

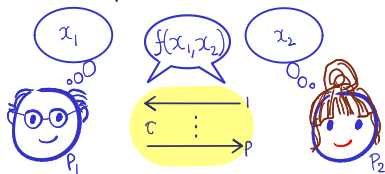
CS783: Theoretical Foundations of Cryptography

Lecture 19 (15/Oct/24)

Instructor: Chethan Kamath

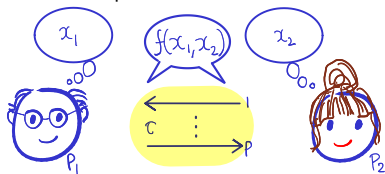
Recall from Last Two Lectures

■ Task 6: Private computation of functions



Recall from Last Two Lectures

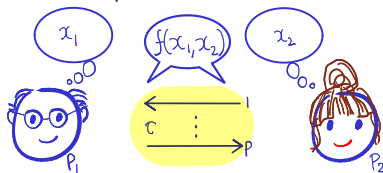
■ Task 6: Private computation of functions



- Perfectly-private MPC for *linear* functions over \mathbb{F}_p
- Perfectly-private 2PC for \wedge is impossible!
- Computationally-private 2PC for general functions over \mathbb{F}_2

Recall from Last Two Lectures

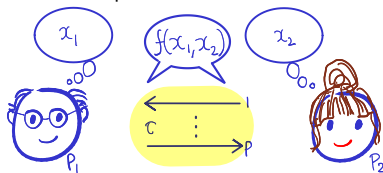
■ Task 6: Private computation of functions



- Perfectly-private MPC for *linear* functions over \mathbb{F}_p
 - Perfectly-private 2PC for \wedge is impossible!
 - Computationally-private 2PC for general functions over \mathbb{F}_2
- Key tools:
- Threshold secret sharing (TSS): privately computes \oplus/\div
 - Construction: Shamir's TSS
 - Linearity: "sum of shares \rightarrow shares of sum"
 - Oblivious transfer (OT): privately computes \wedge/\cdot
 - Trapdoor permutation (TDP) \rightarrow OT

Recall from Last Two Lectures

■ Task 6: Private computation of functions



- Perfectly-private MPC for *linear* functions over \mathbb{F}_p
 - Perfectly-private 2PC for \wedge is impossible!
 - Computationally-private 2PC for general functions over \mathbb{F}_2
- ## ■ Key tools:
- Threshold secret sharing (TSS): privately computes \oplus/\div
 - Construction: Shamir's TSS
 - Linearity: “sum of shares \rightarrow shares of sum”
 - Oblivious transfer (OT): privately computes \wedge/\cdot
 - Trapdoor permutation (TDP) \rightarrow OT
- ## ■ Key idea: “computing over secret shares”

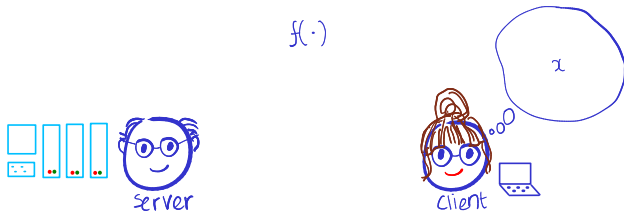
Plan for Today's Lecture...

- Task 7: secure outsourcing in the client-server setting



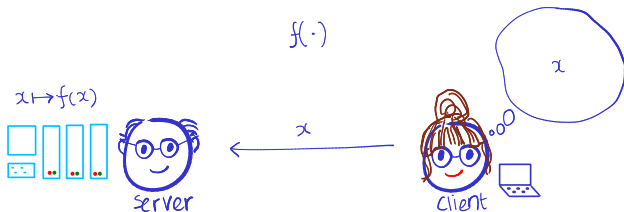
Plan for Today's Lecture...

- Task 7: secure outsourcing in the client-server setting



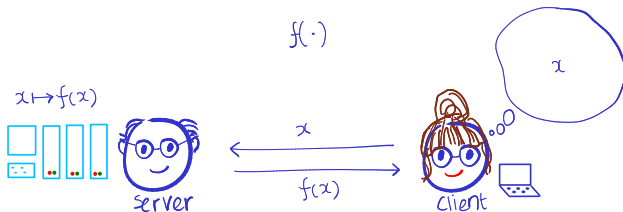
Plan for Today's Lecture...

■ Task 7: secure outsourcing in the client-server setting



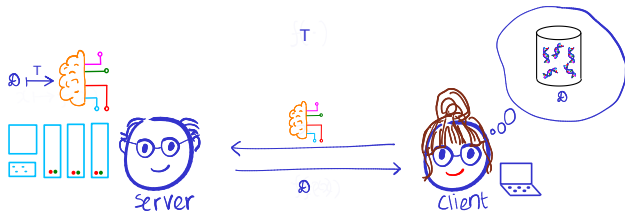
Plan for Today's Lecture...

■ Task 7: secure outsourcing in the client-server setting



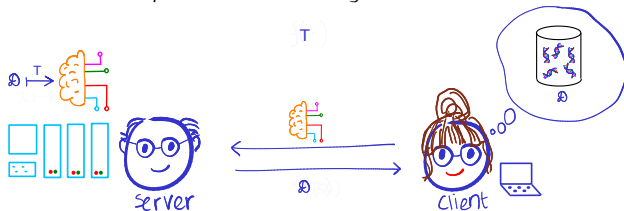
Plan for Today's Lecture...

■ Task 7: secure outsourcing in the client-server setting



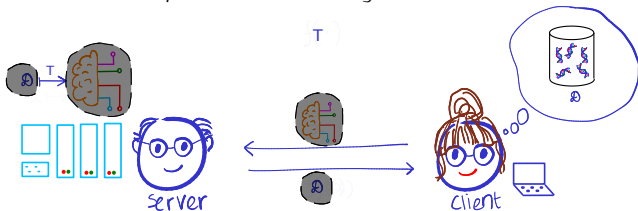
Plan for Today's Lecture...

