

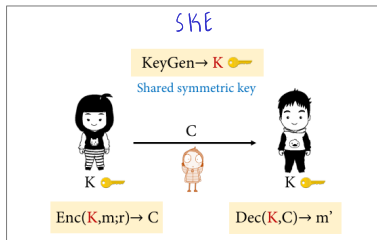
Fully-Homomorphic Encryption

Chethan Kamath

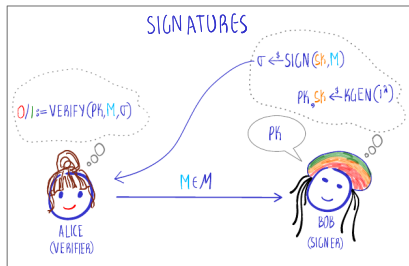
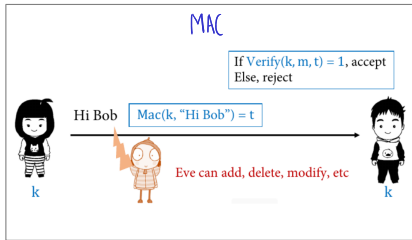
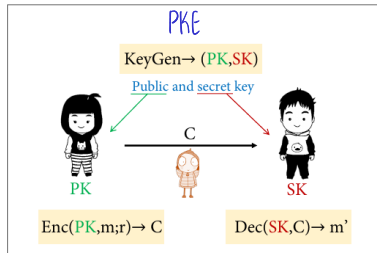


Recall from Yesterday's Sessions

SYMMETRIC KEY UNIVERSE

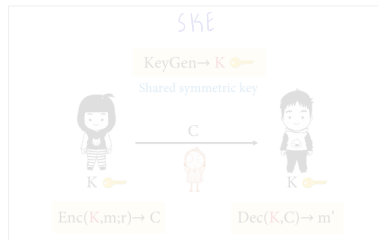


PUBLIC KEY UNIVERSE

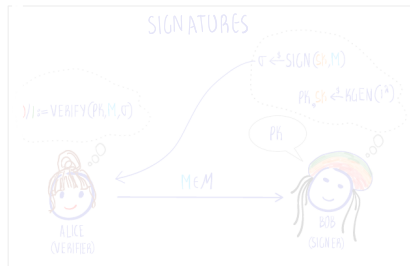
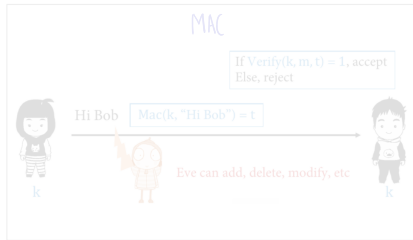
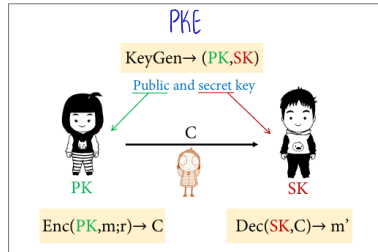


Recall from Yesterday's Sessions

SYMMETRIC KEY UNIVERSE

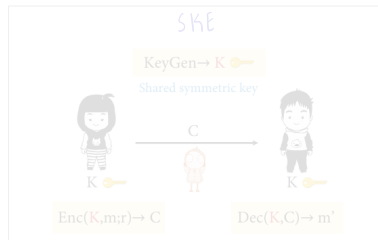


PUBLIC KEY UNIVERSE

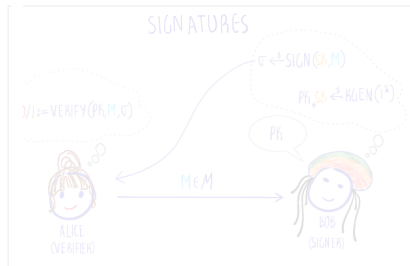
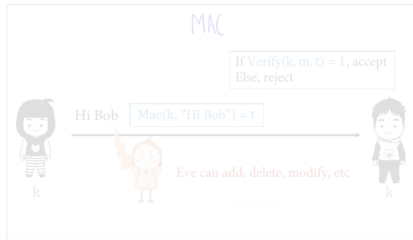
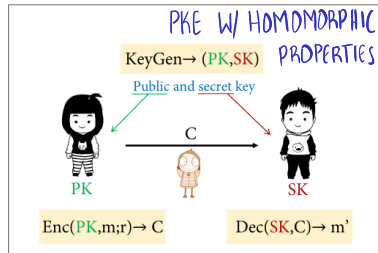


Recall from Yesterday's Sessions

SYMMETRIC KEY UNIVERSE



PUBLIC KEY UNIVERSE



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

(SENDER)



ALICE

(RECEIVER)



BOB

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

Example 1: Elgamal Encryption

(SENDER)



ALICE

$$\begin{aligned} (\mathbb{G}, g, x) &= SK \leftarrow \text{KGEN}(1^\lambda) \\ (\mathbb{G}, g, g^x) &= PK \end{aligned}$$



BOB

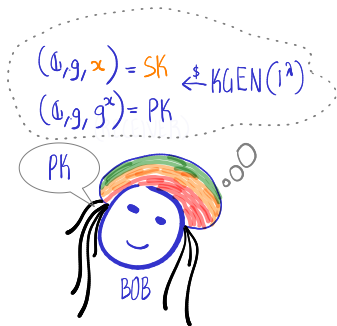
- ▶ What happens when we multiply ciphertexts?
- ▶ Is it possible to compute sum of plaintexts?

Example 1: Elgamal Encryption

(SENDER)

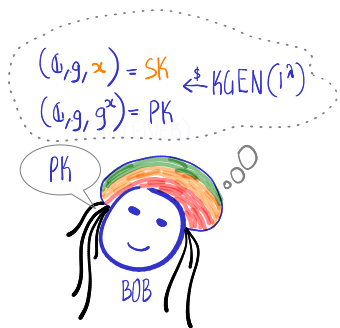
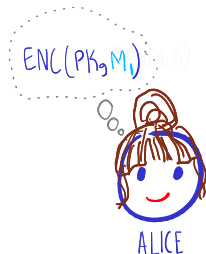


ALICE



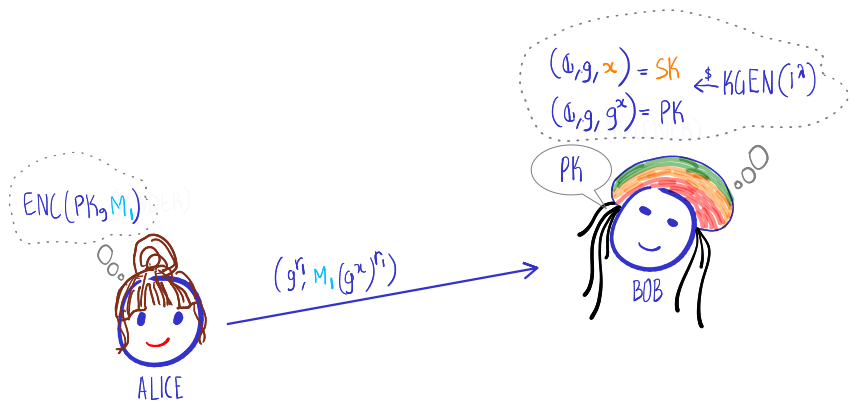
- ▶ What happens when we multiply ciphertexts?
- ▶ Is it possible to compute sum of plaintexts?

Example 1: Elgamal Encryption



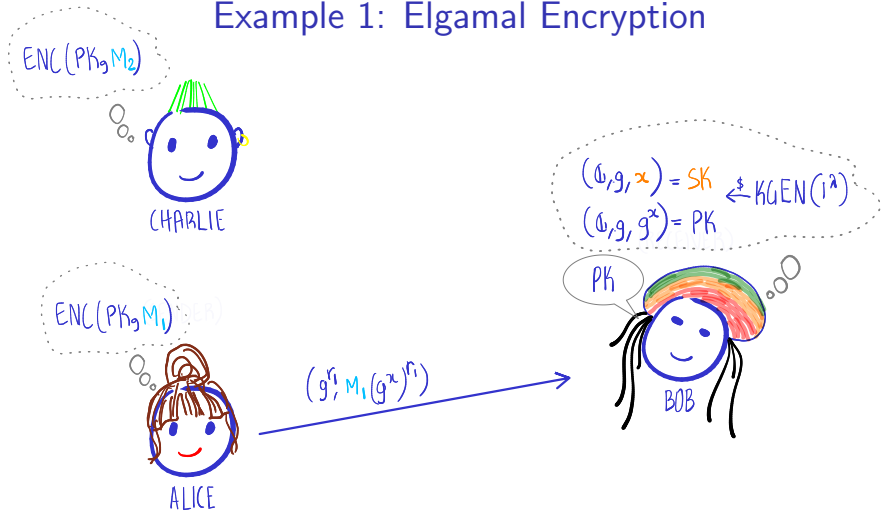
- ▶ What happens when we multiply ciphertexts?
- ▶ Is it possible to compute sum of plaintexts?

Example 1: Elgamal Encryption



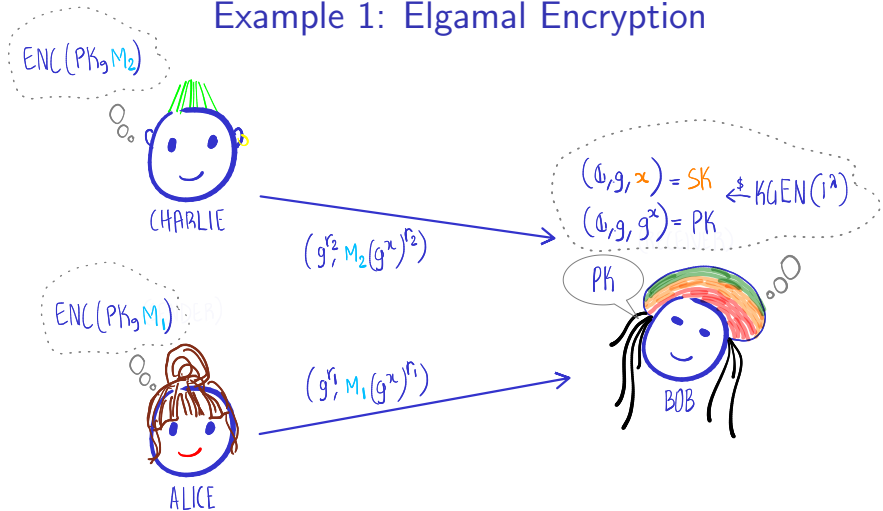
- ▶ What happens when we multiply ciphertexts?
- ▶ Is it possible to compute sum of plaintexts?

Example 1: Elgamal Encryption



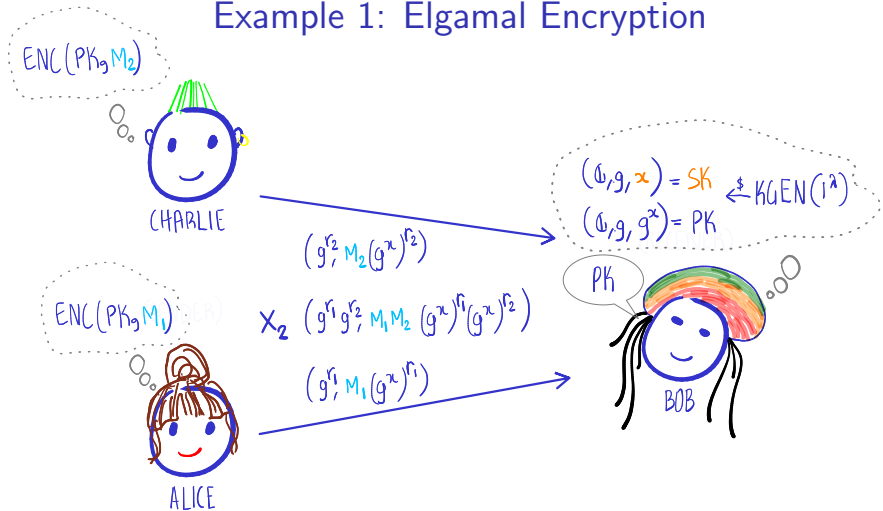
- ▶ What happens when we multiply ciphertexts?
- ▶ Is it possible to compute sum of plaintexts?

