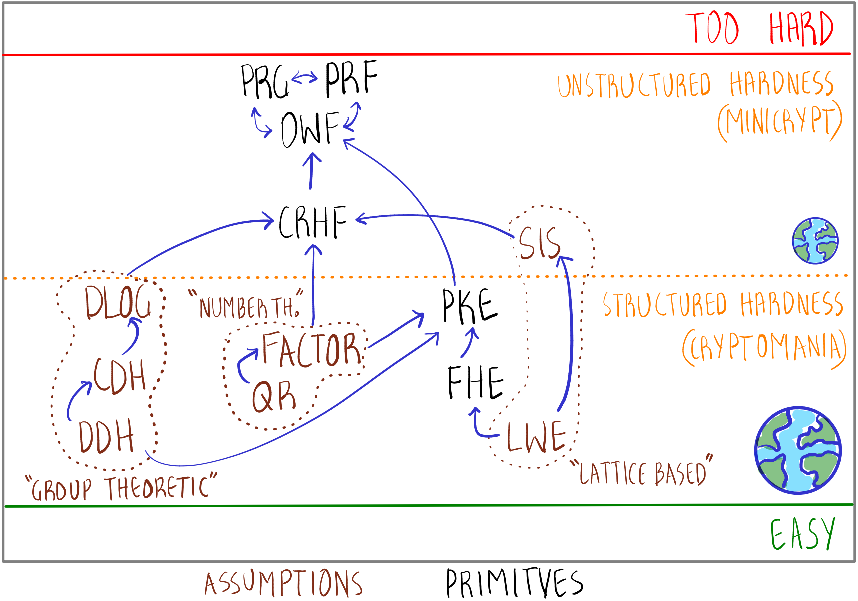
# Cryptography Landscape



# Cryptography Landscape



# Cryptography Landscape



# LWE: Solving "Noisy" Linear Equations is Hard

$$\begin{array}{c|ccccc|}
 & & & m & & \\
\hline
 & a_{11} & & \cdots & & a_{m1} \\
 & & \ddots & & & \\
n & \vdots & & a_{ij} & & \vdots \\
 & & & & \ddots & \\
 & a_{n1} & & \cdots & & a_{mn} \\
\hline
\end{array}$$

- Search vs decision LWE

- Solving LWE is at least as hard as solving certain lattice problems in the *worst case*

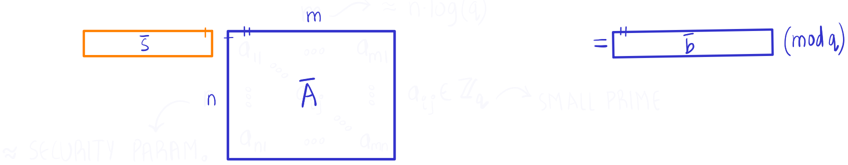
# LWE: Solving "Noisy" Linear Equations is Hard

$$\begin{array}{ccc} & m \rightarrow \approx n\cdot\log(q) & \\ \approx \text{SECURITY PARAM.} \leftarrow n & \boxed{\begin{matrix} a_{11} & \cdots & a_{m1} \\ \vdots & a_{ij} & \vdots \\ a_{n1} & \cdots & a_{mn}\end{matrix}} & a_{ij} \in \mathbb{Z}_q \rightarrow \text{SMALL PRIME} \end{array}$$

- Search vs decision LWE

- Solving LWE is at least as hard as solving certain lattice problems in the *worst case*

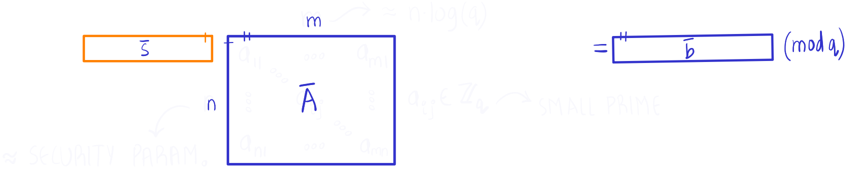
# LWE: Solving "Noisy" Linear Equations is Hard



- Search vs decision LWE

- Solving LWE is at least as hard as solving certain lattice problems in the *worst case*

# LWE: Solving "Noisy" Linear Equations is Hard

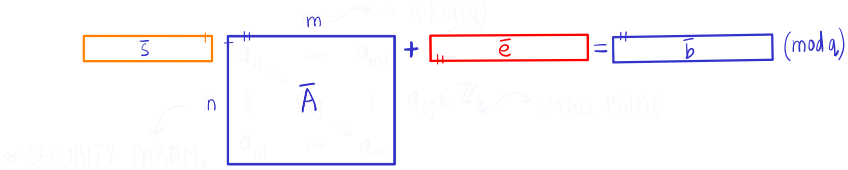


- Search vs decision LW

ELIMINATION

- Solving LWE is at least as hard as solving certain lattice problems in the *worst case*

# LWE: Solving “Noisy” Linear Equations is Hard

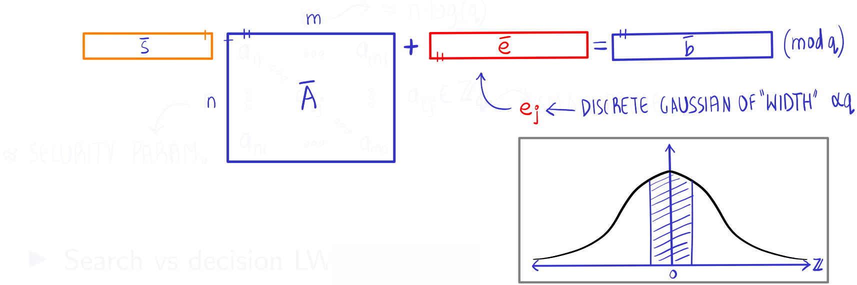


- Search vs decision LW

ELIMINATION

- Solving LWE is at least as hard as solving certain lattice problems in the *worst case*

# LWE: Solving "Noisy" Linear Equations is Hard

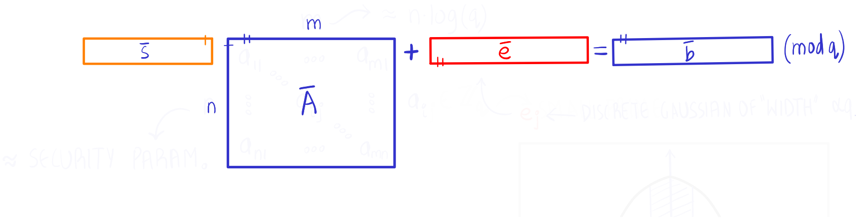


- Search vs decision LW

ELIMINATION

- Solving LWE is at least as hard as solving certain lattice problems in the *worst case*

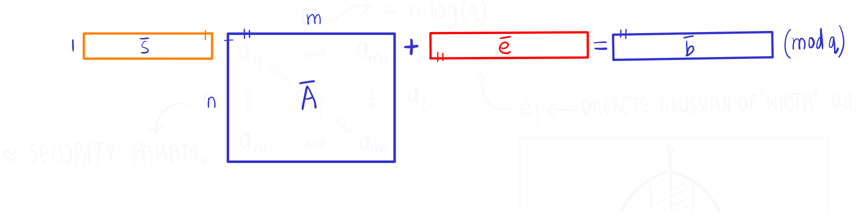
# LWE: Solving “Noisy” Linear Equations is Hard



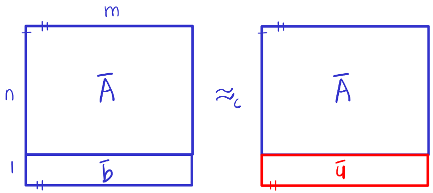
- Search vs decision LWE

- Solving LWE is at least as hard as solving certain lattice problems in the *worst case* [Regev05,Peikert09]

# LWE: Solving "Noisy" Linear Equations is Hard



- Search vs decision LWE

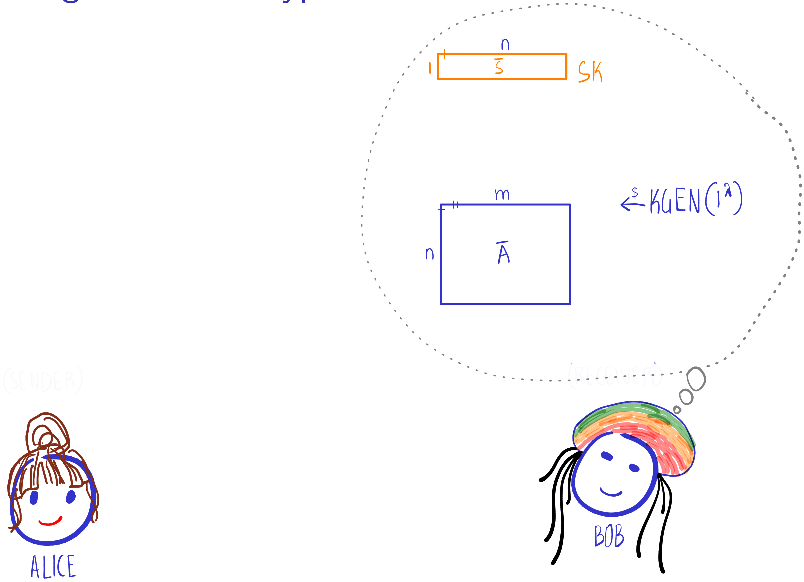


- Solving LWE is at least as hard as solving certain lattice problems in the *worst case* [Regev05,Peikert09]