- Task 7: secure outsourcing in the client-server setting
 - Task 7.a: *private* outsourcing in the client-server setting



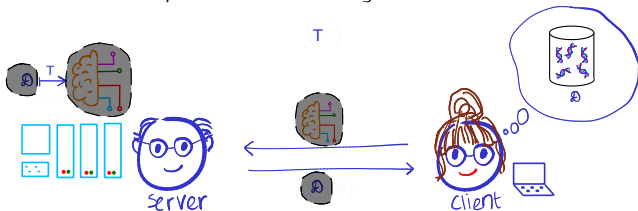
Plan for Today's Lecture...

- Task 7: secure outsourcing in the client-server setting
 - Task 7.a: *private* outsourcing in the client-server setting



Plan for Today's Lecture...

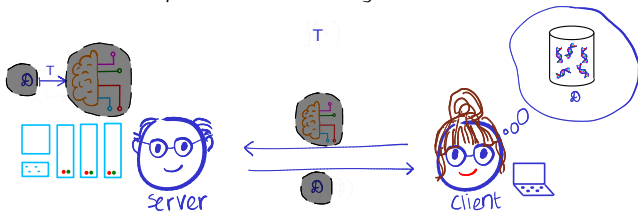
- Task 7: secure outsourcing in the client-server setting
 - Task 7.a: *private* outsourcing in the client-server setting



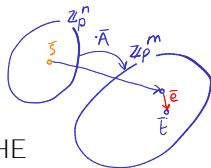
- Key tool: homomorphic (public-key) encryption
 - Operation on ciphertext \implies operation on plaintext
 - Fully homomorphic encryption (FHE)
 - Private outsourcing of computation using FHE

Plan for Today's Lecture...

- Task 7: secure outsourcing in the client-server setting
 - Task 7.a: *private* outsourcing in the client-server setting



- Key tool: homomorphic (public-key) encryption
 - Operation on ciphertext \implies operation on plaintext
 - Fully homomorphic encryption (FHE)
 - Private outsourcing of computation using FHE
- FHE from learning with errors (LWE) assumption
 - Recall LWE and Regev's encryption
 - Gentry-Sahai-Waters construction of (levelled) FHE






Plan for Today's Lecture...

General *template*:  Task 7.a: private outsourcing

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





Plan for Today's Lecture...

General *template*:

- 1 Identify the task  Task 7.a: private outsourcing
- 2 Come up with precise threat model M (a.k.a security model)
 - Adversary/Attack: What are the adversary's capabilities?  Honest-but-curious server
 - Security Goal: What does it mean to be secure?  computational privacy
- 3 Construct a scheme Π
- 4 Formally prove that Π is secure in model M

Plan for Today's Lecture...

General *template*:

- 1 Identify the task  Task 7.a: private outsourcing
- 2 Come up with precise threat model M (a.k.a security model)
 - Adversary/Attack: What are the adversary's capabilities?  Honest-but-curious server
 - Security Goal: What does it mean to be secure?  computational privacy
- 3 Construct a scheme Π FHE-based protocol
- 4 Formally prove that Π is secure in model M
 \hookrightarrow FHE IND-CPA secure $\Rightarrow \Pi$ private

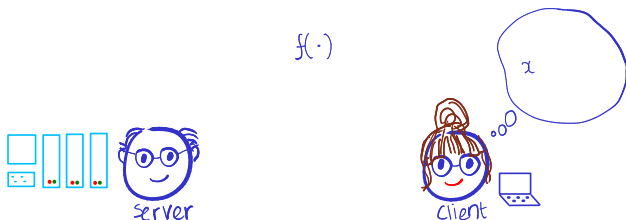
Plan for Today's Lecture

- 1 Private Outsourcing of Computation
- 2 Fully-Homomorphic Encryption (FHE)
- 3 Gentry-Sahai-Waters FHE from Learning with Errors

Plan for Today's Lecture

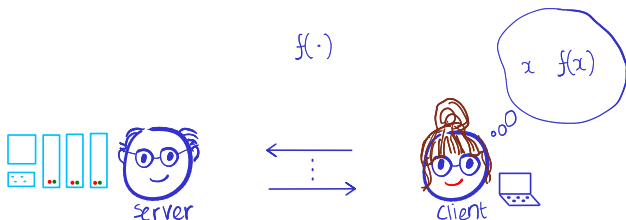
- 1 Private Outsourcing of Computation
- 2 Fully-Homomorphic Encryption (FHE)
- 3 Gentry-Sahai-Waters FHE from Learning with Errors

Private Outsourcing: Setting and Security



- The setting:
 - Client is resource constrained and server is powerful
 - Function f known to both client and server
 - Alternatively: f is known only to server (=2PC)
 - Client's local input is x

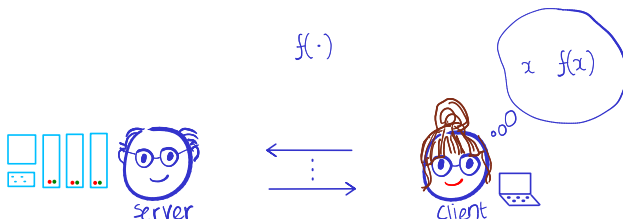
Private Outsourcing: Setting and Security



■ The setting:

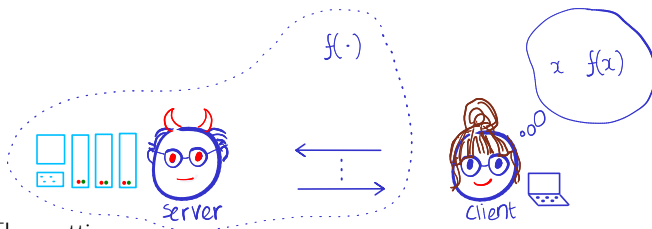
- Client is resource constrained and server is powerful
- Function f known to both client and server
 - Alternatively: f is known only to server (=2PC)
- Client's local input is x
- Server and client interact; in the end client locally outputs $f(x)$

Private Outsourcing: Setting and Security



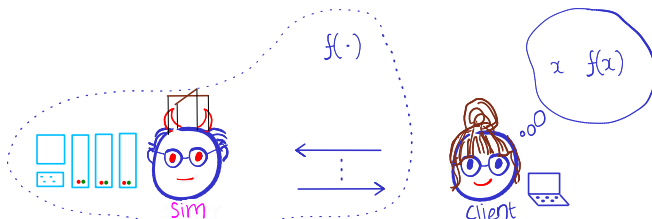
- The setting:
 - Client is resource constrained and server is powerful
 - Function f known to both client and server
 - Alternatively: f is known only to server (=2PC)
 - Client's local input is x
 - Server and client interact; in the end client locally outputs $f(x)$
- Security model: privacy against honest-but-curious server
 - Computational privacy of client's input: there exists a simulator Sim for (honest) server's view

Private Outsourcing: Setting and Security



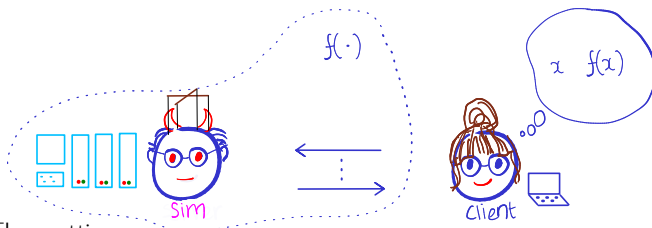
- The setting:
 - Client is resource constrained and server is powerful
 - Function f known to both client and server
 - Alternatively: f is known only to server (=2PC)
 - Client's local input is x
 - Server and client interact; in the end client locally outputs $f(x)$
- Security model: privacy against honest-but-curious server
 - Computational privacy of client's input: there exists a simulator Sim for (honest) server's view

Private Outsourcing: Setting and Security



- The setting:
 - Client is resource constrained and server is powerful
 - Function f known to both client and server
 - Alternatively: f is known only to server (=2PC)
 - Client's local input is x
 - Server and client interact; in the end client locally outputs $f(x)$
- Security model: privacy against honest-but-curious server
 - Computational privacy of client's input: there exists a simulator Sim for (honest) server's view

Private Outsourcing: Setting and Security



- The setting:
 - Client is resource constrained and server is powerful
 - Function f known to both client and server
 - Alternatively: f is known only to server (=2PC)
 - Client's local input is x
 - Server and client interact; in the end client locally outputs $f(x)$
- Security model: privacy against honest-but-curious server
 - Computational privacy of client's input: there exists a simulator Sim for (honest) server's view

Exercise 1

Why is private outsourcing trivial from a 2PC perspective?

Why is it Useful?

- Compute as a service:

Amazon SageMaker

Build, train, and deploy machine learning (ML) models for any use case with fully managed infrastructure, tools, and workflows



Function App



Create



View

Description

Function apps allow you to run event-driven code without managing infrastructure, enabling you to build and deploy applications.



Why is it Useful?

- Compute as a service:

Amazon SageMaker

Build, train, and deploy machine learning (ML) models for any use case with fully managed infrastructure, tools, and workflows



Function App



Create



View

Description

Function apps allow you to run event-driven code without managing infrastructure, enabling you to build and deploy applications.



- We want: *private/confidential* compute as a service
 - Current solutions: some form of trusted hardware (TPM/HSM)



Confidential
Computing:
Hardware-Based
Trusted Execution for
Applications and Data

Why is it Useful?

- Compute as a service:

Amazon SageMaker

Build, train, and deploy machine learning (ML) models for any use case with fully managed infrastructure, tools, and workflows



Function App



Create



View

Description

Function apps allow you to run event-driven code without managing infrastructure, enabling you to build and deploy applications.



- We want: *private/confidential* compute as a service
 - Current solutions: some form of trusted hardware (TPM/HSM)



Confidential
Computing:
Hardware-Based
Trusted Execution for
Applications and Data

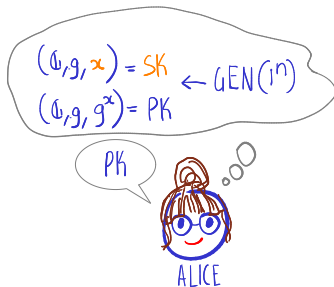
- Private outsourcing avoids trusted hardware

Plan for Today's Lecture

- 1 Private Outsourcing of Computation
- 2 Fully-Homomorphic Encryption (FHE)
- 3 Gentry-Sahai-Waters FHE from Learning with Errors

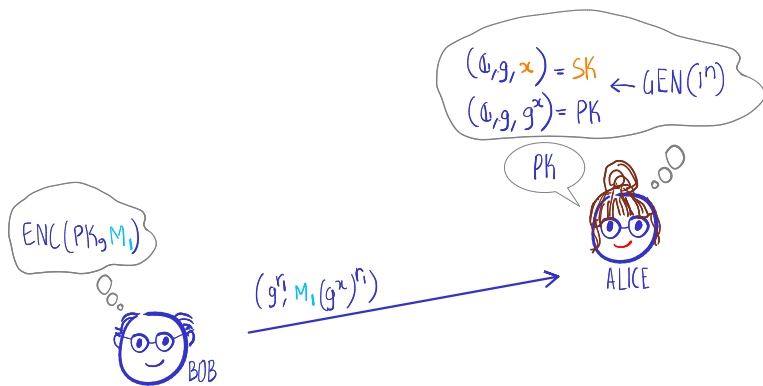
Recall the PKEs We Saw in Lecture 9...