Example 1: Elgamal Encryption



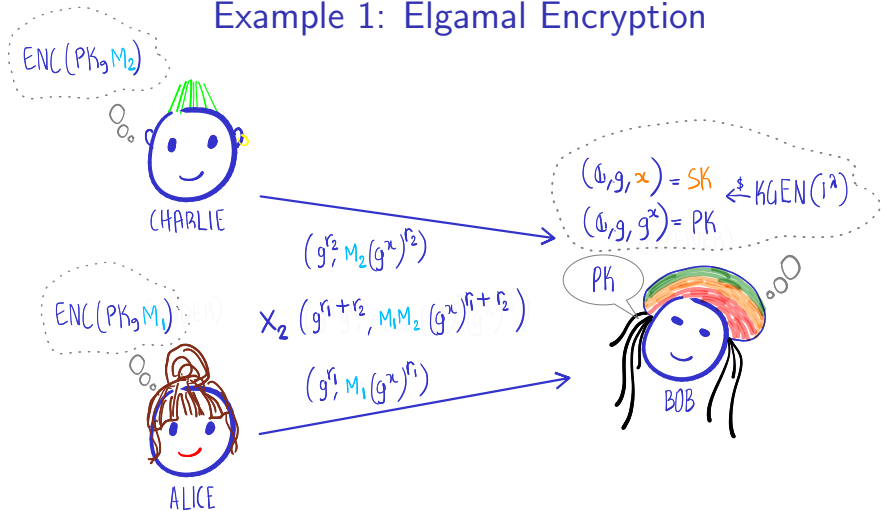
- ▶ What happens when we multiply ciphertexts?
- ▶ Is it possible to compute **sum** of plaintexts?

Example 1: Elgamal Encryption



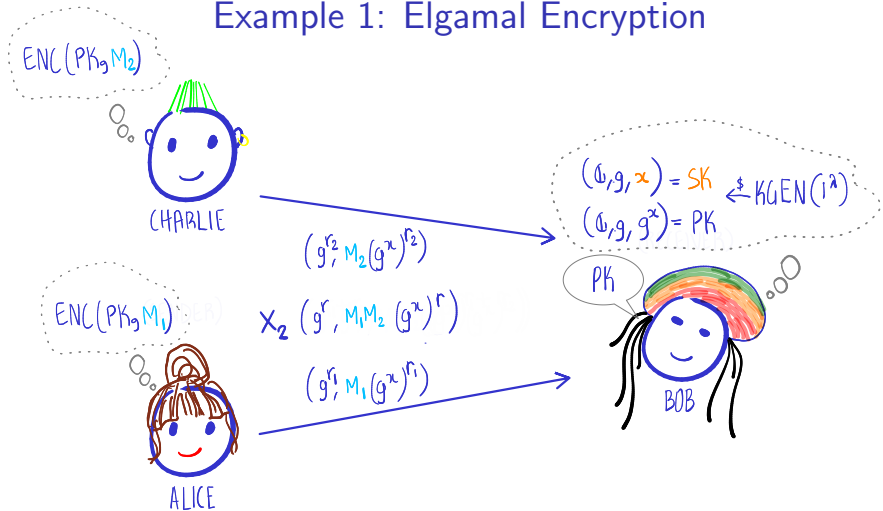
- ▶ What happens when we multiply ciphertexts?
- ▶ Is it possible to compute **sum** of plaintexts?

Example 1: Elgamal Encryption



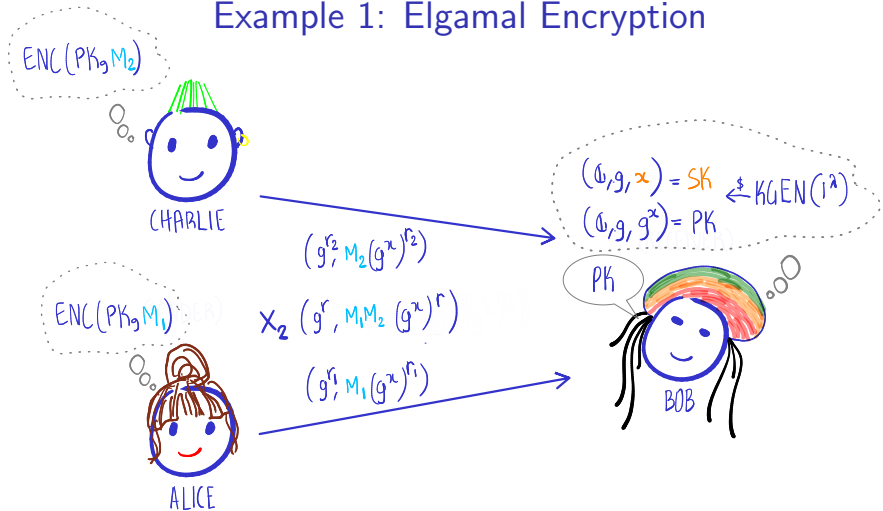
- ▶ What happens when we multiply ciphertexts?
- ▶ Is it possible to compute **sum** of plaintexts?

Example 1: Elgamal Encryption



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

Example 1: Elgamal Encryption



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

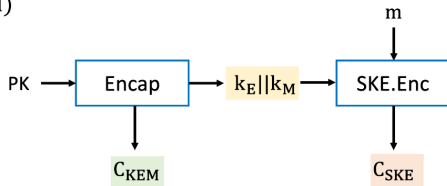
What about DHIES?

Diffie-Hellman Integrated Encryption Scheme (DHIES) IND-CCA Hybrid Encryption

KeyGen: Uses Gen to get (\mathbb{G}, q, 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($k_E || k_M, m$):
 $C_{SKE} = (C = \text{Enc}_{k_E}(m), \text{MAC}_{k_M}(C))$



Exercise 1

What happens when we (say) XOR ciphertexts?

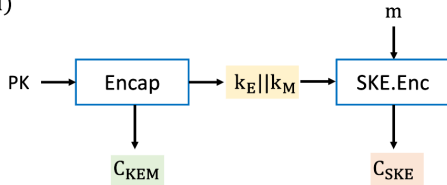
What about DHIES?

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



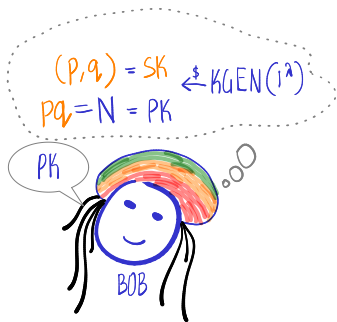
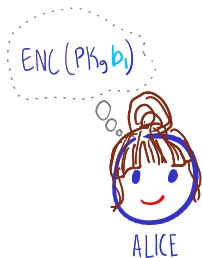
Exercise 1

What happens when we (say) XOR ciphertexts?

Example 2: Goldwasser-Micali Bit Encryption

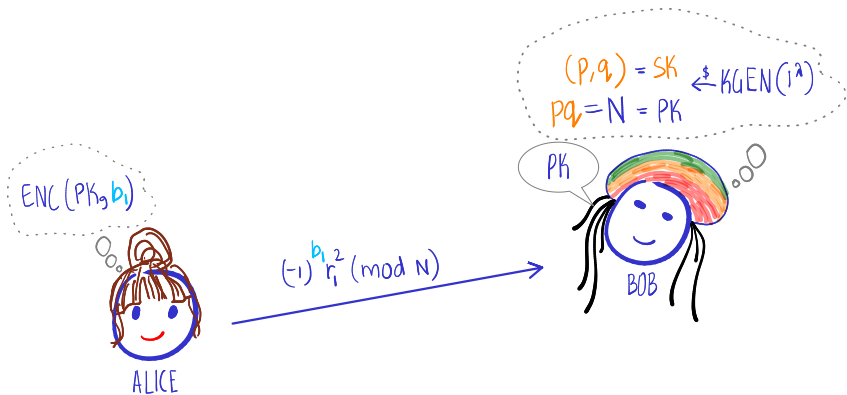
- ▶ What happens when we multiply ciphertexts?
- ▶ Is it possible compute product of plaintexts (modulo 2)?

Example 2: Goldwasser-Micali Bit Encryption



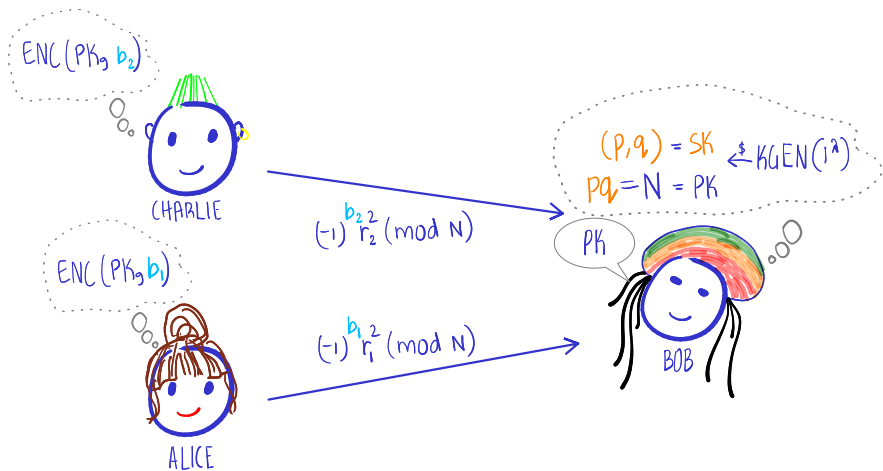
- ▶ What happens when we multiply ciphertexts?
- ▶ Is it possible compute product of plaintexts (modulo 2)?

Example 2: Goldwasser-Micali Bit Encryption



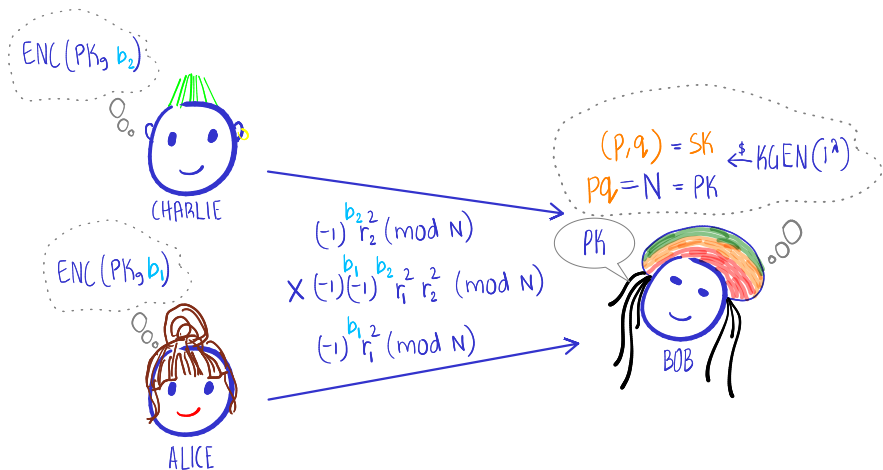
- ▶ What happens when we multiply ciphertexts?
- ▶ Is it possible compute product of plaintexts (modulo 2)?

Example 2: Goldwasser-Micali Bit Encryption



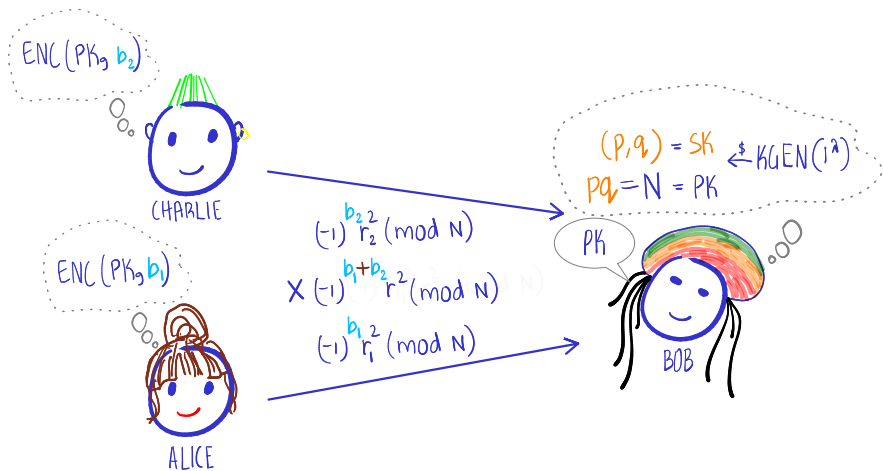
- ▶ What happens when we multiply ciphertexts?
- ▶ Is it possible compute product of plaintexts (modulo 2)?