# Regev's Bit Encryption: PKE from LWE...

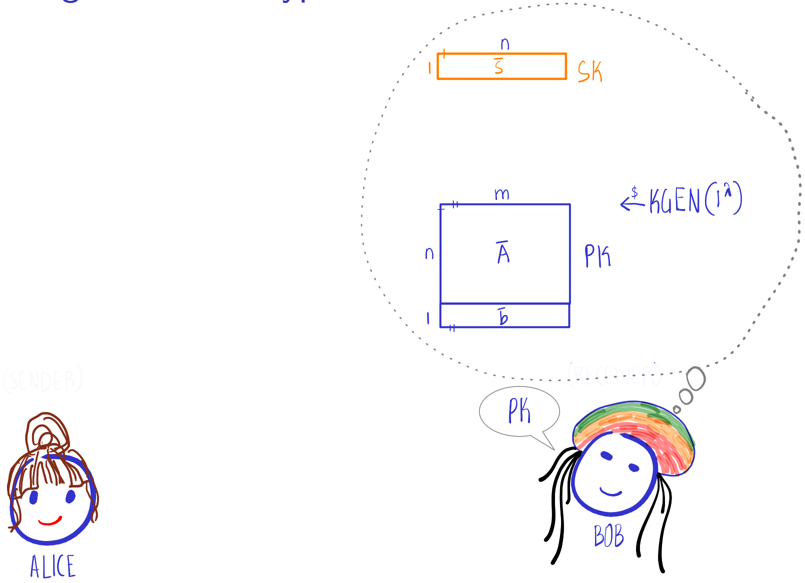
- What happens when you add two ciphertexts?

# Regev's Bit Encryption: PKE from LWE...



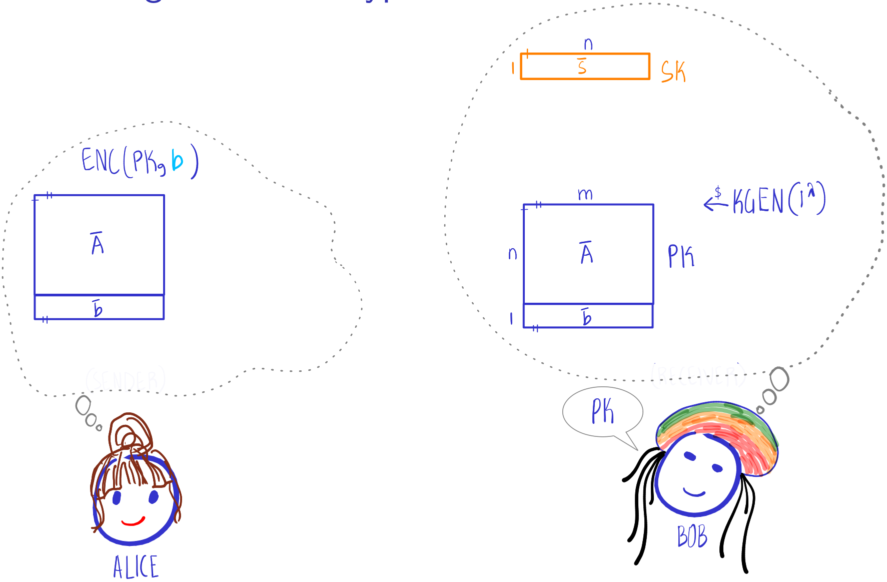
- What happens when you add two ciphertexts?

# Regev's Bit Encryption: PKE from LWE...



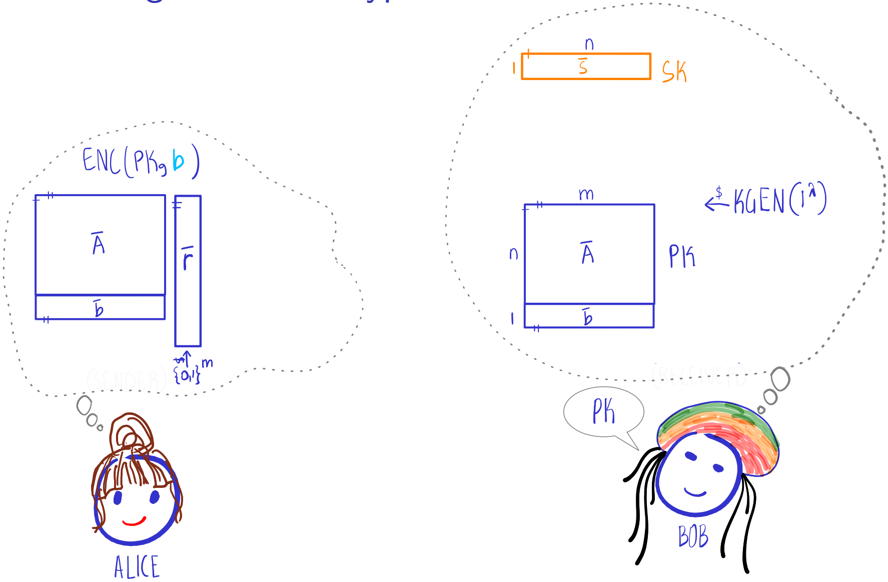
- What happens when you add two ciphertexts?

# Regev's Bit Encryption: PKE from LWE...



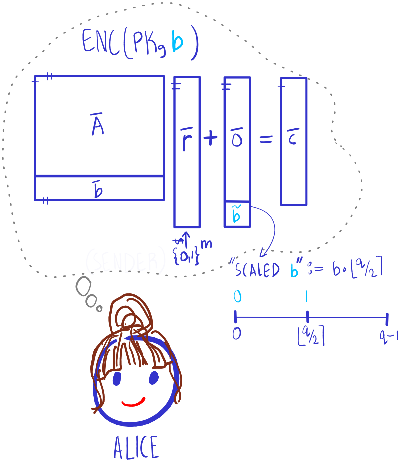
- What happens when you add two ciphertexts?

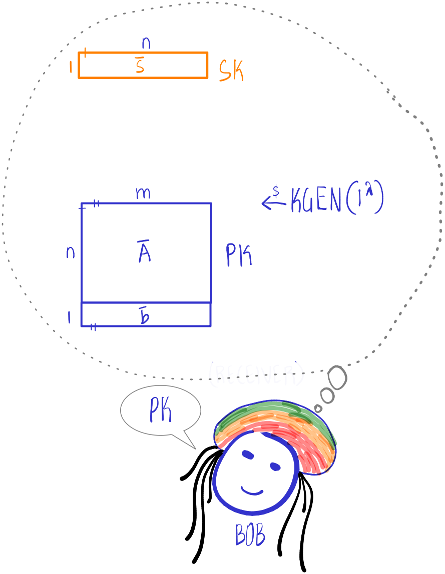
# Regev's Bit Encryption: PKE from LWE...



- What happens when you add two ciphertexts?

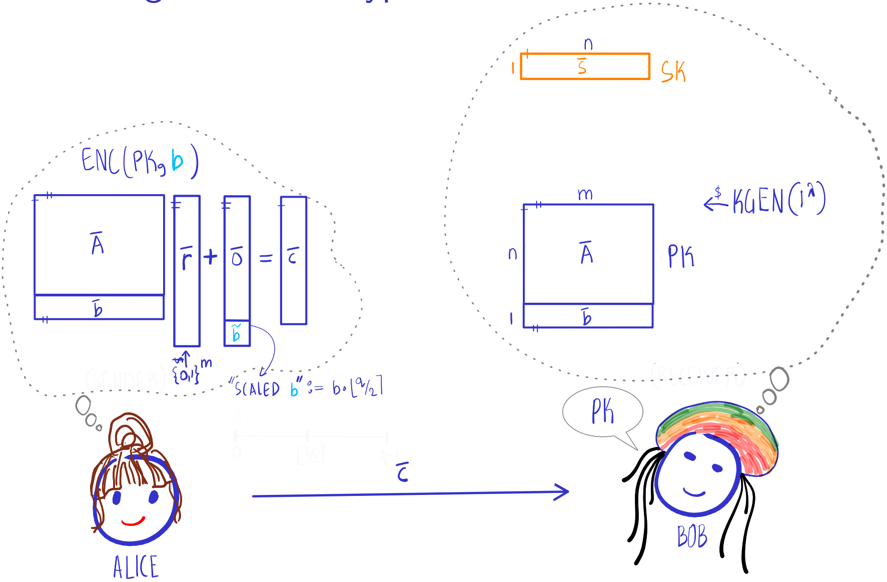
# Regev's Bit Encryption: PKE from LWE...





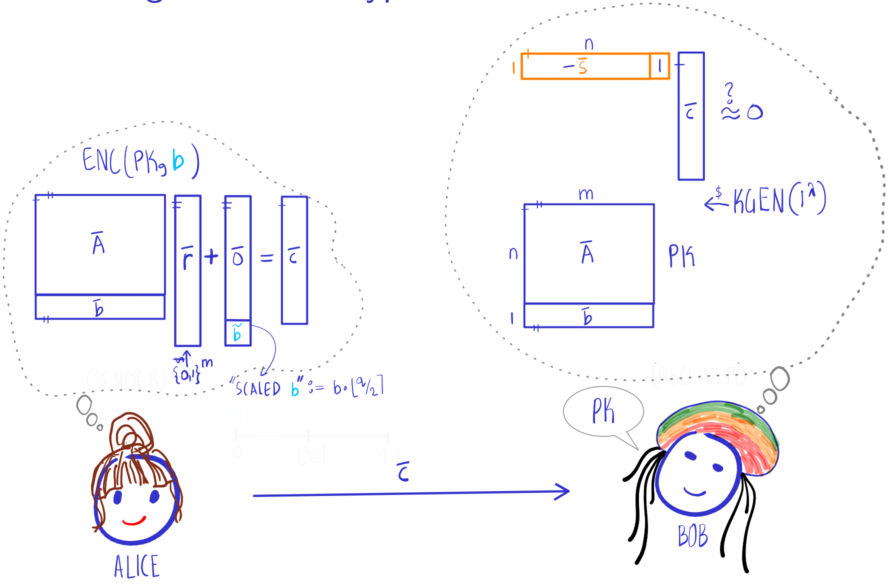
- What happens when you add two ciphertexts?