■ PKE 1: Elgamal encryption



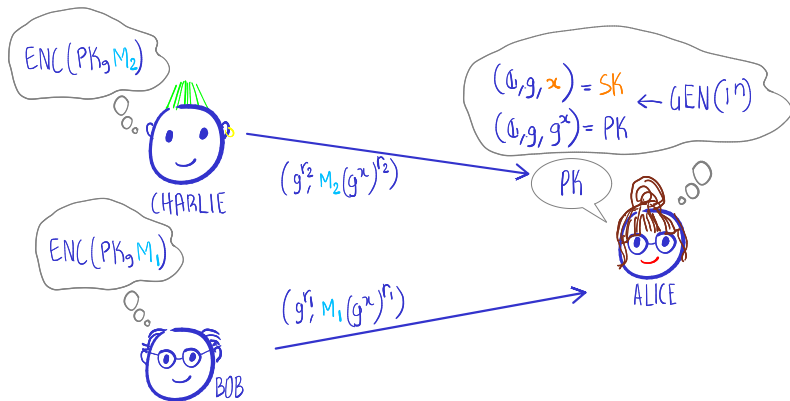
Recall the PKEs We Saw in Lecture 9...

■ PKE 1: Elgamal encryption



Recall the PKEs We Saw in Lecture 9...

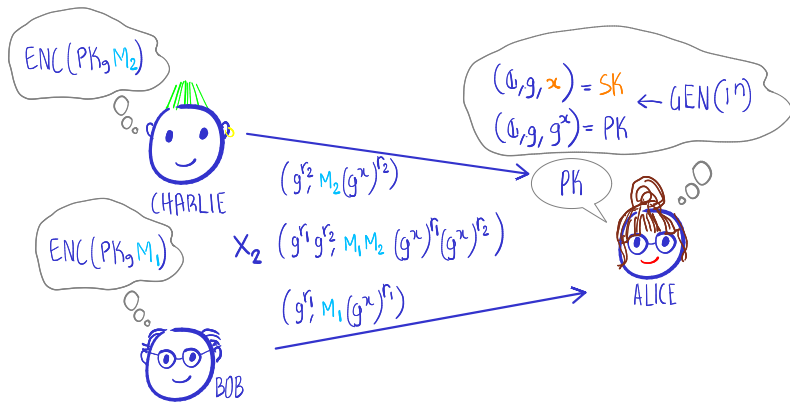
■ PKE 1: Elgamal encryption



❓ What happens when you multiply two ciphertexts?

Recall the PKEs We Saw in Lecture 9

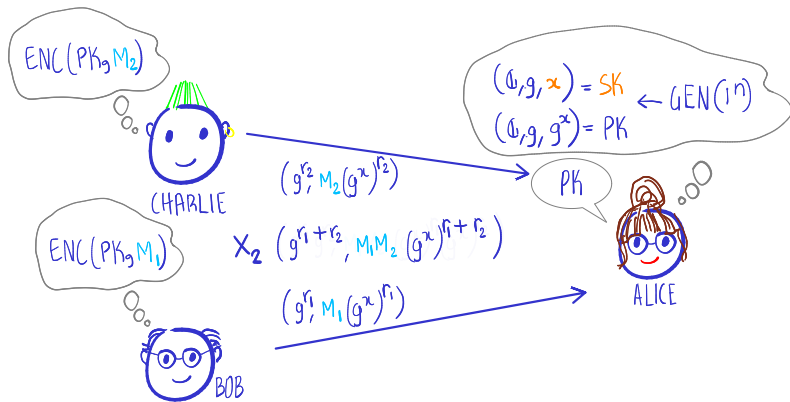
■ PKE 1: Elgamal encryption



❓ What happens when you multiply two ciphertexts?

Recall the PKEs We Saw in Lecture 9

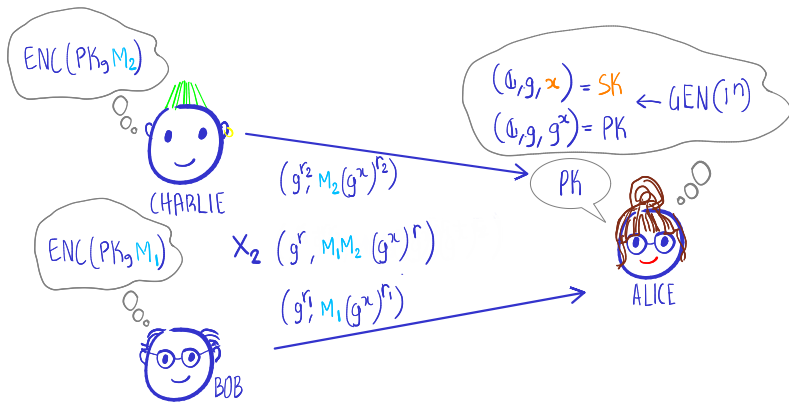
■ PKE 1: Elgamal encryption



❓ What happens when you multiply two ciphertexts?