Example 2: Goldwasser-Micali Bit Encryption



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

Example 2: Goldwasser-Micali Bit Encryption



- ▶ What happens when we multiply ciphertexts?
- ▶ Is it possible to compute the 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

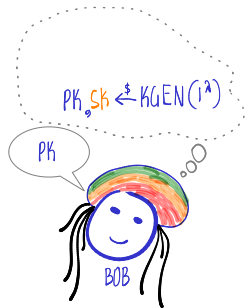
- ▶ Public-key encryption with additional *evaluation* algorithm
 - ▶ Four-tuple of algorithms: (KGEN, ENC, DEC, EVAL)



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

Defining Homomorphic Encryption

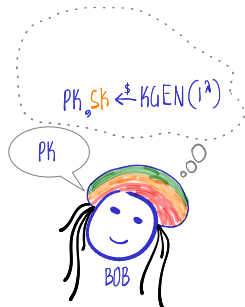
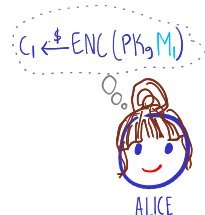
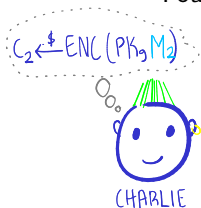
- ▶ Public-key encryption with additional *evaluation* algorithm
 - ▶ Four-tuple of algorithms: (KGEN, ENC, DEC, EVAL)



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

Defining Homomorphic Encryption

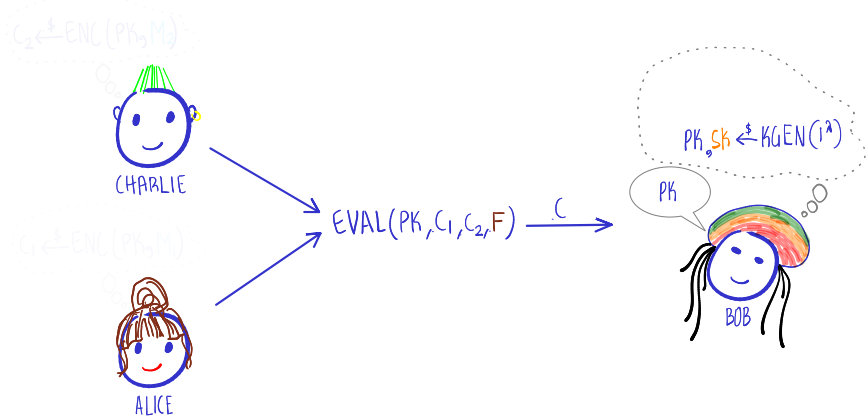
- ▶ Public-key encryption with additional *evaluation* algorithm
 - ▶ Four-tuple of algorithms: (KGEN, ENC, DEC, EVAL)



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

Defining Homomorphic Encryption

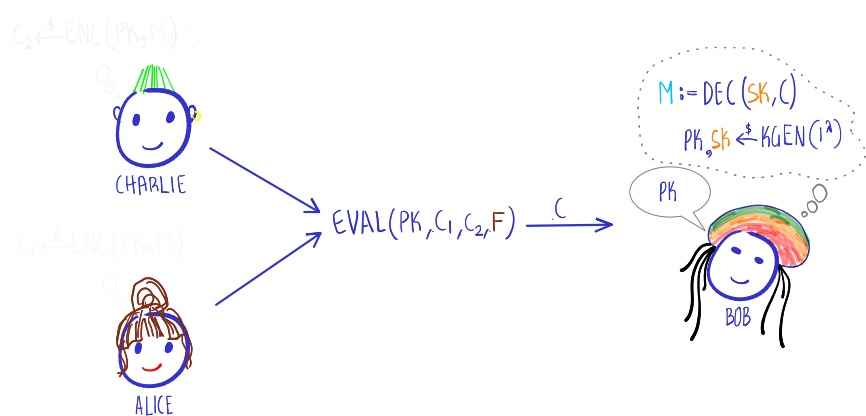
- ▶ Public-key encryption with additional *evaluation* algorithm
 - ▶ Four-tuple of algorithms: (KGEN, ENC, DEC, EVAL)



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

Defining Homomorphic Encryption

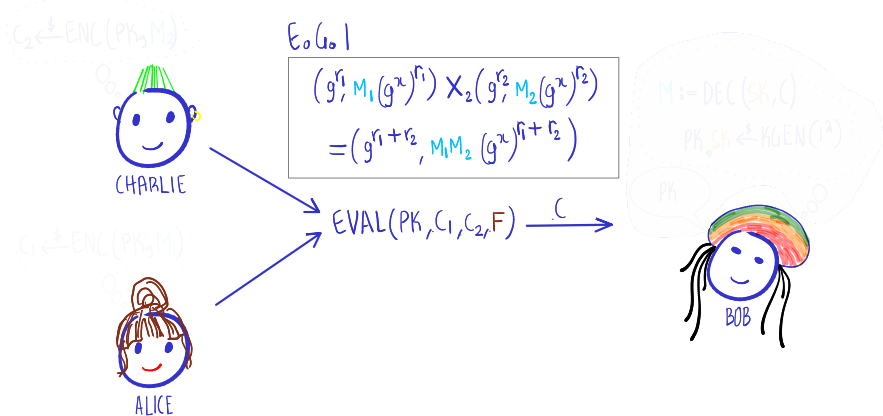
- ▶ Public-key encryption with additional *evaluation* algorithm
 - ▶ Four-tuple of algorithms: (KGEN, ENC, DEC, EVAL)



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

Defining Homomorphic Encryption

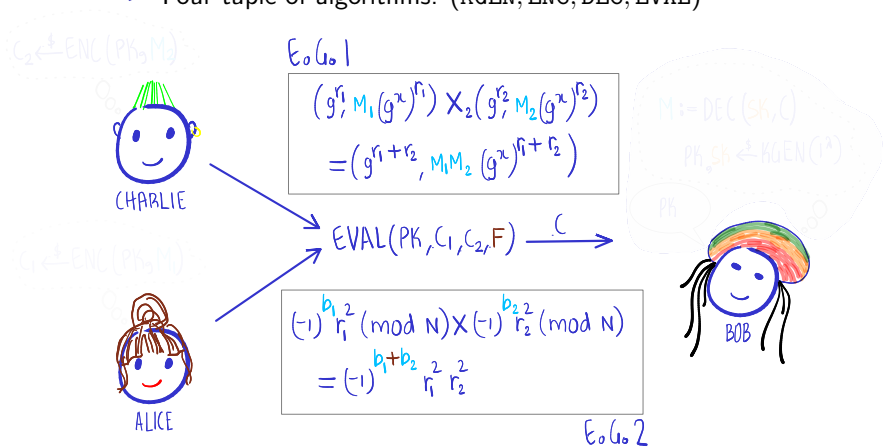
- ▶ Public-key encryption with additional *evaluation* algorithm
 - ▶ Four-tuple of algorithms: (KGEN, ENC, DEC, EVAL)



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

Defining Homomorphic Encryption

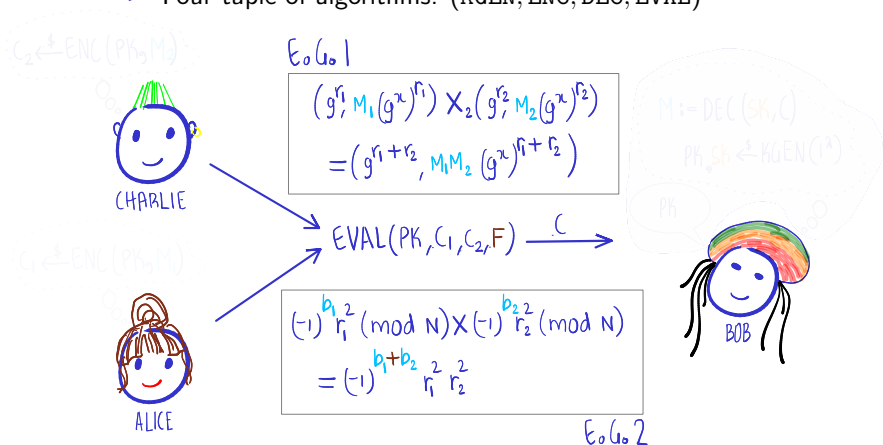
- ▶ Public-key encryption with additional *evaluation* algorithm
 - ▶ Four-tuple of algorithms: (KGEN, ENC, DEC, EVAL)



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

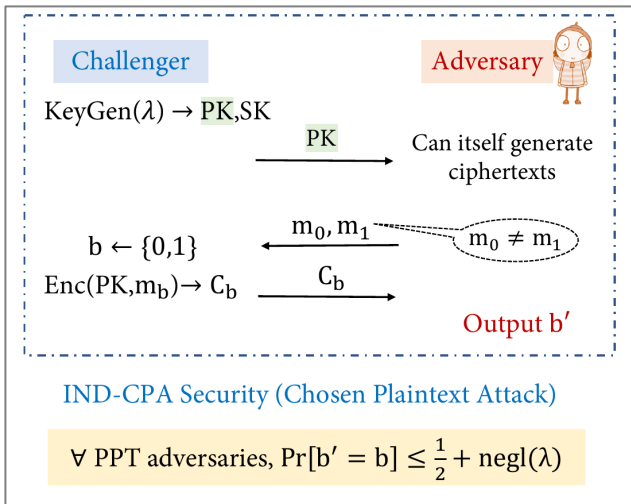
Defining Homomorphic Encryption

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



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

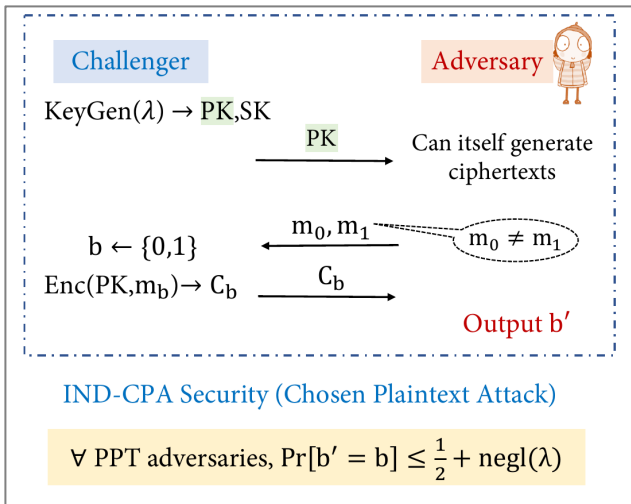
Security Model: IND-CPA for PKE



Exercise 2 (IND-CCA)

Can FHE be IND-CCA secure?

Security Model: IND-CPA for PKE



Exercise 2 (IND-CCA)

Can FHE be IND-CCA secure?

What is FHE Useful for?

- ▶ Privacy-preserving outsourcing of computation

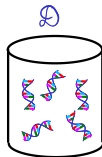


What is FHE Useful for?

- ▶ Privacy-preserving outsourcing of computation

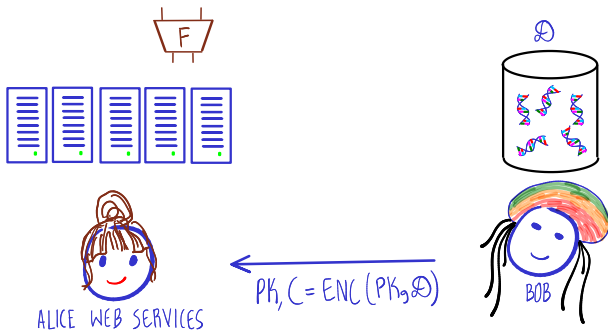



ALICE WEB SERVICES



What is FHE Useful for?

- ▶ Privacy-preserving outsourcing of computation



What is FHE Useful for?