# Regev's Bit Encryption: PKE from LWE...



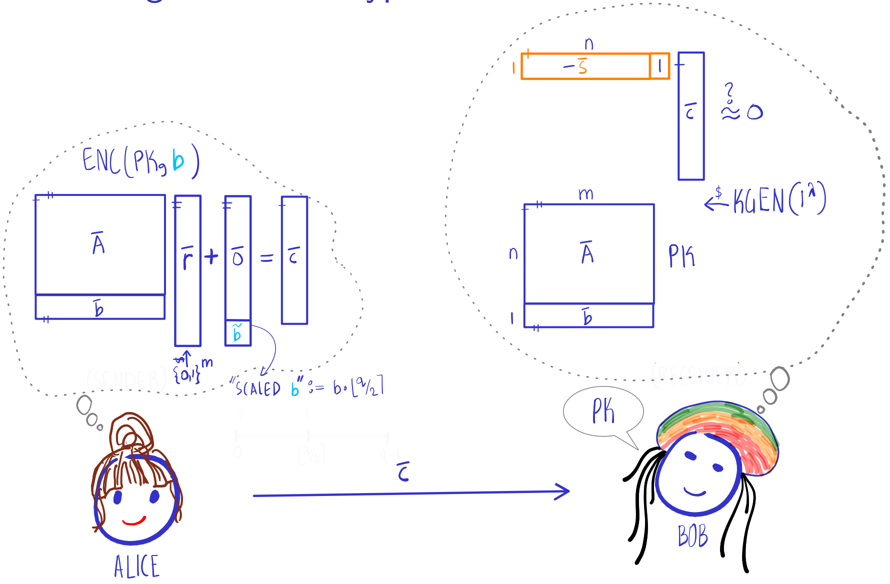
- What happens when you add two ciphertexts?

# Regev's Bit Encryption: PKE from LWE...


ENC(PK, b)
$\bar{A}$
$\bar{b}$
$\bar{r}$ + $\bar{0}$ = $\bar{c}$
$\tilde{b}$
$\{0,1\}^m$
"SCALED b" := $b \cdot \lfloor q/2 \rceil$
(SENDER)
ALICE
$\bar{c}$
$-\bar{s}$ | 1
$\bar{c}$ $\overset{?}{\approx} 0$
n
1
m
n
$\bar{A}$
1
$\bar{b}$
PK
$\xleftarrow{\$}$ KGEN($1^\lambda$)
(RECEIVER)
PK
BOB

- What happens when you add two ciphertexts?

# Regev's Bit Encryption: PKE from LWE...



- What happens when you add two ciphertexts?

# Regev's Bit Encryption: PKE from LWE...

- Correctness:

$$\begin{pmatrix} -\bar{s} & 1 \end{pmatrix} \left( \begin{pmatrix} \bar{A} \\ \bar{b} \end{pmatrix} \bar{r} + \begin{pmatrix} \bar{0} \\ \tilde{b} \end{pmatrix} \right) =$$

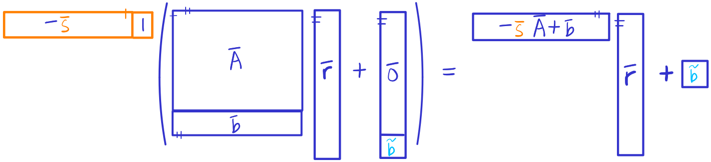
- Security by hybrid argument

**Exercise 3 (Security of Regev's Encryption)**

*Prove security formally.*

# Regev's Bit Encryption: PKE from LWE...

- Correctness:



- Security by hybrid argument

**Exercise 3 (Security of Regev's Encryption)**

*Prove security formally.*

# Regev's Bit Encryption: PKE from LWE...

- Correctness:

$$\begin{pmatrix} -\bar{s} & 1 \end{pmatrix} \left( \begin{pmatrix} \bar{A} \\ \bar{b} \end{pmatrix} \bar{r} + \begin{pmatrix} \bar{0} \\ \tilde{b} \end{pmatrix} \right) = \bar{e}\,\bar{r} + \boxed{\tilde{b}}$$

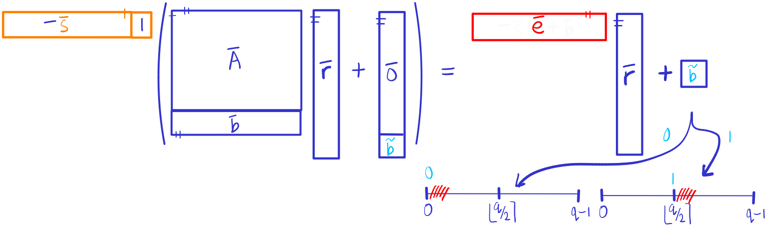
- Security by hybrid argument

**Exercise 3 (Security of Regev's Encryption)**

*Prove security formally.*

# Regev's Bit Encryption: PKE from LWE...

- Correctness:



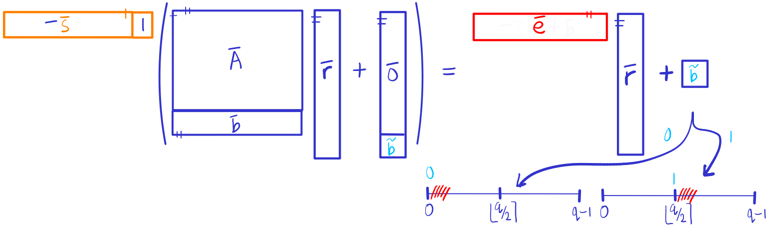
- Security by hybrid argument

**Exercise 3 (Security of Regev's Encryption)**

*Prove security formally.*

# Regev's Bit Encryption: PKE from LWE...

- Correctness:



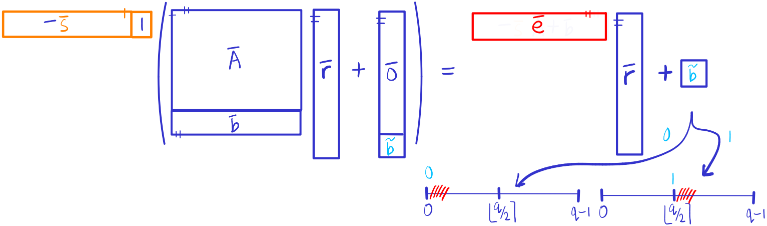
- Security by hybrid argument

**Exercise 3 (Security of Regev's Encryption)**

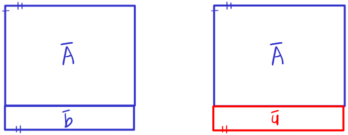
*Prove security formally.*

# Regev's Bit Encryption: PKE from LWE...

- Correctness:



- Security by hybrid argument

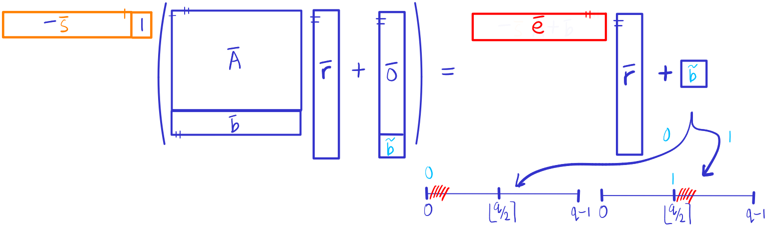


**Exercise 3 (Security of Regev's Encryption)**

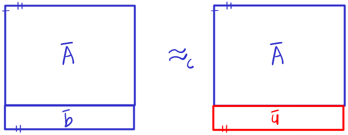
*Prove security formally.*

# Regev's Bit Encryption: PKE from LWE...

- Correctness:



- Security by hybrid argument

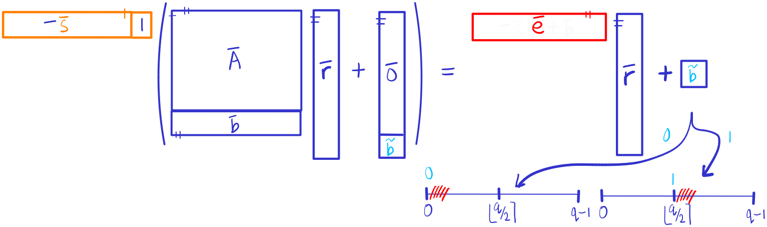


**Exercise 3 (Security of Regev's Encryption)**

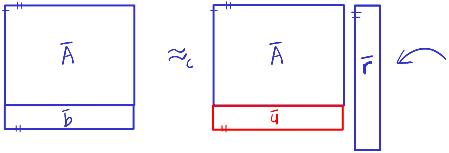
*Prove security formally.*

# Regev's Bit Encryption: PKE from LWE...

- Correctness:



- Security by hybrid argument



STATISTICALLY HIDES b

"LEFTOVER HASH LEMMA"

### Exercise 3 (Security of Regev's Encryption)

*Prove security formally.*

# Plan for this Session

Homomorphic Encryption

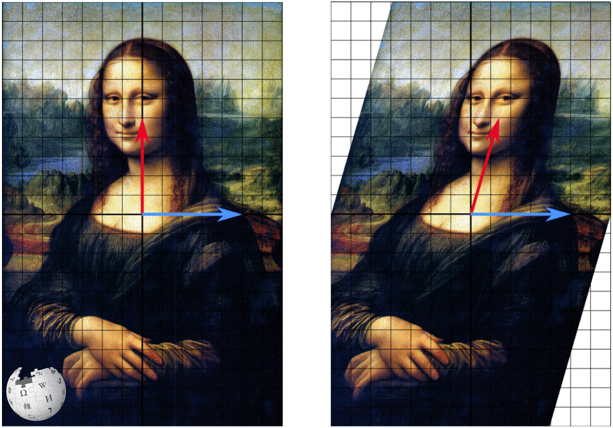
Fully-Homomorphic Encryption (FHE)

Learning with Errors (LWE)

Gentry-Sahai-Waters FHE from LWE

Wrapping Up

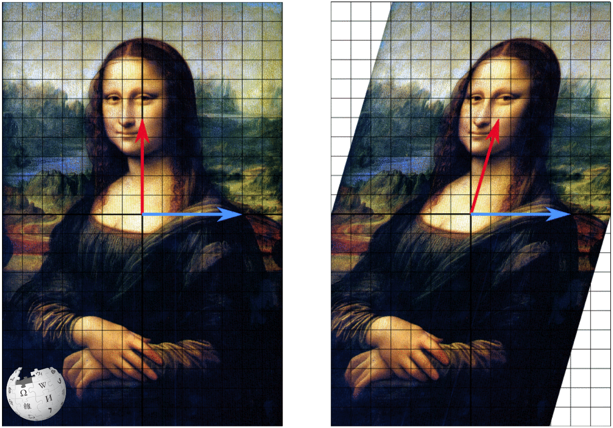
# Let's Recall Eigenvectors



**Definition 1**

A (left) eigenvector of a square matrix $\bar{C}$ is a vector $\bar{v}$ such that $\bar{v}\bar{C} = \mu\bar{v}$ for some scalar $\mu$, which is the eigenvalue.

# Let's Recall Eigenvectors

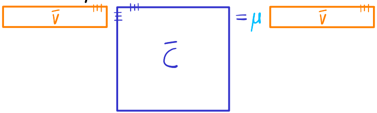


**Definition 1**

A (left) eigenvector of a square matrix $\bar{C}$ is a vector $\bar{v}$ such that $\bar{v}\bar{C} = \mu\bar{v}$ for some scalar $\mu$, which is the eigenvalue.

# Toy Example: “Eigenvector” Encryption

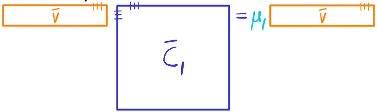
- An $N \times N$ matrix $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} = \mu\bar{v}$

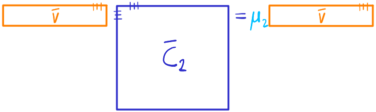


- Do we have an FHE?

# Toy Example: "Eigenvector" Encryption

- An $N \times N$ matrix $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} = \mu\bar{v}$

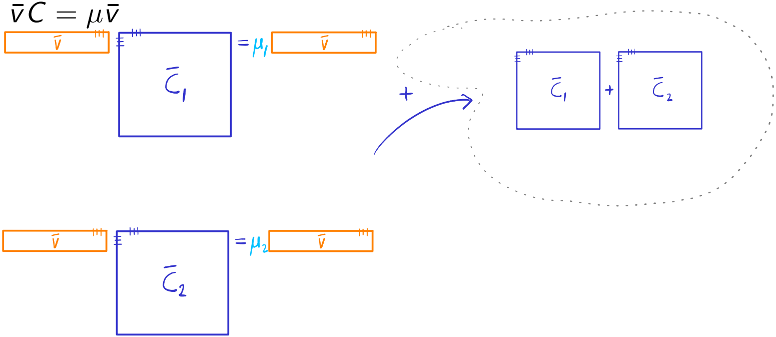




- Do we have an FHE?

# Toy Example: “Eigenvector” Encryption

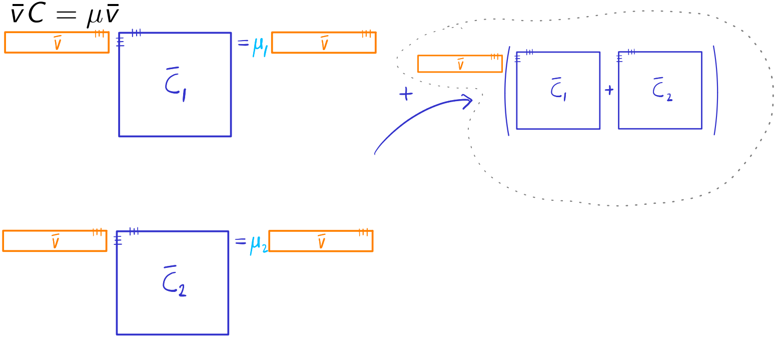
- An $N \times N$ matrix $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} = \mu\bar{v}$



- Do we have an FHE?

# Toy Example: “Eigenvector” Encryption

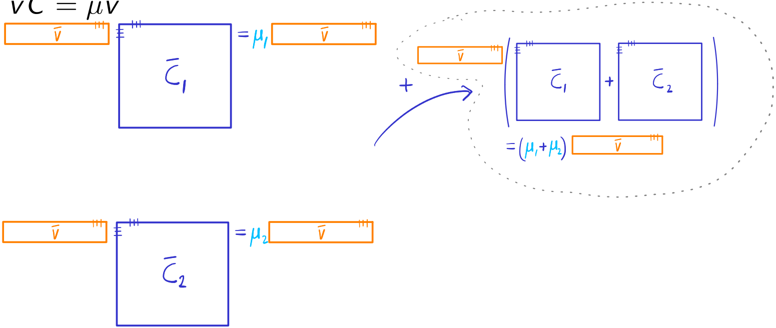
- An $N \times N$ matrix $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} = \mu\bar{v}$



- Do we have an FHE?

# Toy Example: “Eigenvector” Encryption

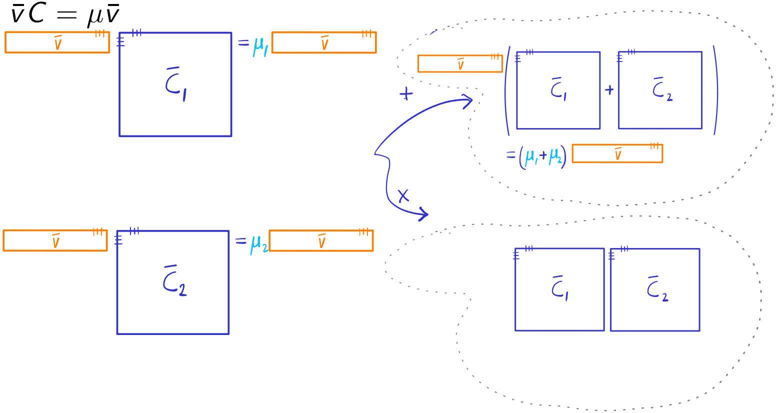
- An $N \times N$ matrix $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} = \mu\bar{v}$



- Do we have an FHE?

# Toy Example: “Eigenvector” Encryption

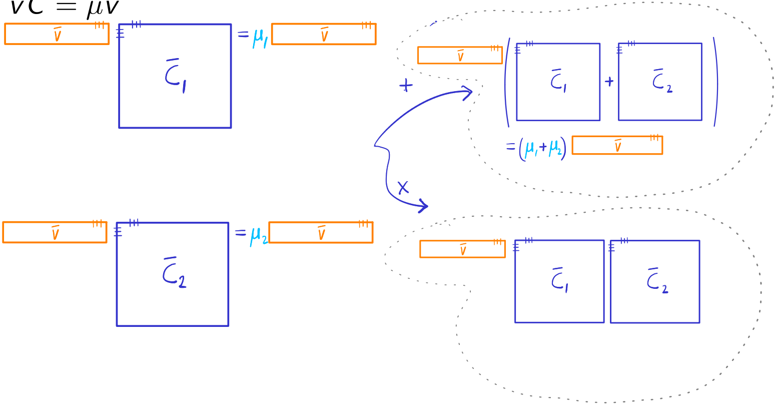
- An $N \times N$ matrix $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} = \mu\bar{v}$



- Do we have an FHE?