Recall the PKEs We Saw in Lecture 9

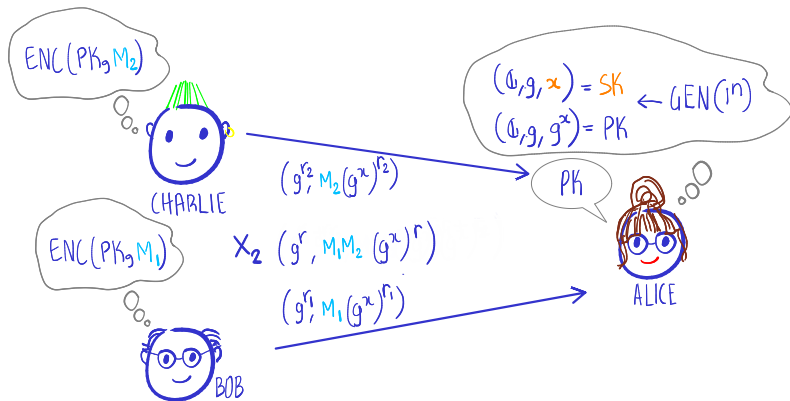
■ PKE 1: Elgamal encryption



❓ What happens when you multiply two ciphertexts?

Recall the PKEs We Saw in Lecture 9

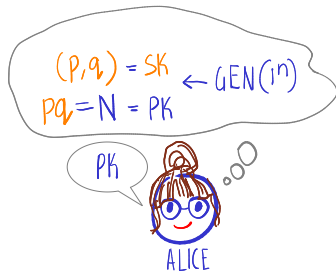
■ PKE 1: Elgamal encryption



- ② What happens when you multiply two ciphertexts?
- ② Is it possible to compute **sum** of plaintexts modulo p ?

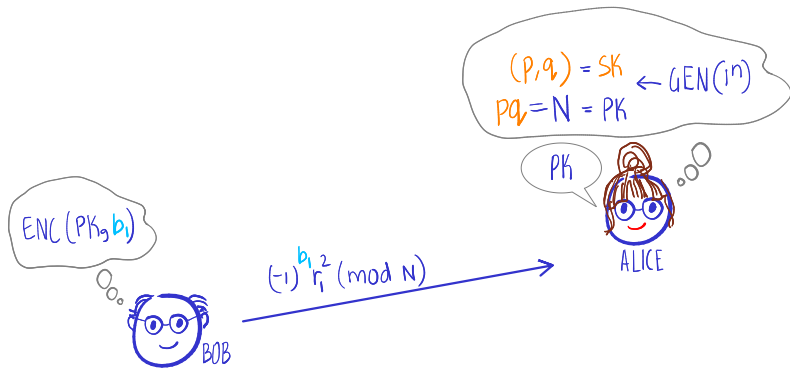
Recall the PKEs We Saw in Lecture 9...

■ PKE 2: Goldwasser-Micali bit encryption



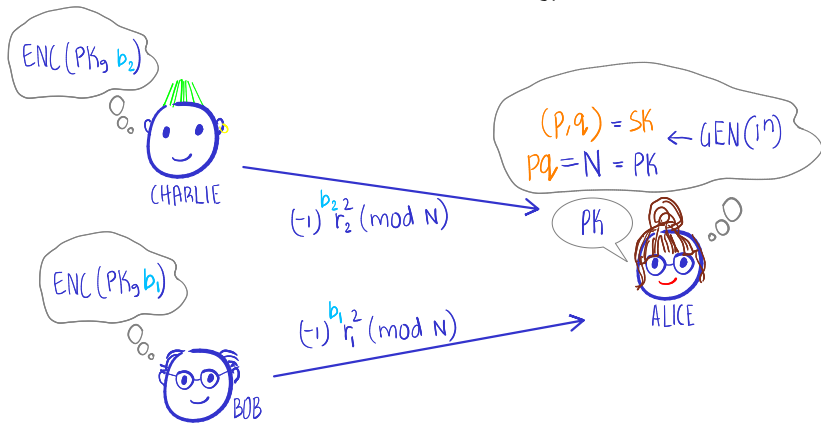
Recall the PKEs We Saw in Lecture 9...

■ PKE 2: Goldwasser-Micali bit encryption



Recall the PKEs We Saw in Lecture 9...

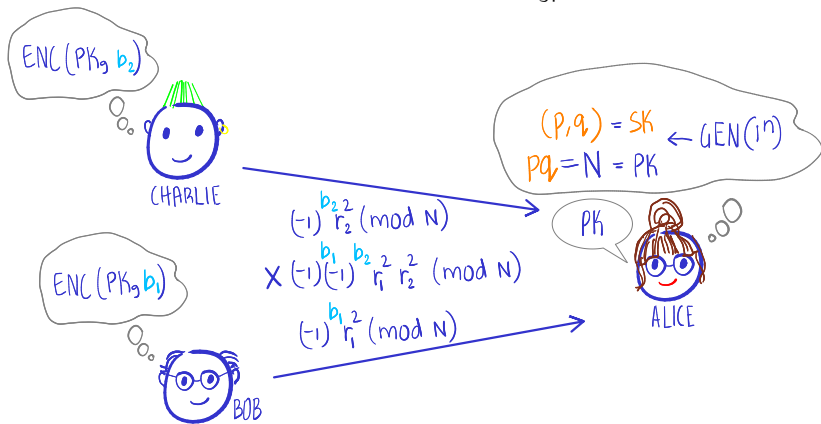
■ PKE 2: Goldwasser-Micali bit encryption



❓ What happens when you multiply ciphertexts?

Recall the PKEs We Saw in Lecture 9...

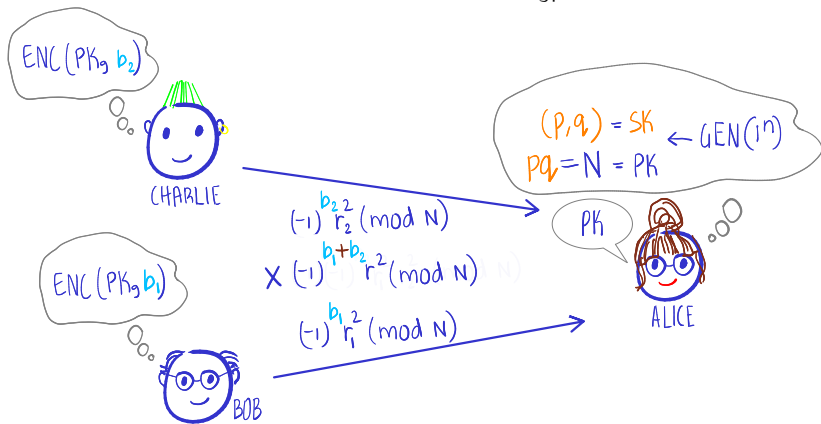
■ PKE 2: Goldwasser-Micali bit encryption



❓ What happens when you multiply ciphertexts?