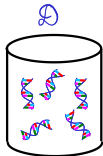
- ▶ Privacy-preserving outsourcing of computation

$$C' = \text{EVAL}(PK, C, F)$$



ALICE WEB SERVICES

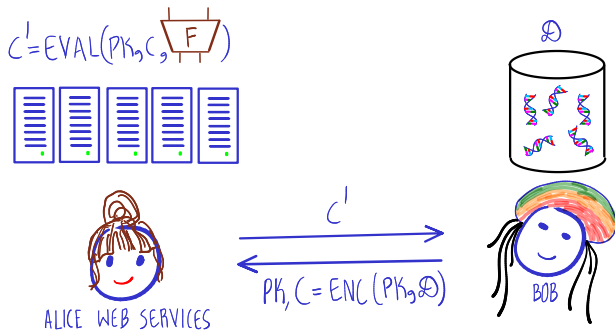
$$\leftarrow PK, C = \text{ENC}(PK, \mathcal{D})$$



BOB

What is FHE Useful for?

- ▶ Privacy-preserving outsourcing of computation



Plan for this Session

Homomorphic Encryption

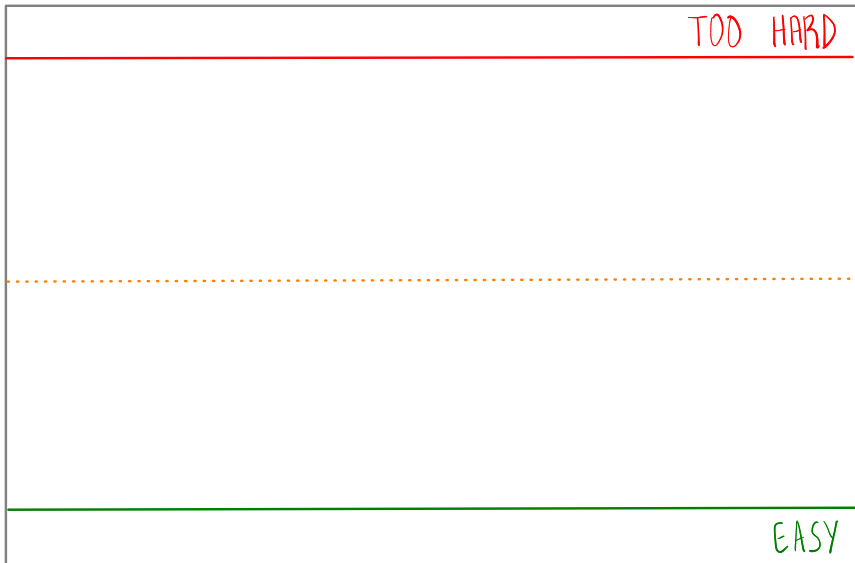
Fully-Homomorphic Encryption (FHE)

Learning with Errors (LWE)

Gentry-Sahai-Waters FHE from LWE

Wrapping Up

Cryptography Landscape



Cryptography Landscape

TOD HARD

UNSTRUCTURED HARDNESS
(MINICRYPT)

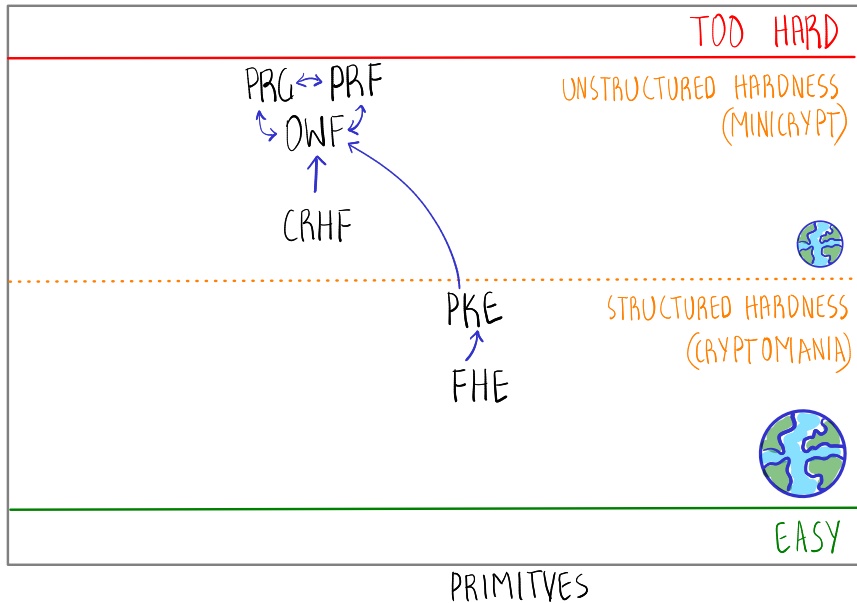


STRUCTURED HARDNESS
(CRYPTOMANIA)

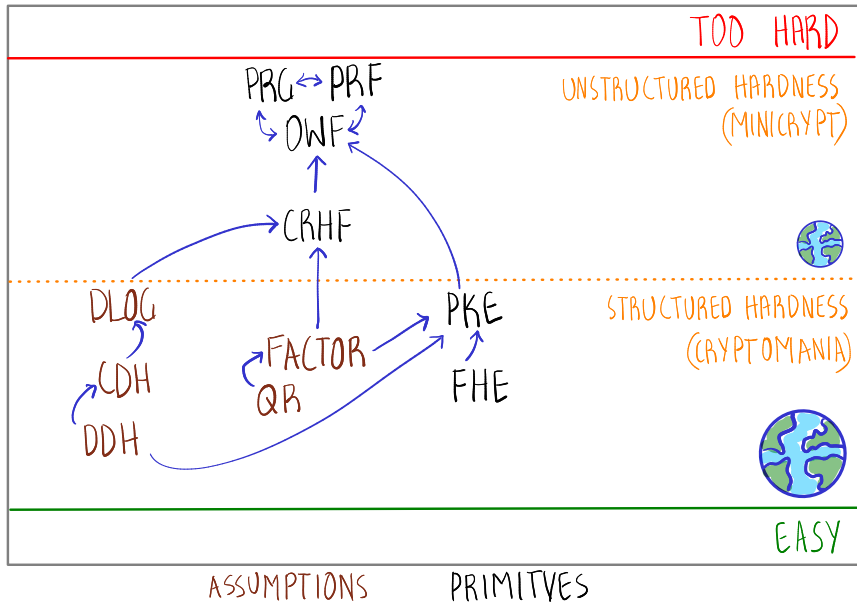


EASY

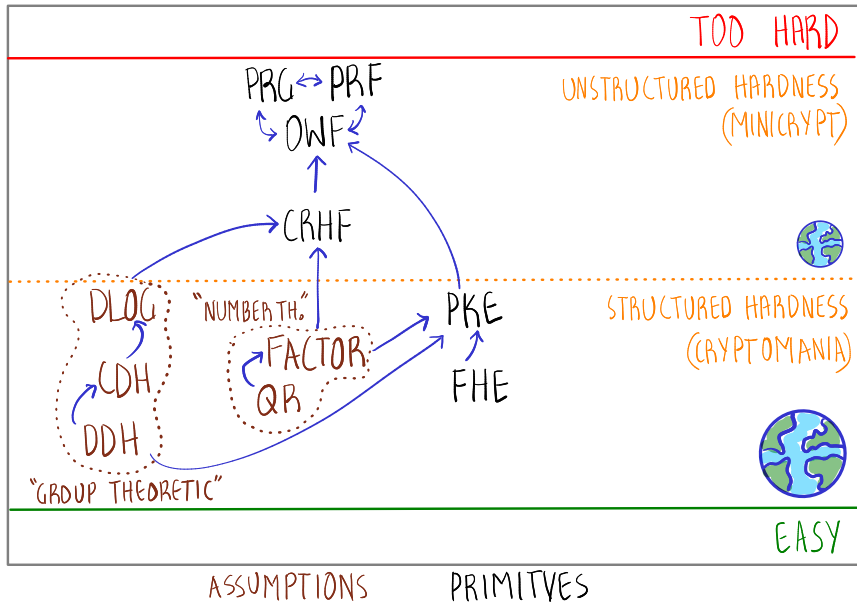
Cryptography Landscape



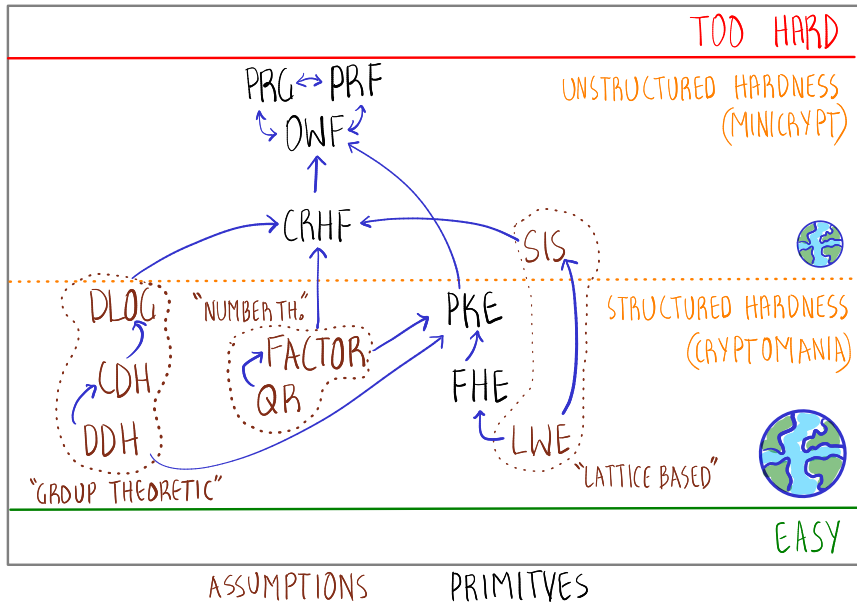
Cryptography Landscape



Cryptography Landscape



Cryptography Landscape

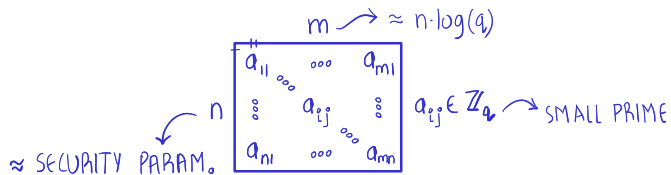


LWE: Solving “Noisy” Linear Equations is Hard

$$\begin{array}{c} \begin{array}{c} \text{m} \\ \hline a_{11} \quad \dots \quad a_{1m} \\ \vdots \quad \quad \quad \vdots \\ a_{n1} \quad \dots \quad a_{nm} \end{array} \\ \text{n} \end{array}$$

- ▶ 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

\bar{x} (orange box) \times $\begin{matrix} m \approx n \cdot \log(q) \\ \begin{matrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{matrix} \\ \bar{A} \end{matrix}$ $=$ $\begin{matrix} \bar{b} \end{matrix}$ (blue box) $(\text{mod } q)$

n \approx SECURITY PARAM. $a_{ij} \in \mathbb{Z}_q \rightarrow$ SMALL PRIME

► 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

\bar{s} (orange box) \rightarrow SECURITY PARAM.

\bar{A} (blue box) \rightarrow matrix with entries $a_{ij} \in \mathbb{Z}_q$ (SMALL PRIME). Dimensions: n (rows), m (columns). $m \approx n \cdot \log(q)$.

\bar{b} (blue box) \rightarrow vector $(\text{mod } q)$.

$$\bar{s} \bar{A} = \bar{b} \pmod{q}$$

- ▶ Search vs decision LWE



ELIMINATION

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

LWE: Solving "Noisy" Linear Equations is Hard

$$\bar{s} \cdot \bar{A} + \bar{e} = \bar{b} \pmod{q}$$

\bar{s} (orange box) \bar{A} (blue box, $n \times m$) \bar{e} (red box) \bar{b} (blue box)

$m \approx n \cdot \log(q)$

$a_{ij} \in \mathbb{Z}_q \rightarrow$ SMALL PRIME

\approx SECURITY PARAM.