# Toy Example: “Eigenvector” Encryption

- An $N \times N$ matrix $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} = \mu\bar{v}$

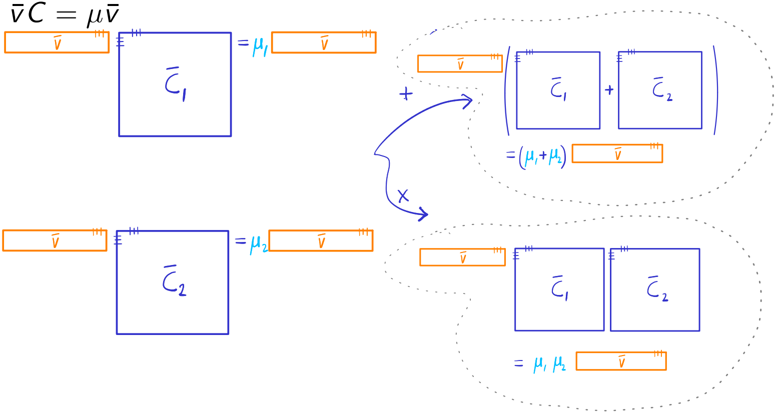


$\bar{v}$ $\bar{C}_1$ $= \mu_1$ $\bar{v}$
$\bar{v}$ $\bar{C}_2$ $= \mu_2$ $\bar{v}$
+
×
$\bar{v}$ $(\bar{C}_1 + \bar{C}_2)$
$= (\mu_1 + \mu_2)$ $\bar{v}$
$\bar{v}$ $\bar{C}_1$ $\bar{C}_2$

- Do we have an FHE?

# Toy Example: "Eigenvector" Encryption

- An $N \times N$ matrix $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} = \mu\bar{v}$



$\bar{v}$ $\bar{C}_1$ $= \mu_1$ $\bar{v}$
$\bar{v}$ $\bar{C}_2$ $= \mu_2$ $\bar{v}$
+
×
$\bar{v}$ $(\bar{C}_1 + \bar{C}_2)$
$= (\mu_1 + \mu_2)$ $\bar{v}$
$\bar{v}$ $\bar{C}_1$ $\bar{C}_2$
$= \mu_1 \mu_2$ $\bar{v}$

- Do we have an FHE?

# Toy Example: “Eigenvector” Encryption

- An $N \times N$ matrix $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} = \mu\bar{v}$

$$\bar{v}\,\bar{C}_1 = \mu_1\,\bar{v}$$

$$\bar{v}\,\bar{C}_2 = \mu_2\,\bar{v}$$

$+$

$$\bar{v}\left(\bar{C}_1 + \bar{C}_2\right) = (\mu_1 + \mu_2)\,\bar{v}$$

$\times$

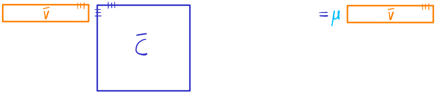
$$\bar{v}\,\bar{C}_1\,\bar{C}_2 = \mu_1\,\mu_2\,\bar{v}$$

- Do we have an FHE?

ELIMINATION

# How to Fix? *Approximate* Eigenvector Encryption

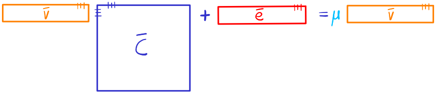
- $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" $\bar{e}$



- Do we have an FHE?
  - For "$B$-bounded" $\bar{C}$, $\bar{e}$ and $\mu$, error grows exp. in levels
  - Somewhat homomorphic: levelled FHE supporting log-depth $F$

# How to Fix? *Approximate* Eigenvector Encryption

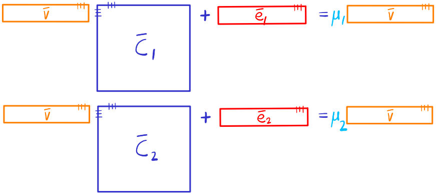
- $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" $\bar{e}$



- Do we have an FHE?
  - For "$B$-bounded" $\bar{C}$, $\bar{e}$ and $\mu$, error grows exp. in levels
  - Somewhat homomorphic: levelled FHE supporting log-depth $F$

# How to Fix? *Approximate* Eigenvector Encryption

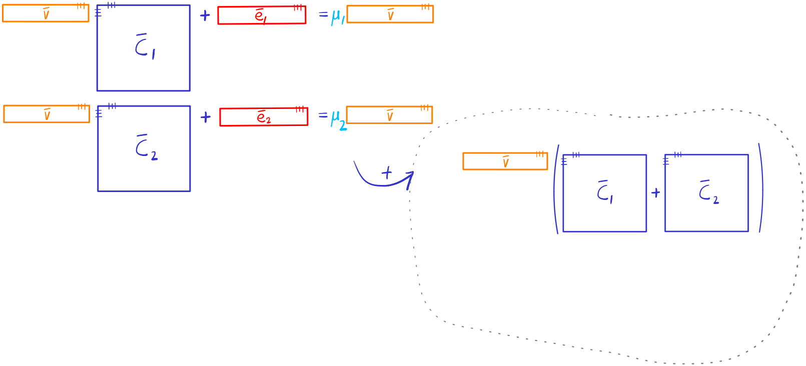
- $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" $\bar{e}$



- Do we have an FHE?
  - For "$B$-bounded" $\bar{C}$, $\bar{e}$ and $\mu$, error grows exp. in levels
  - Somewhat homomorphic: levelled FHE supporting log-depth $F$

# How to Fix? *Approximate* Eigenvector Encryption

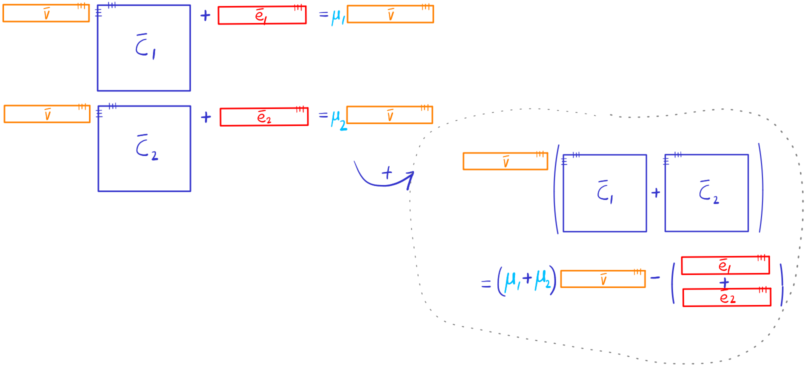
- $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" $\bar{e}$



- Do we have an FHE?
  - For "$B$-bounded" $\bar{C}$, $\bar{e}$ and $\mu$, error grows exp. in levels
  - Somewhat homomorphic: levelled FHE supporting log-depth $F$

# How to Fix? *Approximate* Eigenvector Encryption

- $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" $\bar{e}$

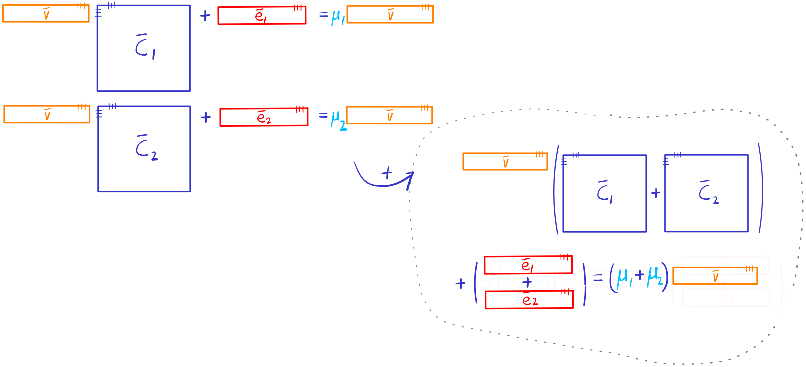


$\bar{v}$ $\bar{C}_1$ + $\bar{e}_1$ $= \mu_1$ $\bar{v}$
$\bar{v}$ $\bar{C}_2$ + $\bar{e}_2$ $= \mu_2$ $\bar{v}$
+
$\bar{v}$ $(\bar{C}_1 + \bar{C}_2)$
$= (\mu_1 + \mu_2)$ $\bar{v}$ $-$ $(\bar{e}_1 + \bar{e}_2)$

- Do we have an FHE?
  - For "$B$-bounded" $\bar{C}$, $\bar{e}$ and $\mu$, error grows exp. in levels
  - Somewhat homomorphic: levelled FHE supporting log-depth $F$

# How to Fix? *Approximate* Eigenvector Encryption

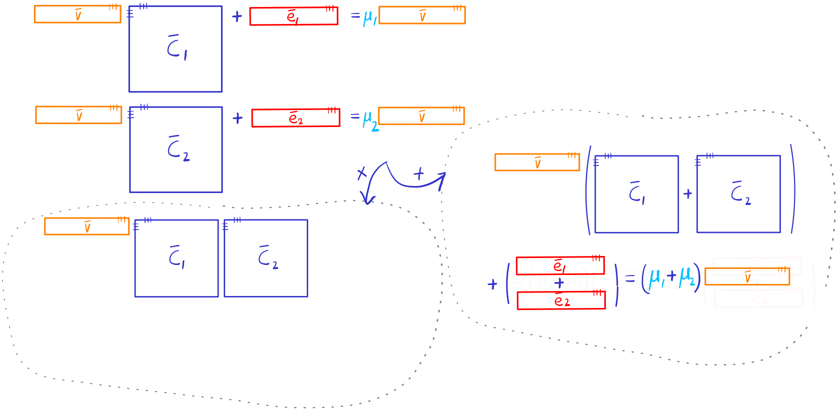
- $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" $\bar{e}$



- Do we have an FHE?
  - For "$B$-bounded" $\bar{C}$, $\bar{e}$ and $\mu$, error grows exp. in levels
  - Somewhat homomorphic: levelled FHE supporting log-depth *F*