Recall the PKEs We Saw in Lecture 9...

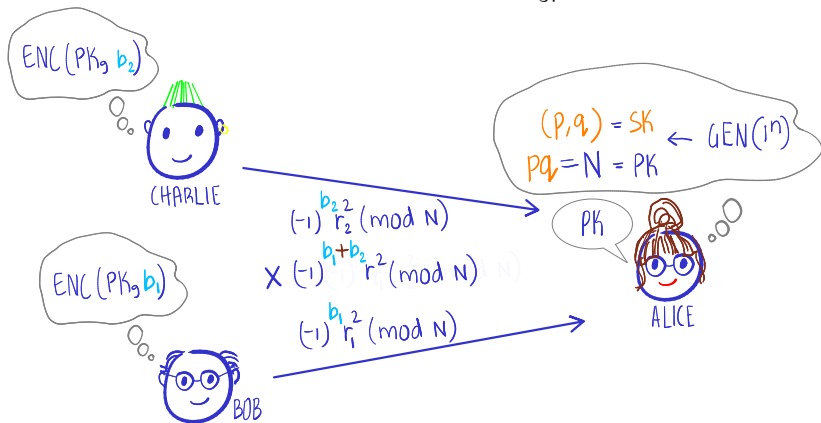
■ PKE 2: Goldwasser-Micali bit encryption



❓ What happens when you multiply ciphertexts?

Recall the PKEs We Saw in Lecture 9...

■ PKE 2: Goldwasser-Micali bit encryption



- ❓ What happens when you multiply ciphertexts?
- ❓ Is it possible to compute \wedge of plaintexts?

Let's Define Homomorphic Encryption ..

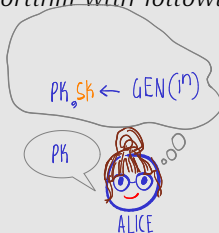
- Public-key encryption + public *evaluation* algorithm

Let's Define Homomorphic Encryption ..

- Public-key encryption + public *evaluation* algorithm

Definition 1 (Homomorphic encryption (HE) for function class \mathcal{F})

A PKE $\Sigma = (\text{Gen}, \text{Enc}, \text{Dec}) + \text{Eval}$ algorithm with following syntax

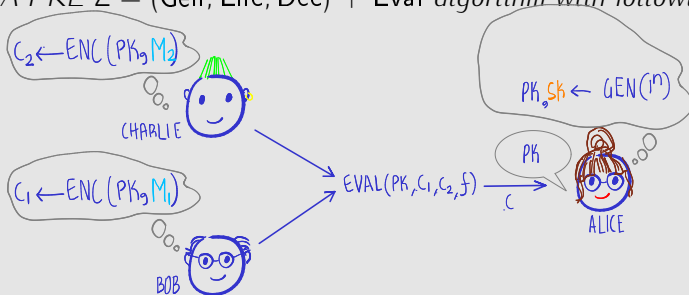


Let's Define Homomorphic Encryption ..

- Public-key encryption + public *evaluation* algorithm

Definition 1 (Homomorphic encryption (HE) for function class \mathcal{F})

A PKE $\Sigma = (\text{Gen}, \text{Enc}, \text{Dec}) + \text{Eval}$ algorithm with following syntax

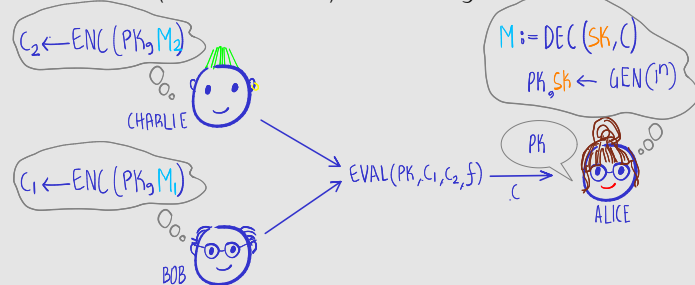


Let's Define Homomorphic Encryption ..

- Public-key encryption + public *evaluation* algorithm

Definition 1 (Homomorphic encryption (HE) for function class \mathcal{F})

A PKE $\Sigma = (\text{Gen}, \text{Enc}, \text{Dec}) + \text{Eval}$ algorithm with following syntax

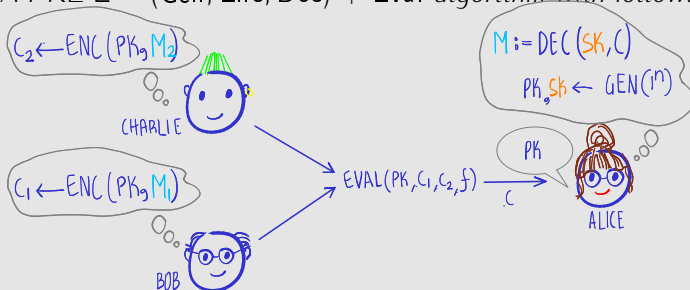


Let's Define Homomorphic Encryption ..