- ▶ Search vs decision LW

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

LWE: Solving "Noisy" Linear Equations is Hard

$$\bar{s} + \bar{A} \bar{e} = \bar{b} \pmod{q}$$

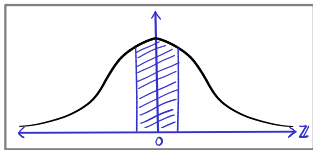
\bar{s} (orange box) \bar{A} (matrix, $n \times m$) \bar{e} (red box) \bar{b} (blue box) $(\text{mod } q)$

$\approx n \cdot \log(q)$ (handwritten above m)

$e_j \leftarrow \text{DISCRETE GAUSSIAN OF "WIDTH" } \alpha q$ (handwritten next to \bar{e})

$\approx \text{SECURITY PARAM.}$ (handwritten next to n)

$a_{ij} \in \mathbb{Z}_q$ (handwritten next to matrix)



► Search vs decision LW

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

LWE: Solving “Noisy” Linear Equations is Hard

The diagram illustrates the LWE equation: $\bar{s} \bar{A} + \bar{e} = \bar{b} \pmod{q}$. The vector \bar{s} is shown in an orange box, the matrix \bar{A} is in a blue box with dimensions n (rows) and m (columns), the vector \bar{e} is in a red box, and the vector \bar{b} is in a blue box. Handwritten notes include: $m \approx n \cdot \log(q)$, $\bar{e}_j \leftarrow \text{DISCRETE GAUSSIAN OF WIDTH } \sigma$, and $\sigma \approx \text{SECURITY PARAM.}$. A faint diagram of a discrete Gaussian distribution is also visible.

- ▶ 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

$$s \cdot \bar{A} + \bar{e} = \bar{b} \pmod{q}$$

n (rows), m (columns) for \bar{A} . $\approx n \cdot \log_2(q)$ for \bar{e} . $e_j \leftarrow \text{DISCRETE GAUSSIAN OF "WIDTH" } n$. $\approx \text{SECURITY PARAM.}$

- Search vs decision LWE

$$\begin{matrix} m \\ \hline \bar{A} \\ \hline \bar{b} \\ n \end{matrix} \approx_c \begin{matrix} m \\ \hline \bar{A} \\ \hline \bar{u} \\ n \end{matrix}$$

- 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...

(SENDER)

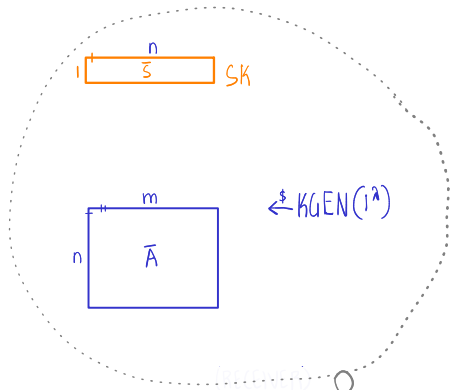


(RECEIVER)



► What happens when you add two ciphertexts?

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



(SENDER)

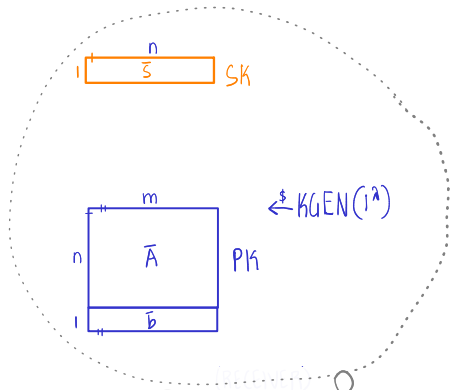


(RECEIVER)



► What happens when you add two ciphertexts?

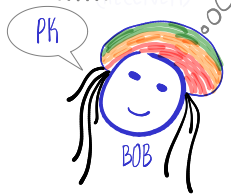
Regev's Bit Encryption: PKE from LWE...



(SENDER)



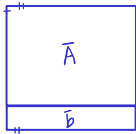
(RECEIVER)



► What happens when you add two ciphertexts?

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

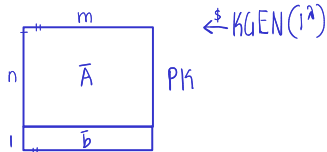
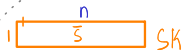
$ENC(PK, b)$



(SENDER)

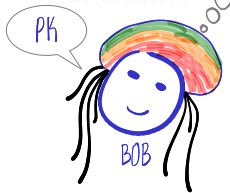


ALICE



$\leftarrow KGEN(1^n)$

(RECEIVER)

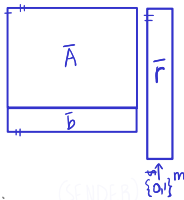


BOB

► What happens when you add two ciphertexts?

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

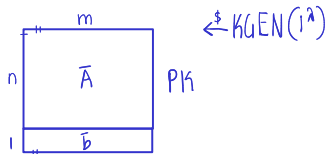
$ENC(PK, b)$



(SENDER)

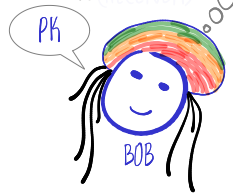


\bar{s} SK



$\leftarrow KGEN(1^n)$

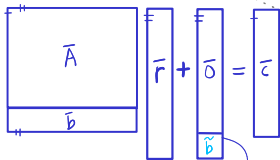
(RECEIVER)



► What happens when you add two ciphertexts?

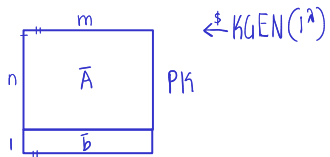
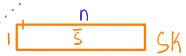
Regev's Bit Encryption: PKE from LWE...

ENC(PK, b)



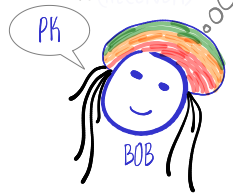
(SENDER)

"SCALED b " := $b \cdot \lfloor q/2 \rfloor$



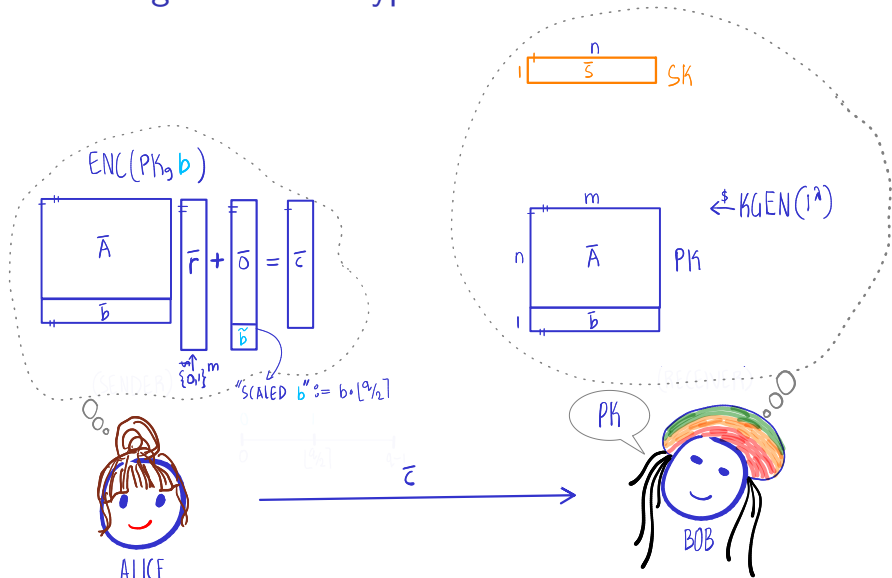
$\leftarrow KGEN(1^n)$

(RECEIVER)



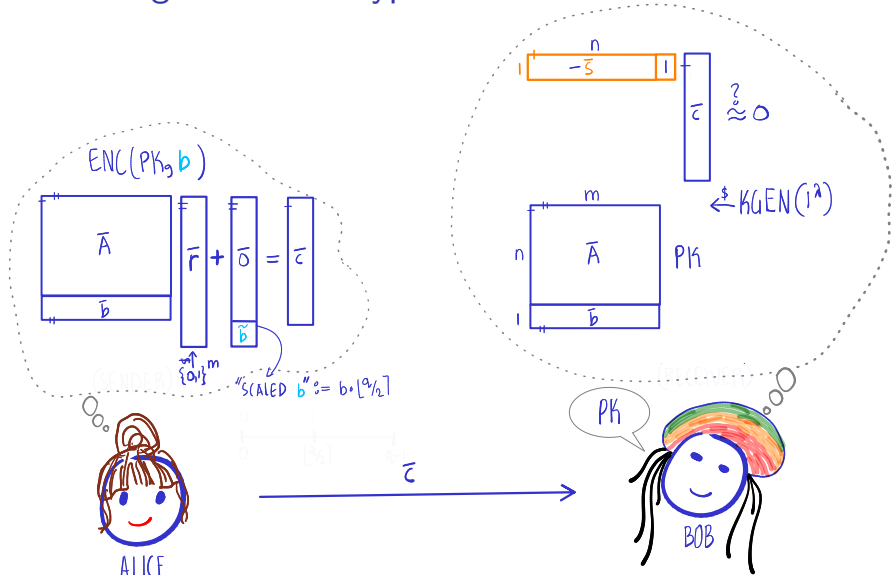
► What happens when you add two ciphertexts?

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



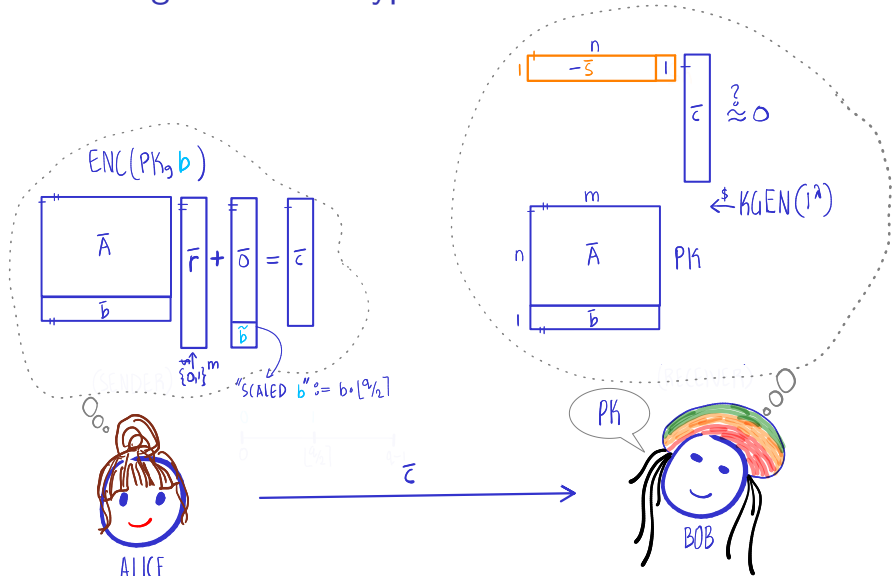
► What happens when you add two ciphertexts?

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...

- ▶ Correctness:

$$\left(\begin{array}{c|c} \bar{A} & \bar{r} \\ \hline & \tilde{b} \end{array} \right) = \left(\begin{array}{c|c} \bar{A} & \tilde{r} \\ \hline & \tilde{\tilde{b}} \end{array} \right)$$

- ▶ Security by hybrid argument

Exercise 3 (Security of Regev's Encryption)

Prove security formally.

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

- ▶ Correctness:

$$\begin{pmatrix} \boxed{-\bar{s}} & | & 1 \\ \hline \bar{A} & & \bar{r} \\ \hline & & \bar{b} \end{pmatrix} + \begin{pmatrix} \bar{0} \\ \tilde{b} \end{pmatrix} = \begin{pmatrix} \boxed{-\bar{s}\bar{A} + \bar{b}} \\ \bar{r} \end{pmatrix} + \boxed{\tilde{b}}$$

- ▶ Security by hybrid argument

Exercise 3 (Security of Regev's Encryption)

Prove security formally.

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

- ▶ Correctness:

$$\begin{pmatrix} \boxed{-\bar{s}} & | & 1 \\ \hline \bar{A} & & \bar{r} \\ \hline & & \bar{b} \end{pmatrix} + \begin{pmatrix} \bar{0} \\ \tilde{b} \end{pmatrix} = \begin{pmatrix} \boxed{-\bar{e}} & | & 1 \\ \hline \bar{r} \end{pmatrix} + \boxed{\tilde{b}}$$

- ▶ Security by hybrid argument

Exercise 3 (Security of Regev's Encryption)

Prove security formally.