# How to Fix? *Approximate* Eigenvector Encryption

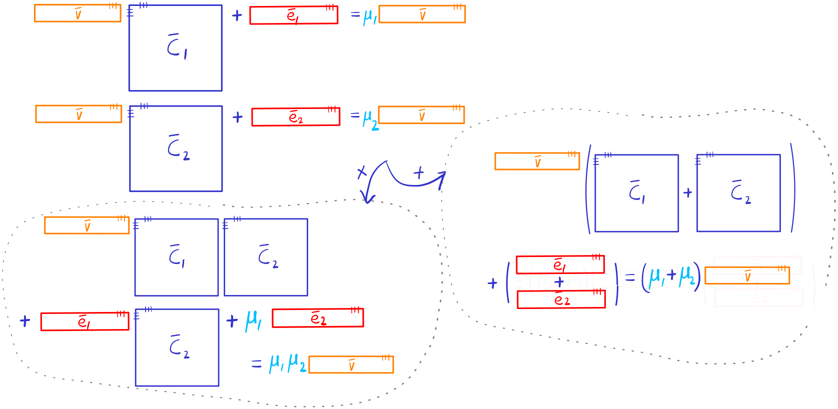
- $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" $\bar{e}$



- Do we have an FHE?
  - For "$B$-bounded" $\bar{C}$, $\bar{e}$ and $\mu$, error grows exp. in levels
  - Somewhat homomorphic: levelled FHE supporting log-depth *F*

# How to Fix? *Approximate* Eigenvector Encryption

- $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" $\bar{e}$

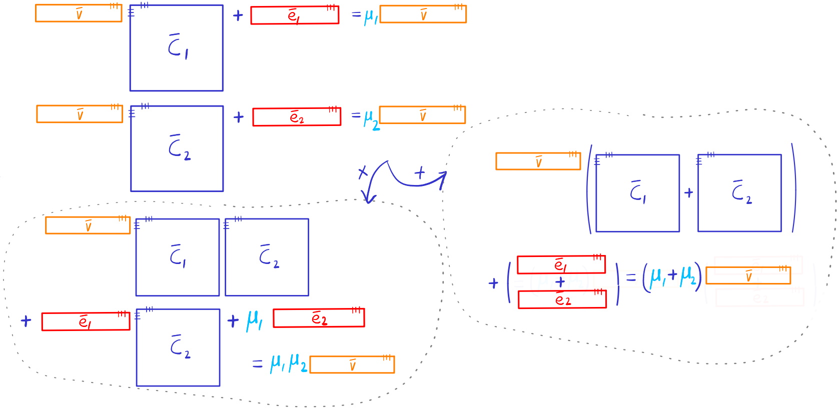


$\bar{v}\,\bar{C}_1 + \bar{e}_1 = \mu_1\bar{v}$
$\bar{v}\,\bar{C}_2 + \bar{e}_2 = \mu_2\bar{v}$
×
+
$\bar{v}\,\bar{C}_1\,\bar{C}_2 + \bar{e}_1\,\bar{C}_2 + \mu_1\,\bar{e}_2 = \mu_1\mu_2\,\bar{v}$
$\bar{v}\left(\bar{C}_1 + \bar{C}_2\right) + \left(\bar{e}_1 + \bar{e}_2\right) = (\mu_1+\mu_2)\,\bar{v}$

- Do we have an FHE?
  - For "$B$-bounded" $\bar{C}$, $\bar{e}$ and $\mu$, error grows exp. in levels
  - Somewhat homomorphic: levelled FHE supporting log-depth *F*

# How to Fix? *Approximate* Eigenvector Encryption

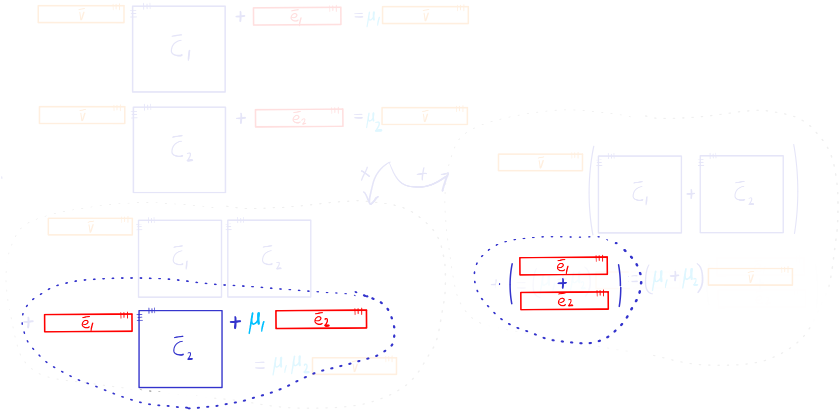
- $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" $\bar{e}$


$\bar{v}$ $\bar{C}_1$ + $\bar{e}_1$ = $\mu_1$ $\bar{v}$
$\bar{v}$ $\bar{C}_2$ + $\bar{e}_2$ = $\mu_2$ $\bar{v}$
×
+
$\bar{v}$ $\bar{C}_1$ $\bar{C}_2$
+ $\bar{e}_1$ $\bar{C}_2$ + $\mu_1$ $\bar{e}_2$
= $\mu_1\mu_2$ $\bar{v}$
$\bar{v}$ ( $\bar{C}_1$ + $\bar{C}_2$ )
+ ( $\bar{e}_1$ + $\bar{e}_2$ ) = $(\mu_1+\mu_2)$ $\bar{v}$

- Do we have an FHE?
  - For "$B$-bounded" $\bar{C}$, $\bar{e}$ and $\mu$, error grows exp. in levels
  - Somewhat homomorphic: levelled FHE supporting log-depth *F*

# How to Fix? *Approximate* Eigenvector Encryption

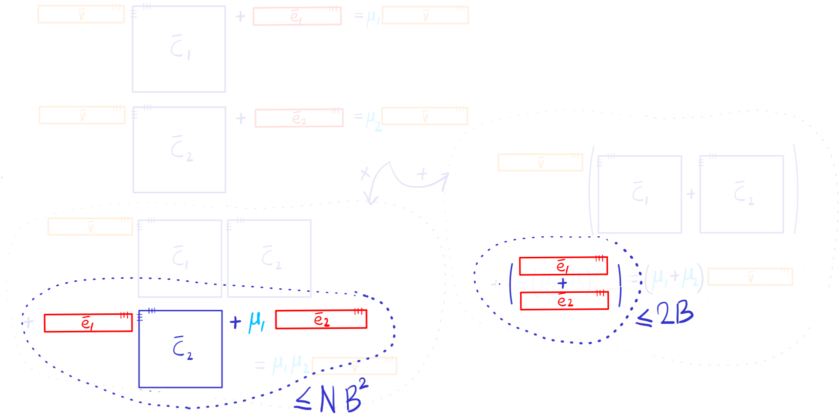
- $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" $\bar{e}$



- Do we have an FHE?
  - For "$B$-bounded" $\bar{C}$, $\bar{e}$ and $\mu$, error grows exp. in levels
  - Somewhat homomorphic: levelled FHE supporting log-depth $F$

# How to Fix? *Approximate* Eigenvector Encryption

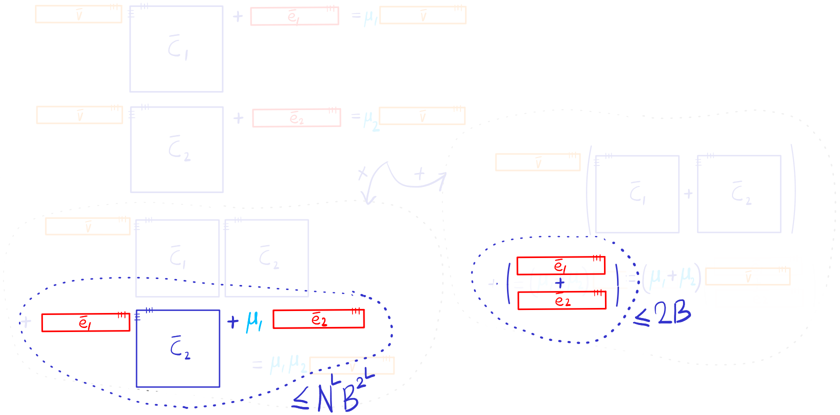
- $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" $\bar{e}$



- Do we have an FHE?
  - For "$B$-bounded" $\bar{C}$, $\bar{e}$ and $\mu$, error grows exp. in levels
  - Somewhat homomorphic: levelled FHE supporting log-depth *F*

# How to Fix? *Approximate* Eigenvector Encryption

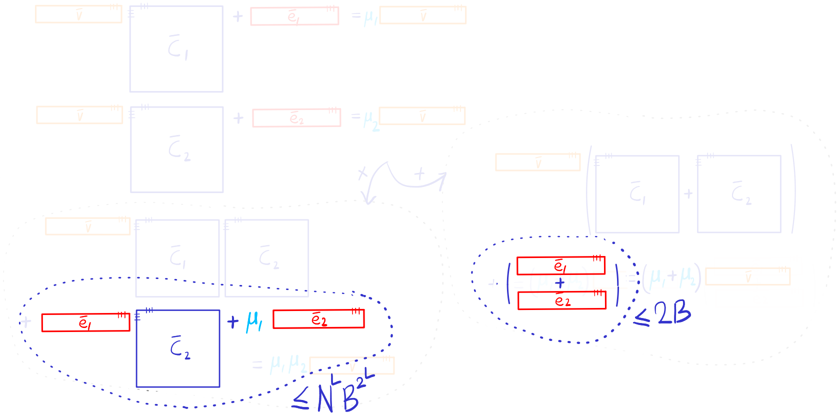
- $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" $\bar{e}$