- Public-key encryption + public *evaluation* algorithm

Definition 1 (Homomorphic encryption (HE) for function class \mathcal{F})

A PKE $\Sigma = (\text{Gen}, \text{Enc}, \text{Dec}) + \text{Eval}$ algorithm with following syntax



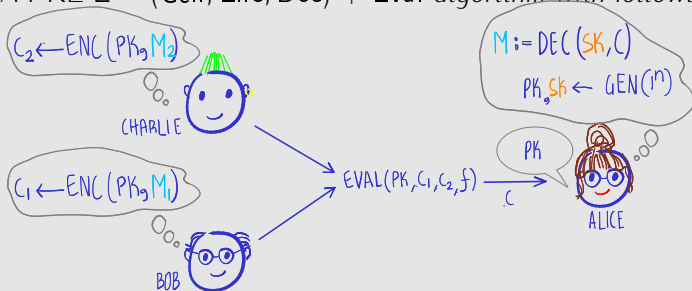
- Compactness of evaluation: $|c|$ obtained from **Eval** independent of $|f|$
- Correctness of evaluation

Let's Define Homomorphic Encryption ..

- Public-key encryption + public *evaluation* algorithm

Definition 1 (Homomorphic encryption (HE) for function class \mathcal{F})

A PKE $\Sigma = (\text{Gen}, \text{Enc}, \text{Dec}) + \text{Eval}$ algorithm with following syntax



- Compactness of evaluation: $|c|$ obtained from Eval independent of $|f|$
- Correctness of evaluation

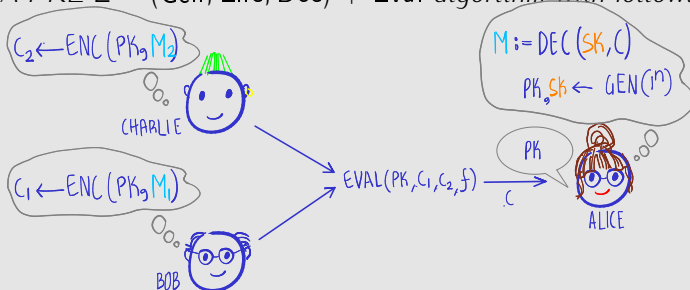
- Fully HE: \mathcal{F} = functions computable by poly.-sized circuits
 - We will represent f using a Boolean circuit of NAND gates

Let's Define Homomorphic Encryption ..

- Public-key encryption + public *evaluation* algorithm

Definition 1 (Homomorphic encryption (HE) for function class \mathcal{F})

A PKE $\Sigma = (\text{Gen}, \text{Enc}, \text{Dec}) + \text{Eval}$ algorithm with following syntax



- Compactness of evaluation: $|c|$ obtained from Eval independent of $|f|$
- Correctness of evaluation

- Fully HE: \mathcal{F} =functions computable by poly.-sized circuits
 - We will represent f using a Boolean circuit of NAND gates
- Levelled FHE: \mathcal{F} =functions computable by depth L circuits

Let's Define Homomorphic Encryption...

- Security model: same as PKE (IND-CPA)

Let's Define Homomorphic Encryption...

- Security model: same as PKE (IND-CPA)

Definition 2 (CPA Secrecy for FHE)

An FHE $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}, \text{Eval})$ is *CPA-secret* if for every PPT eavesdropper *Eve*, the following is negligible:

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

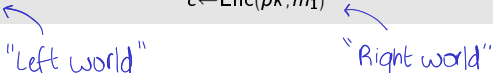
Let's Define Homomorphic Encryption...

- Security model: same as PKE (IND-CPA)

Definition 2 (CPA Secrecy for FHE)

An FHE $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}, \text{Eval})$ is *CPA-secret* if for every PPT eavesdropper *Eve*, the following is negligible:

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


"Left world" "Right world"

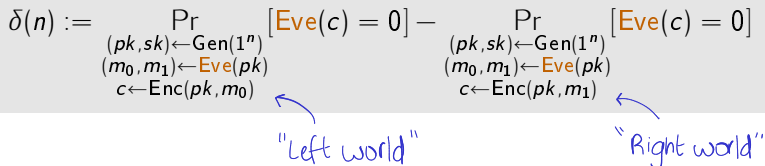
Let's Define Homomorphic Encryption...

- Security model: same as PKE (IND-CPA)

Definition 2 (CPA Secrecy for FHE)

An FHE $\Pi = (\text{Gen}, \text{Enc}, \text{Dec}, \text{Eval})$ is *CPA-secret* if for every PPT eavesdropper *Eve*, the following is negligible:

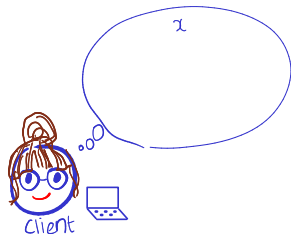
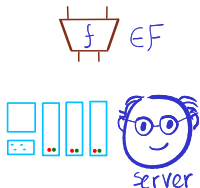
$$\delta(n) := \Pr_{\substack{(pk, sk) \leftarrow \text{Gen}(1^n) \\ (m_0, m_1) \leftarrow \text{Eve}(pk) \\ c \leftarrow \text{Enc}(pk, m_0)}}} [\text{Eve}(c) = 0] - \Pr_{\substack{(pk, sk) \leftarrow \text{Gen}(1^n) \\ (m_0, m_1) \leftarrow \text{Eve}(pk) \\ c \leftarrow \text{Enc}(pk, m_1)}}} [\text{Eve}(c)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



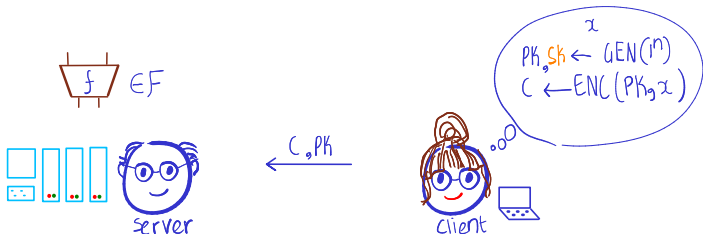
Exercise 2 (Recall: IND-CCA=IND-CPA+decryption oracle)

Can FHE be IND-CCA secure?