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

- ▶ Correctness:

$$\begin{pmatrix} -\bar{s} & 1 \\ \bar{A} & \bar{b} \end{pmatrix} \begin{pmatrix} \bar{r} \\ \bar{o} \end{pmatrix} = \begin{pmatrix} \bar{e} \\ \tilde{b} \end{pmatrix}$$

The diagram shows a number line from 0 to $q-1$. The interval $[0, \lfloor q/2 \rfloor]$ is marked with a red hatched line and labeled '0'. The interval $[\lfloor q/2 \rfloor, q-1]$ is marked with a red hatched line and labeled '1'. Arrows point from the ciphertext components \bar{e} and \tilde{b} to these intervals on the number line.

- ▶ Security by hybrid argument

Exercise 3 (Security of Regev's Encryption)

Prove security formally.

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

- ▶ Correctness:

$$\begin{pmatrix} \bar{A} \\ \bar{b} \end{pmatrix} \begin{pmatrix} \bar{r} \\ \bar{0} \end{pmatrix} + \begin{pmatrix} \bar{b} \\ \bar{0} \end{pmatrix} = \begin{pmatrix} \bar{r} \\ \bar{0} \end{pmatrix} + \begin{pmatrix} \bar{b} \\ \bar{0} \end{pmatrix} + \begin{pmatrix} \bar{s} \\ \bar{0} \end{pmatrix} = \begin{pmatrix} \bar{e} \\ \bar{0} \end{pmatrix}$$

The diagram illustrates the correctness of Regev's bit encryption. It shows the encryption of a bit s into ciphertext e . The encryption process is shown as a matrix multiplication of a matrix \bar{A} with a vector \bar{r} , plus a vector \bar{b} , plus a scalar \bar{s} . The result is a ciphertext vector \bar{e} . The decryption process is shown as a dot product of \bar{e} with a vector \tilde{r} , resulting in a bit \tilde{s} . The diagram uses color coding: orange for s and \tilde{s} , red for e and \tilde{e} , and blue for r , \tilde{r} , b , and \tilde{b} . A number line at the bottom shows the distribution of the ciphertext e , with a peak at 0 and a peak at 1, indicating that the ciphertext is close to 0 or 1 with high probability.

- ▶ Security by hybrid argument

Exercise 3 (Security of Regev's Encryption)

Prove security formally.

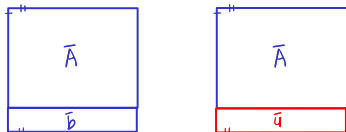
Regev's Bit Encryption: PKE from LWE...

- ▶ Correctness:

$$\begin{pmatrix} \bar{A} \\ \bar{b} \end{pmatrix} \begin{pmatrix} \bar{r} \\ \bar{0} \end{pmatrix} + \begin{pmatrix} \bar{0} \\ \tilde{b} \end{pmatrix} = \begin{pmatrix} \bar{e} \\ \tilde{b} \end{pmatrix}$$

The diagram shows a matrix \bar{A} and a vector $\begin{pmatrix} \bar{r} \\ \bar{0} \end{pmatrix}$ being multiplied together, and then a vector $\begin{pmatrix} \bar{0} \\ \tilde{b} \end{pmatrix}$ is added to the result. The result is a ciphertext vector $\begin{pmatrix} \bar{e} \\ \tilde{b} \end{pmatrix}$. Below this, a number line is shown with a peak at 0 and a peak at $q/2$, indicating that the ciphertext is close to 0 if $s=0$ and close to $q/2$ if $s=1$.

- ▶ Security by hybrid argument



Exercise 3 (Security of Regev's Encryption)

Prove security formally.

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

- ▶ Correctness:

$$\begin{pmatrix} \bar{A} \\ \bar{b} \end{pmatrix} \bar{r} + \begin{pmatrix} \bar{o} \\ \tilde{b} \end{pmatrix} = \begin{pmatrix} \bar{e} \\ \tilde{b} \end{pmatrix} + \begin{pmatrix} \bar{r} \\ \tilde{b} \end{pmatrix}$$

The diagram also includes a number line from 0 to $q-1$. The value 0 is marked with a red slash, and $\lfloor q/2 \rfloor$ is also marked with a red slash. Arrows point from the ciphertext \tilde{b} to these two points on the number line, indicating that the ciphertext is either near 0 or near $q/2$.

- ▶ Security by hybrid argument

$$\begin{pmatrix} \bar{A} \\ \bar{b} \end{pmatrix} \approx_c \begin{pmatrix} \bar{A} \\ \bar{u} \end{pmatrix}$$

Exercise 3 (Security of Regev's Encryption)

Prove security formally.

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

- ▶ Correctness:

Diagram illustrating the correctness of Regev's bit encryption. The encryption process is shown as:

$$\left(\begin{array}{c} \bar{A} \\ \bar{b} \end{array} \right) + \bar{r} = \bar{e} + \tilde{b}$$

The ciphertext \bar{e} is a vector of length q . The bit b is hidden in the middle of the ciphertext, specifically in the range $[q/2]$.

- ▶ Security by hybrid argument

Diagram illustrating the security of Regev's bit encryption. The encryption process is shown as:

$$\left(\begin{array}{c} \bar{A} \\ \bar{b} \end{array} \right) \approx_c \left(\begin{array}{c} \bar{A} \\ \bar{a} \end{array} \right) + \bar{r}$$

The vector \bar{a} is shown as a red box, indicating it is statistically hidden. The text "STATISTICALLY HIDES b " and "LEFTOVER HASH LEMMA" is written next to the diagram.

Exercise 3 (Security of Regev's Encryption)

Prove security formally.

Plan for this Session

Homomorphic Encryption

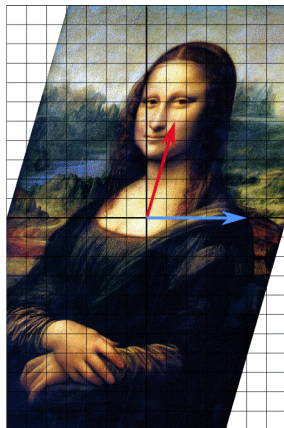
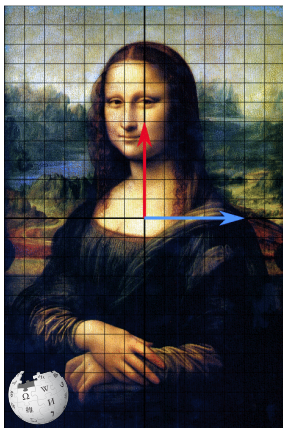
Fully-Homomorphic Encryption (FHE)

Learning with Errors (LWE)

Gentry-Sahai-Waters FHE from LWE

Wrapping Up

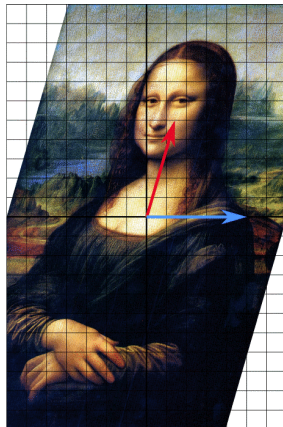
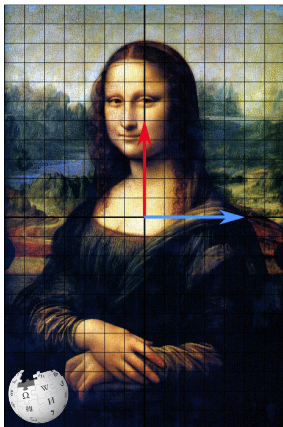
Let's Recall Eigenvectors



Definition 1

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

Let's Recall Eigenvectors

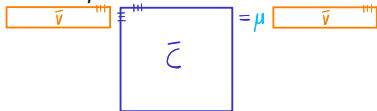


Definition 1

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 μ , which is the **eigenvalue**.

Toy Example: "Eigenvector" Encryption

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

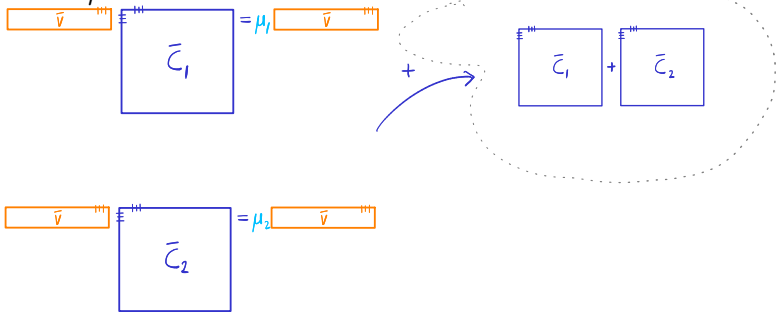
A diagram illustrating the encryption process. On the left, an orange box contains the vector \bar{v} . An arrow points from this box to a blue square box labeled \bar{C}_1 . From the right side of the \bar{C}_1 box, an arrow points to another orange box containing \bar{v} . The equation $\bar{v}\bar{C}_1 = \mu_1\bar{v}$ is written above the arrow connecting the two boxes.

A diagram illustrating the encryption process. On the left, an orange box contains the vector \bar{v} . An arrow points from this box to a blue square box labeled \bar{C}_2 . From the right side of the \bar{C}_2 box, an arrow points to another orange box containing \bar{v} . The equation $\bar{v}\bar{C}_2 = \mu_2\bar{v}$ is written above the arrow connecting the two boxes.

- ▶ Do we have an FHE?

Toy Example: "Eigenvector" Encryption

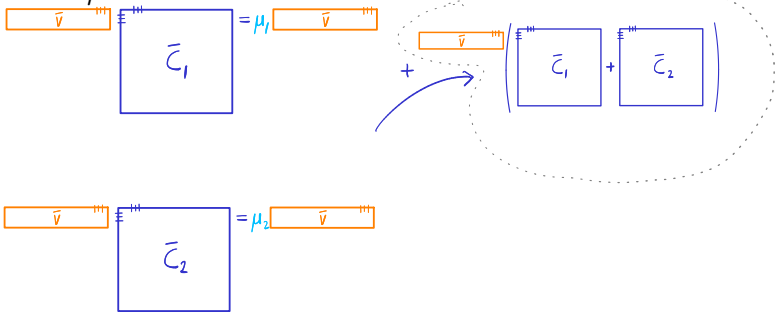
- An $N \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} = \mu\bar{v}$



- Do we have an FHE?

Toy Example: "Eigenvector" Encryption

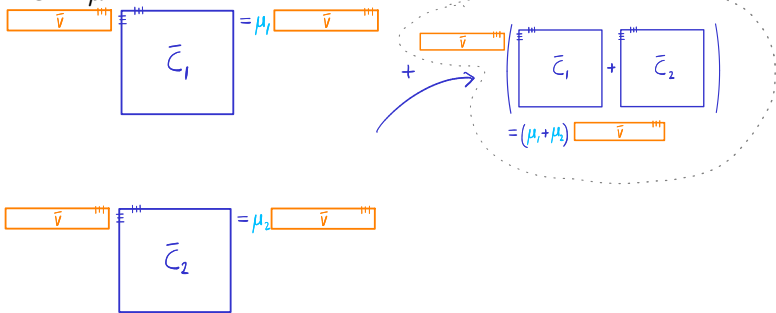
- ▶ An $N \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} = \mu\bar{v}$



- ▶ Do we have an FHE?

Toy Example: "Eigenvector" Encryption

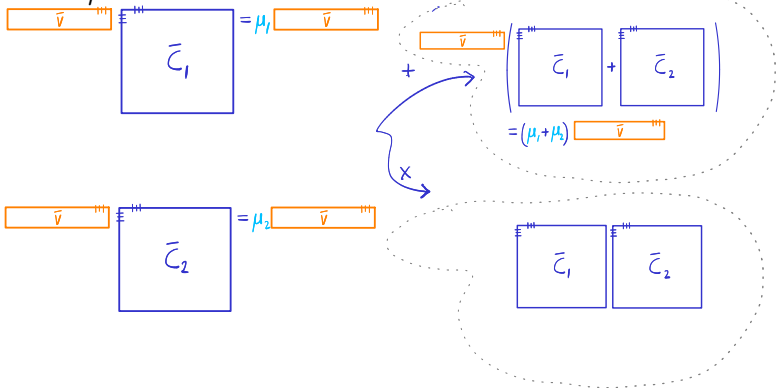
- ▶ An $N \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} = \mu\bar{v}$



- ▶ Do we have an FHE?