- Do we have an FHE?
  - For "$B$-bounded" $\bar{C}$, $\bar{e}$ and $\mu$, error grows exp. in levels
  - Somewhat homomorphic: levelled FHE supporting log-depth *F*

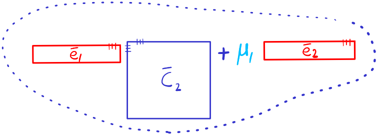
# How to Fix? *Approximate* Eigenvector Encryption

- $\bar{C}$ encrypts a bit $\mu$ under secret $\bar{v}$ if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" $\bar{e}$



- Do we have an FHE?
  - For "$B$-bounded" $\bar{C}$, $\bar{e}$ and $\mu$, error grows exp. in levels
  - Somewhat homomorphic: levelled FHE supporting log-depth $F$

# Supporting Arbitrary Depth

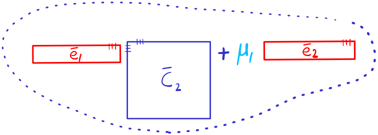


- Two tricks:
  1. Stick to messages $\mu$ from $\{0, 1\}$ and $F$ with NAND gates
  2. "Flattening": embed matrix $\bar{C}$ into a higher dimensional matrix $\bar{C}'$ such that
     - 2.1 $\bar{C}'$ has low (infinity) norm
     - 2.2 Certain inner products "preserved"

     Implemented using "gadget" matrix $\bar{G} : \mathbb{Z}_q^{n\times N} \to \mathbb{Z}_q^{n\times m}$

     bit-decomposition function $G^{-1} : \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times N}$

# Supporting Arbitrary Depth



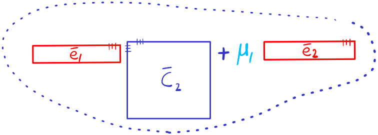
- Two tricks:
  1. Stick to messages $\mu$ from $\{0, 1\}$ and $F$ with NAND gates
  2. "Flattening": embed matrix $\bar{C}$ into a higher dimensional matrix $\bar{C}'$ such that
     - 2.1 $\bar{C}'$ has low (infinity) norm
     - 2.2 Certain inner products "preserved"

     $m\lceil\log q\rceil$

     Implemented using "gadget" matrix $\bar{G} : \mathbb{Z}_q^{n\times N} \to \mathbb{Z}_q^{n\times m}$
     bit-decomposition function $G^{-1} : \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times N}$

$$\sum_{k\in[\ell]} a_{11k}\, 2^k = a_{11}$$

# Supporting Arbitrary Depth



- Two tricks:
  1. Stick to messages $\mu$ from $\{0, 1\}$ and $F$ with NAND gates
  2. "Flattening": embed matrix $\bar{C}$ into a higher dimensional matrix $\bar{C}'$ such that
     - 2.1 $\bar{C}'$ has low (infinity) norm
     - 2.2 Certain inner products "preserved"

     $m\lceil \log q\rceil$

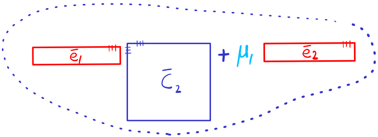
     Implemented using "gadget" matrix $\bar{G} : \mathbb{Z}_q^{n\times N} \to \mathbb{Z}_q^{n\times m}$

     bit-decomposition function $G^{-1} : \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times N}$

$\sum_{k\in[\ell]} a_{11k} 2^k = a_{11}$

$\boxed{a_{11} \cdots a_{1\ell}}$

# Supporting Arbitrary Depth



- Two tricks:
  1. Stick to messages $\mu$ from $\{0, 1\}$ and $F$ with NAND gates
  2. "Flattening": embed matrix $\bar{C}$ into a higher dimensional matrix $\bar{C}'$ such that
     - 2.1 $\bar{C}'$ has low (infinity) norm
     - 2.2 Certain inner products "preserved"

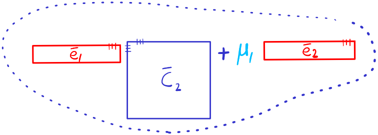
     $m\lceil \log q \rceil$

     Implemented using "gadget" matrix $\bar{G} : \mathbb{Z}_q^{n\times N} \to \mathbb{Z}_q^{n\times m}$

     bit-decomposition function $G^{-1} : \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times N}$

$$\sum_{k\in[\ell]} a_{11k} 2^k = a_{11} \quad \circ\circ\circ \quad a_{1m} \qquad \boxed{a_{11} \circ\circ\circ\, a_{1\ell}} \quad \circ\circ\circ \quad \boxed{a_{n11} \circ\circ\circ\, a_{1m\ell}}$$

# Supporting Arbitrary Depth

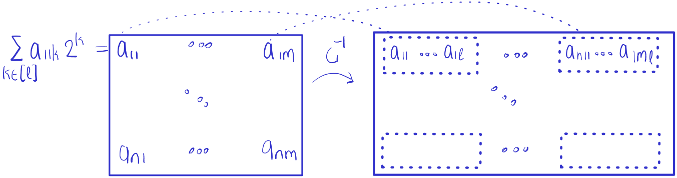


- Two tricks:
  1. Stick to messages $\mu$ from $\{0, 1\}$ and $F$ with NAND gates
  2. "Flattening": embed matrix $\bar{C}$ into a higher dimensional matrix $\bar{C}'$ such that
     - 2.1 $\bar{C}'$ has low (infinity) norm
     - 2.2 Certain inner products "preserved"

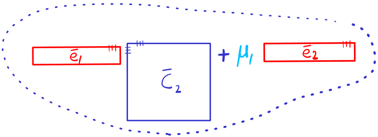
  $m\lceil \log q \rceil$

  Implemented using "gadget" matrix $\bar{G} : \mathbb{Z}_q^{n\times N} \to \mathbb{Z}_q^{n\times m}$

  bit-decomposition function $G^{-1} : \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times N}$



# Supporting Arbitrary Depth

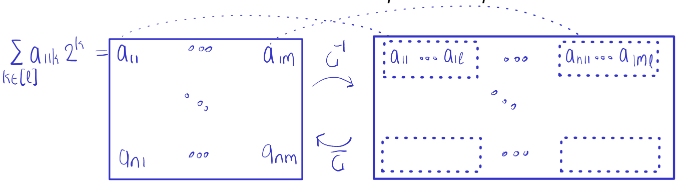


- Two tricks:
  1. Stick to messages $\mu$ from $\{0,1\}$ and $F$ with NAND gates
  2. "Flattening": embed matrix $\bar{C}$ into a higher dimensional matrix $\bar{C}'$ such that
     - 2.1 $\bar{C}'$ has low (infinity) norm
     - 2.2 Certain inner products "preserved"

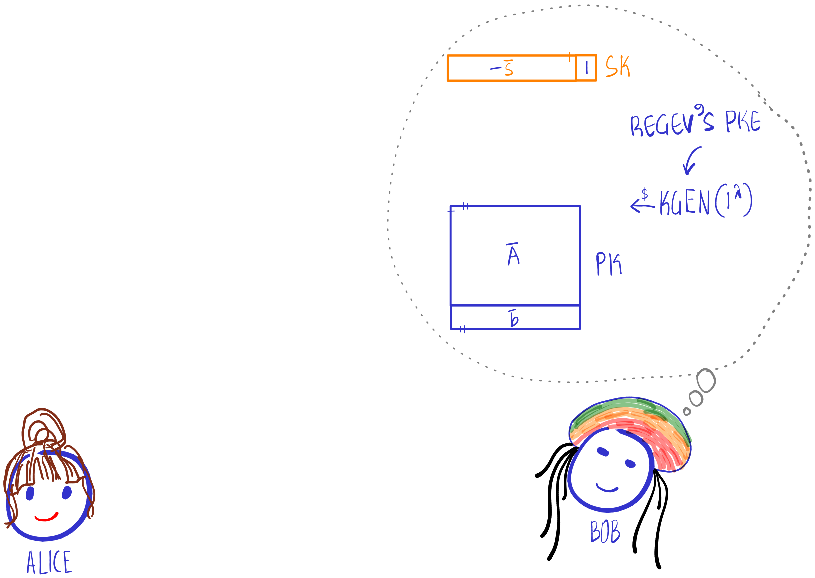
     $m\lceil \log q\rceil$

     Implemented using "gadget" matrix $\bar{G} : \mathbb{Z}_q^{n\times N} \to \mathbb{Z}_q^{n\times m}$