Toy Example: "Eigenvector" Encryption

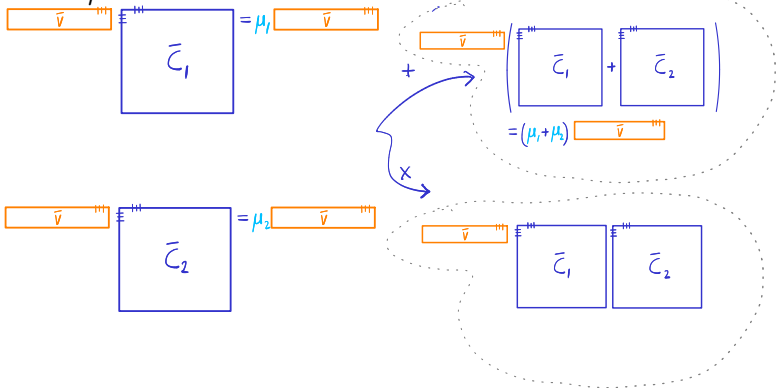
- ▶ An $N \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} = \mu\bar{v}$



- ▶ Do we have an FHE?

Toy Example: "Eigenvector" Encryption

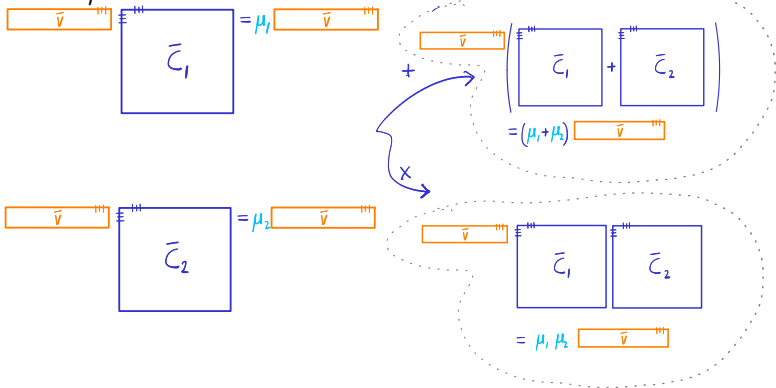
- ▶ An $N \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} = \mu\bar{v}$



- ▶ Do we have an FHE?

Toy Example: "Eigenvector" Encryption

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

$$\boxed{\bar{v}} \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \end{matrix} \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \end{matrix} \bar{C}_1 = \mu_1 \boxed{\bar{v}}$$

$$\boxed{\bar{v}} \left(\begin{matrix} \text{---} \\ \text{---} \\ \text{---} \end{matrix} \bar{C}_1 + \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \end{matrix} \bar{C}_2 \right) = (\mu_1 + \mu_2) \boxed{\bar{v}}$$

$$\boxed{\bar{v}} \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \end{matrix} \bar{C}_2 = \mu_2 \boxed{\bar{v}}$$

$$\boxed{\bar{v}} \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \end{matrix} \bar{C}_1 \quad \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \end{matrix} \bar{C}_2 = \mu_1, \mu_2 \boxed{\bar{v}}$$

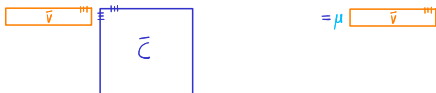
- Do we have an FHE?



ELIMINATION

How to Fix? *Approximate* Eigenvector Encryption

- ▶ \bar{C} encrypts a bit μ 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 μ , error grows exp. in levels
 - ▶ Somewhat homomorphic: levelled FHE supporting log-depth F

How to Fix? *Approximate* Eigenvector Encryption

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

$$\boxed{\bar{v}} + \boxed{\bar{C}} + \boxed{\bar{e}} = \mu \boxed{\bar{v}}$$

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

How to Fix? *Approximate* Eigenvector Encryption

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

$$\begin{array}{l} \boxed{\bar{v}} + \boxed{\bar{C}_1} + \boxed{\bar{e}_1} = \mu_1 \boxed{\bar{v}} \\ \boxed{\bar{v}} + \boxed{\bar{C}_2} + \boxed{\bar{e}_2} = \mu_2 \boxed{\bar{v}} \end{array}$$

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

How to Fix? *Approximate* Eigenvector Encryption

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

$$\boxed{\bar{v}} + \boxed{\bar{C}_1} + \boxed{\bar{e}_1} = \mu_1 \boxed{\bar{v}}$$

$$\boxed{\bar{v}} + \boxed{\bar{C}_2} + \boxed{\bar{e}_2} = \mu_2 \boxed{\bar{v}}$$

+

$$\boxed{\bar{v}} + \left(\boxed{\bar{C}_1} + \boxed{\bar{C}_2} \right)$$

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

How to Fix? *Approximate* Eigenvector Encryption

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

$$\boxed{\bar{v}} + \boxed{\bar{C}_1} + \boxed{\bar{e}_1} = \mu_1 \boxed{\bar{v}}$$

$$\boxed{\bar{v}} + \boxed{\bar{C}_2} + \boxed{\bar{e}_2} = \mu_2 \boxed{\bar{v}}$$

$$\begin{aligned} & + \left(\boxed{\bar{v}} + \left(\boxed{\bar{C}_1} + \boxed{\bar{C}_2} \right) \right) \\ & = (\mu_1 + \mu_2) \boxed{\bar{v}} - \left(\boxed{\bar{e}_1} + \boxed{\bar{e}_2} \right) \end{aligned}$$

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

How to Fix? *Approximate* Eigenvector Encryption

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

$$\boxed{\bar{v}} + \boxed{\bar{C}_1} + \boxed{\bar{e}_1} = \mu_1 \boxed{\bar{v}}$$

$$\boxed{\bar{v}} + \boxed{\bar{C}_2} + \boxed{\bar{e}_2} = \mu_2 \boxed{\bar{v}}$$

$$\begin{aligned} & + \left(\boxed{\bar{v}} + \left(\boxed{\bar{C}_1} + \boxed{\bar{C}_2} \right) \right) \\ & + \left(\begin{array}{c} \boxed{\bar{e}_1} \\ + \\ \boxed{\bar{e}_2} \end{array} \right) = (\mu_1 + \mu_2) \boxed{\bar{v}} \end{aligned}$$

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

How to Fix? *Approximate* Eigenvector Encryption

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

$$\boxed{\bar{v}} + \boxed{\bar{C}_1} = \mu_1 \boxed{\bar{v}}$$

$$\boxed{\bar{v}} + \boxed{\bar{C}_2} = \mu_2 \boxed{\bar{v}}$$

$$\boxed{\bar{v}} + \boxed{\bar{C}_1} + \boxed{\bar{C}_2}$$

$$\boxed{\bar{v}} + \left(\boxed{\bar{C}_1} + \boxed{\bar{C}_2} \right) + \left(\begin{array}{c} \boxed{\bar{e}_1} \\ + \\ \boxed{\bar{e}_2} \end{array} \right) = (\mu_1 + \mu_2) \boxed{\bar{v}}$$

- ▶ Do we have an FHE?

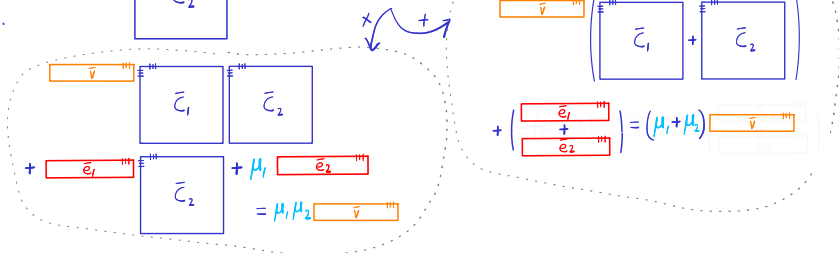
- ▶ For “ B -bounded” \bar{C} , \bar{e} and μ , error grows exp. in levels
- ▶ Somewhat homomorphic: levelled FHE supporting log-depth F

How to Fix? *Approximate* Eigenvector Encryption

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

$$\boxed{\bar{v}} + \boxed{\bar{C}_1} + \boxed{\bar{e}_1} = \mu_1 \boxed{\bar{v}}$$

$$\boxed{\bar{v}} + \boxed{\bar{C}_2} + \boxed{\bar{e}_2} = \mu_2 \boxed{\bar{v}}$$



- ▶ Do we have an FHE?

- ▶ For “ B -bounded” \bar{C} , \bar{e} and μ , error grows exp. in levels
- ▶ Somewhat homomorphic: levelled FHE supporting log-depth F

How to Fix? *Approximate* Eigenvector Encryption

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

$$\boxed{\bar{v}} + \boxed{\bar{C}_1} + \boxed{\bar{e}_1} = \mu_1 \boxed{\bar{v}}$$

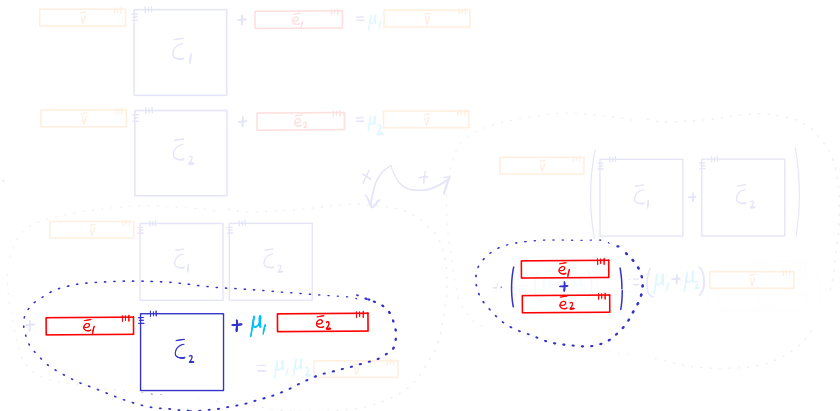
$$\boxed{\bar{v}} + \boxed{\bar{C}_2} + \boxed{\bar{e}_2} = \mu_2 \boxed{\bar{v}}$$

$$\boxed{\bar{v}} + \begin{pmatrix} \boxed{\bar{C}_1} & \boxed{\bar{C}_2} \end{pmatrix} + \begin{pmatrix} \boxed{\bar{e}_1} \\ \boxed{\bar{e}_2} \end{pmatrix} = (\mu_1 + \mu_2) \boxed{\bar{v}}$$

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

How to Fix? *Approximate* Eigenvector Encryption

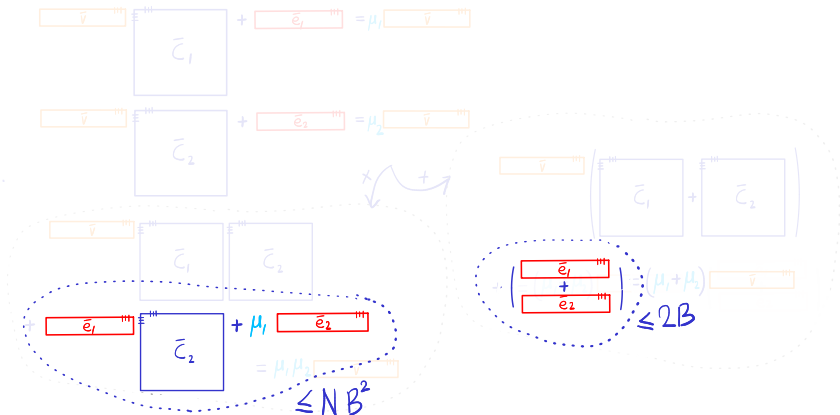
- ▶ \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for “short” \bar{e}



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

How to Fix? *Approximate* Eigenvector Encryption

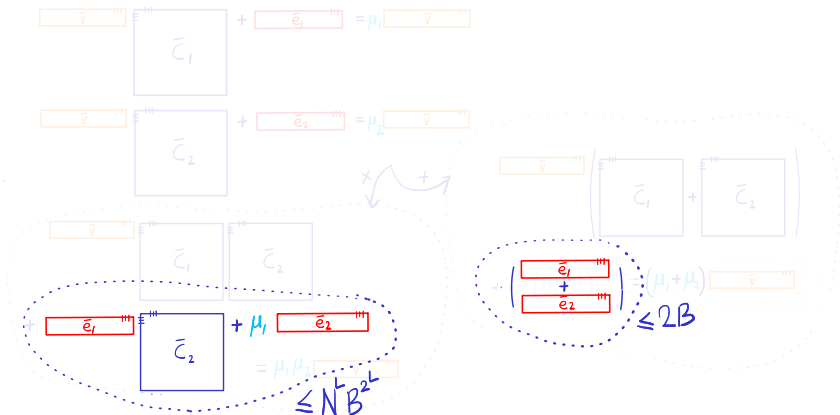
- ▶ \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for “short” \bar{e}



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

How to Fix? *Approximate* Eigenvector Encryption

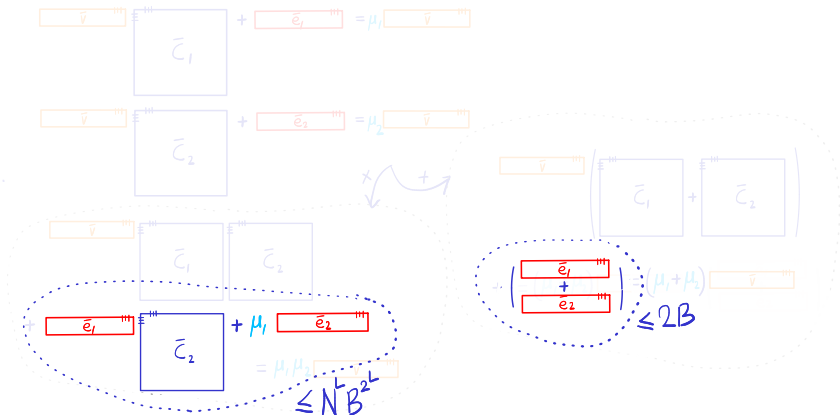
- ▶ \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for “short” \bar{e}



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

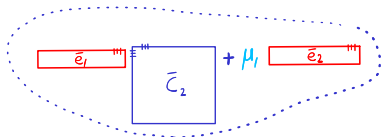
How to Fix? *Approximate* Eigenvector Encryption

- ▶ \bar{C} encrypts a bit μ 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 μ , error grows exp. in levels
 - ▶ Somewhat homomorphic: levelled FHE supporting log-depth F

Supporting Arbitrary Depth

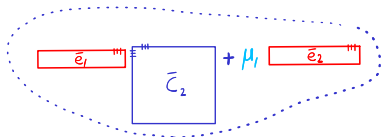


► Two tricks:

1. Stick to messages μ 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 $\bar{G} : \mathbb{Z}_q^{n \times N} \rightarrow \mathbb{Z}_q^{n \times m}$
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 μ 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

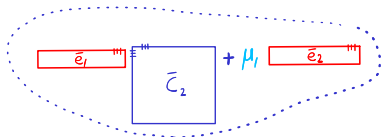
2.2 Certain inner products “preserved”

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

$m \lceil \log q \rceil$

$$\sum_{k \in [l]} a_{ik} 2^k = a_i$$

Supporting Arbitrary Depth



► Two tricks:

1. Stick to messages μ 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”

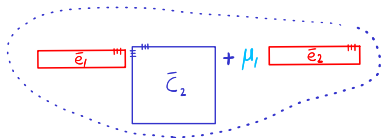
Implemented using “gadget” matrix $\bar{G} : \mathbb{Z}_q^{n \times N} \rightarrow \mathbb{Z}_q^{n \times m}$
 bit-decomposition function $G^{-1} : \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{n \times N}$

$$\sum_{k \in [l]} a_{ik} 2^k = a_{ii}$$

$$a_{i1} \dots a_{il}$$

$m \lceil \log q \rceil$

Supporting Arbitrary Depth



► Two tricks:

1. Stick to messages μ 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”

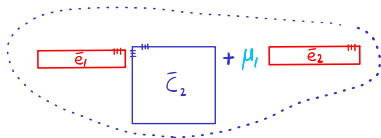
Implemented using “gadget” matrix $\bar{G} : \mathbb{Z}_q^{n \times N} \rightarrow \mathbb{Z}_q^{n \times m}$

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

$$\sum_{k \in [l]} a_{ik} 2^k = a_{i1} \quad \dots \quad a_{im} \quad \boxed{a_{i1} \dots a_{il}} \quad \dots \quad \boxed{a_{i1} \dots a_{lm}}$$

$\swarrow m \lceil \log q \rceil$

Supporting Arbitrary Depth



► Two tricks:

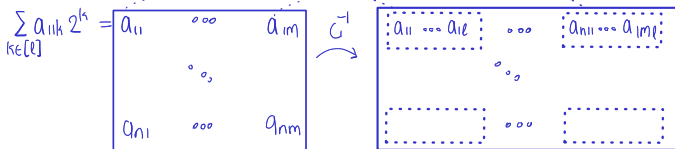
1. Stick to messages μ from $\{0, 1\}$ and F with NAND gates
2. “Flattening”: embed matrix \bar{C} into a higher dimensional matrix \bar{C}' such that

2.1 \bar{C}' has low (infinity) norm

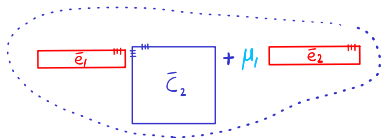
2.2 Certain inner products “preserved”

Implemented using “gadget” matrix $\bar{G} : \mathbb{Z}_q^{n \times N} \rightarrow \mathbb{Z}_q^{n \times m}$

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



Supporting Arbitrary Depth



► Two tricks:

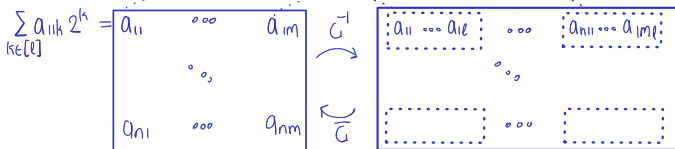
1. Stick to messages μ from $\{0, 1\}$ and F with NAND gates
2. “Flattening”: embed matrix \bar{C} into a higher dimensional matrix \bar{C}' such that

2.1 \bar{C}' has low (infinity) norm

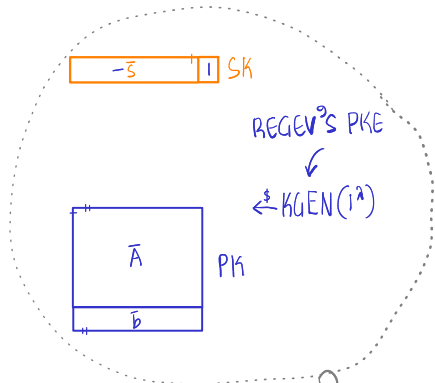
2.2 Certain inner products “preserved”

Implemented using “gadget” matrix $\bar{G} : \mathbb{Z}_q^{n \times N} \rightarrow \mathbb{Z}_q^{n \times m}$

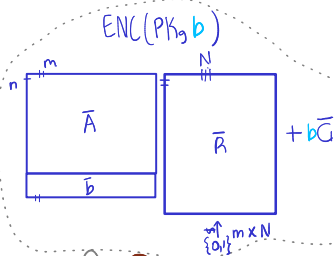
bit-decomposition function $G^{-1} : \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{n \times N}$



Putting it all Together



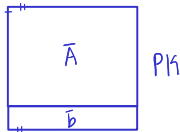
Putting it all Together



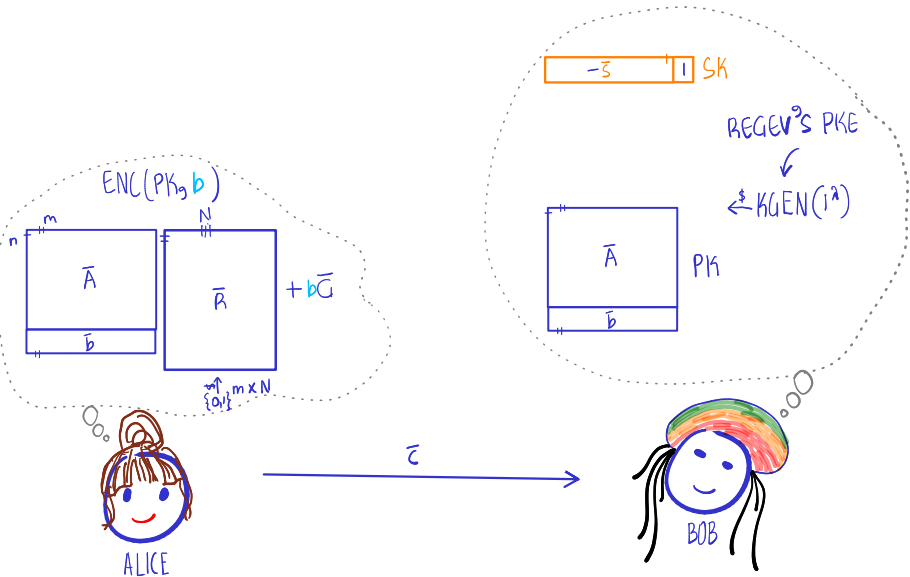
$-\bar{s} \quad || \quad SK$

RECEV'S PKE

$\leftarrow KGEN(1^\lambda)$



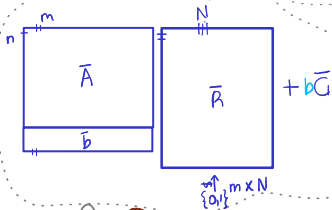
Putting it all Together



Putting it all Together

$$\text{EVAL}(\text{PK}, \bar{c}_1, \dots, \bar{c}_k, F)$$

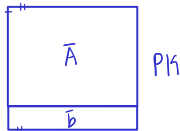
$$\text{ENC}(\text{PK}, b)$$



$$-\bar{s} \quad || \quad \text{SK}$$

RECEV'S PKE

$\leftarrow \text{KGEN}(1^\lambda)$

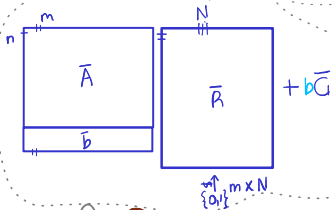


Putting it all Together

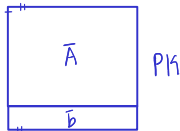
$$\text{EVAL}(\text{PK}, \bar{c}_1 \dots \bar{c}_k, F)$$

\bar{c}_1, \bar{c}_2 (with XOR gate icon)
 $\bar{c}_i := \bar{a}^i - c_i \times \bar{G}(c_i)$

$\text{ENC}(\text{PK}_g, b)$



RECEIVER'S PKE
 $\leftarrow \text{KGEN}(1^\lambda)$



Plan for this Session

Homomorphic Encryption

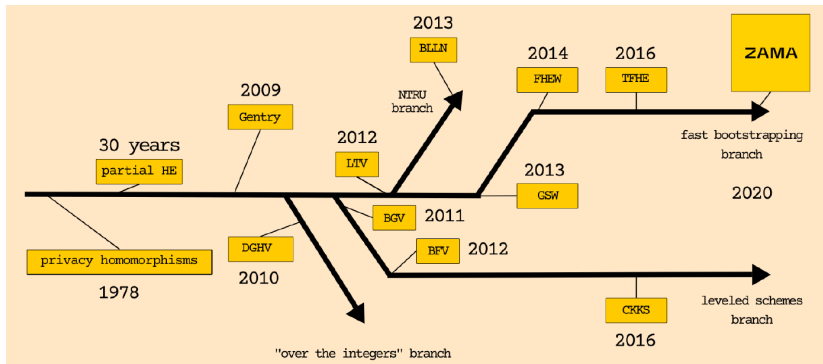
Fully-Homomorphic Encryption (FHE)

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 [EIG84, 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.