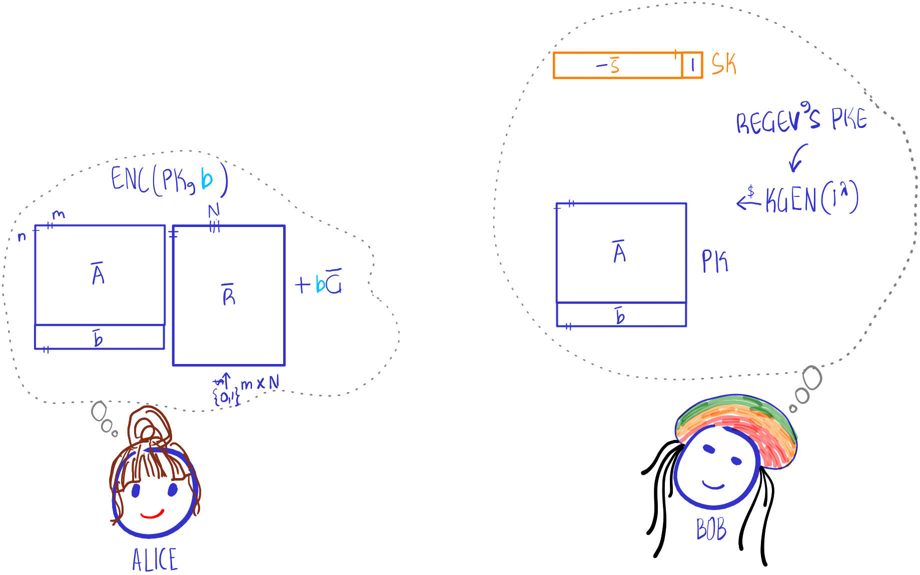
     bit-decomposition function $G^{-1} : \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times N}$



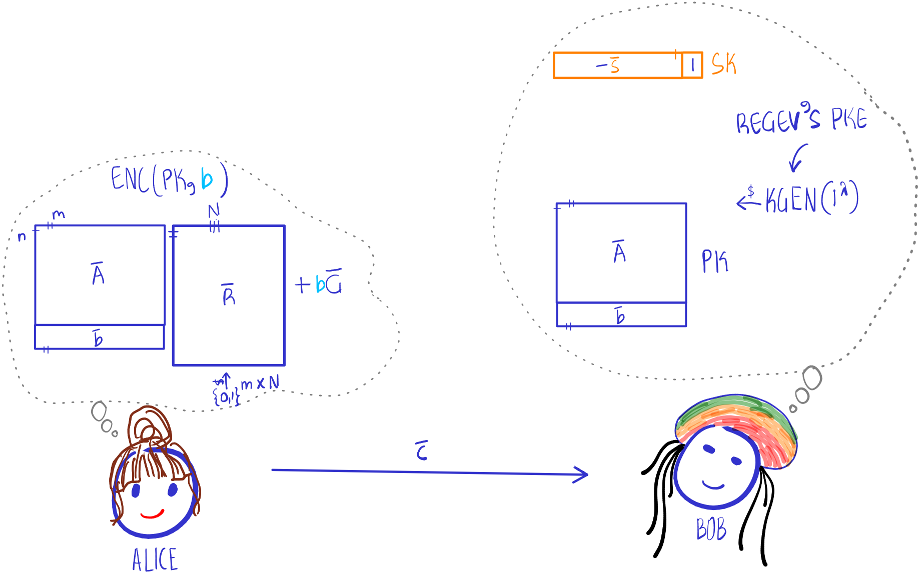
# Putting it all Together



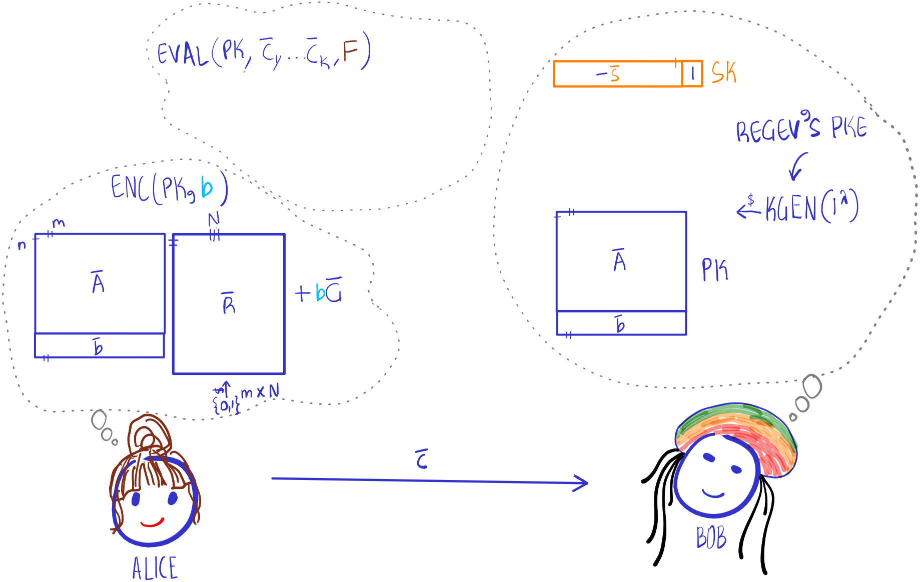
# Putting it all Together



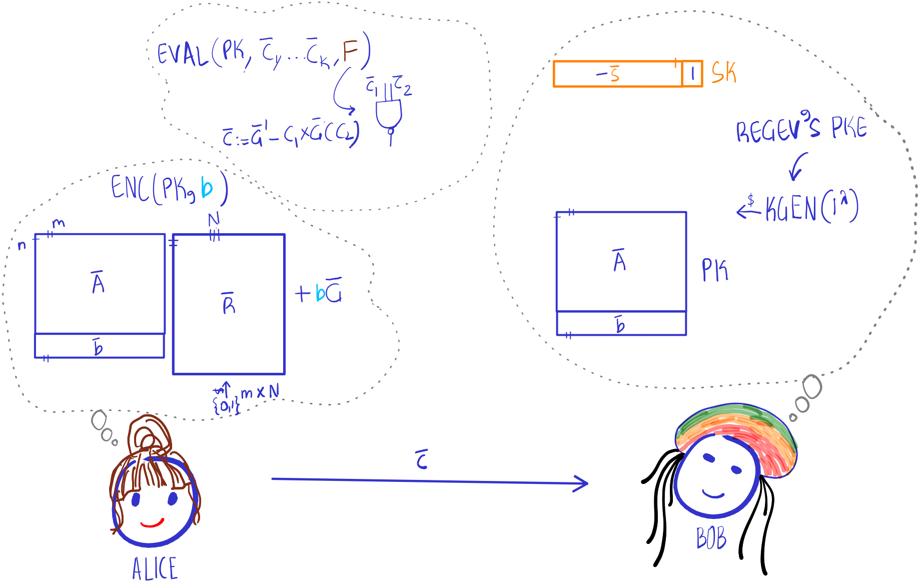
# Putting it all Together



# Putting it all Together



# Putting it all Together



# Plan for this Session

Homomorphic Encryption

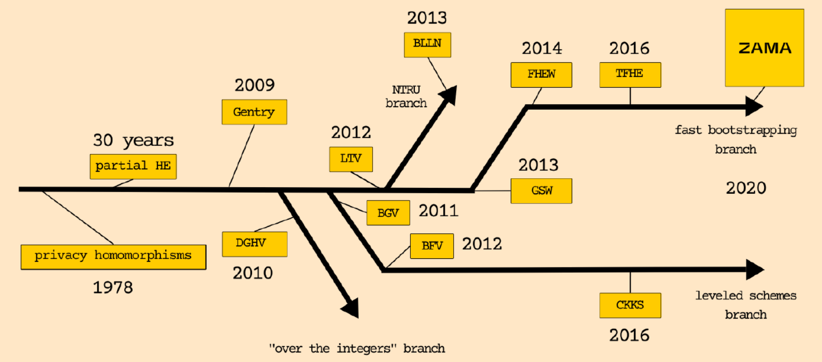
Fully-Homomorphic Encryption (FHE)

Learning with Errors (LWE)

Gentry-Sahai-Waters FHE from LWE

Wrapping Up

# Genealogy of FHE Schemes



COURTESY: ZAMA.AI

# To Recap

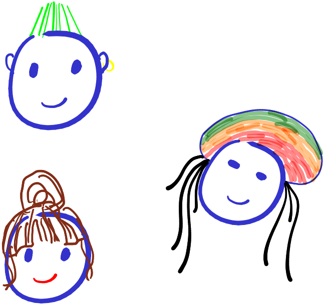
- Saw partially homomorphic encryption schemes
- Learned about LWE and Regev's PKE based on LWE
- GSW FHE via approximate eigenvectors

# To Recap

- Saw partially homomorphic encryption schemes
- Learned about LWE and Regev's PKE based on LWE
- GSW FHE via approximate eigenvectors

- Archisman's session for how to use FHE

# Thank You for Your Attention! Questions?



# References

1. The partially homomorphic schemes we discussed are from [ElG84, GM82]
2. The LWE problem was introduced in [Reg05], and the reduction from worst-case lattices problems was established in [Pei09]
3. The GSW FHE is from [GSW13]. The presentation here is from Halevi's survey [Hal17].
4. To learn more about lattices-based cryptography, the survey by Peikert [Pei16] is an excellent source.

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 and how to play mental poker keeping secret all partial information.

In *14th ACM STOC*, pages 365–377. ACM Press, May 1982.

Craig Gentry, Amit Sahai, and Brent Waters.

Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based.

In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92. Springer, Heidelberg, August 2013.

Shai Halevi.

Homomorphic encryption.

In *Tutorials on the Foundations of Cryptography*, pages 219–276. Springer International Publishing, 2017.

Chris Peikert.

Public-key cryptosystems from the worst-case shortest vector problem: extended abstract.
In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 333–342. ACM Press, May / June 2009.

Chris Peikert.
A decade of lattice cryptography.
*Found. Trends Theor. Comput. Sci.*, 10(4):283–424, 2016.

Oded Regev.
On lattices, learning with errors, random linear codes, and cryptography.
In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.