❓ How to Privately Outsource using FHE?



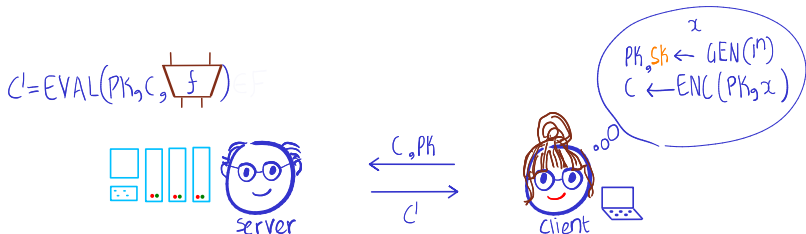
? How to Privately Outsource using FHE?



1 Client:

- 1 Generate FHE public-secret key-pair (pk, sk)
- 2 Encrypt input x using pk to get ciphertext c ; send it to server

? How to Privately Outsource using FHE?



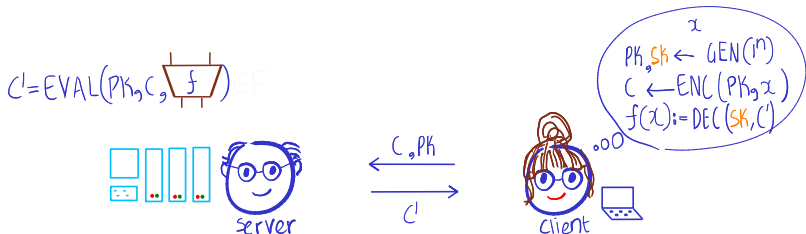
1 Client:

- 1 Generate FHE public-secret key-pair (pk, sk)
- 2 Encrypt input x using pk to get ciphertext c ; send it to server

2 Server:

- 1 Use **Eval** to run f on c and obtain encrypted output c'
- 2 Send c' to client

? How to Privately Outsource using FHE?



1 Client:

- 1 Generate FHE public-secret key-pair (pk, sk)
- 2 Encrypt input x using pk to get ciphertext c ; send it to server

2 Server:

- 1 Use **Eval** to run f on c and obtain encrypted output c'
- 2 Send c' to client

3 Client: decrypt c' using sk to retrieve output $f(x)$

Exercise 3

Prove that the above protocol is private if FHE is IND-CPA secure

Plan for this Session

- 1 Private Outsourcing of Computation
- 2 Fully-Homomorphic Encryption (FHE)
- 3 Gentry-Sahai-Waters FHE from Learning with Errors

Let's Recall LWE from Lecture 10

- Solving “noisy” linear equations over $(\mathbb{Z}_p, +, \cdot)$ is hard

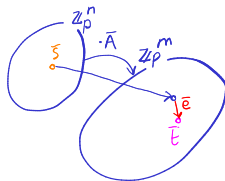
Let's Recall LWE from Lecture 10

- Solving “noisy” linear equations over $(\mathbb{Z}_p, +, \cdot)$ is hard

- Input (\bar{A}, \bar{t}) , where $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}$, $\bar{s} \leftarrow \mathbb{Z}_p^n$, $\bar{e} \leftarrow E^m$ and

$$\bar{t}^\top := \bar{s}^\top \bar{A} + \bar{e}^\top \bmod p$$

- Solution: \bar{s}

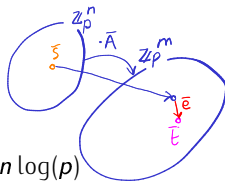


Let's Recall LWE from Lecture 10

- Solving “noisy” linear equations over $(\mathbb{Z}_p, +, \cdot)$ is hard

- Input (\bar{A}, \bar{t}) , where $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}$, $\bar{s} \leftarrow \mathbb{Z}_p^n$, $\bar{e} \leftarrow E^m$ and

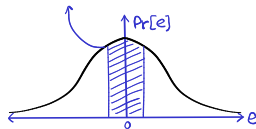
$$\bar{t}^\top := \bar{s}^\top \bar{A} + \bar{e}^\top \bmod p$$



- Solution: \bar{s}
- Usual parameters:

- n =security parameter, $p = \text{poly}(n)$ and $m \approx n \log(p)$
 - Noise distribution $E = E_\alpha$, the *discrete Gaussian distribution* over \mathbb{Z}
 - Centred at 0; parameter $\alpha < 1$ determines s.d. $\sigma := \alpha p \approx n$

$$\Pr[e] = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(-\frac{e^2}{2\sigma^2}\right)$$

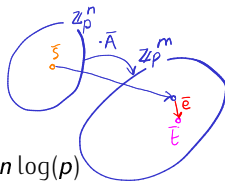


Let's Recall LWE from Lecture 10

- Solving “noisy” linear equations over $(\mathbb{Z}_p, +, \cdot)$ is hard

- Input (\bar{A}, \bar{t}) , where $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}$, $\bar{s} \leftarrow \mathbb{Z}_p^n$, $\bar{e} \leftarrow E^m$ and

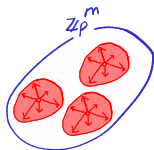
$$\bar{t}^\top := \bar{s}^\top \bar{A} + \bar{e}^\top \bmod p$$



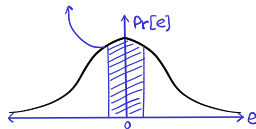
- Solution: \bar{s}
- Usual parameters:

- n =security parameter, $p = \text{poly}(n)$ and $m \approx n \log(p)$
 - Noise distribution $E = E_\alpha$, the *discrete Gaussian distribution* over \mathbb{Z}

- Centred at 0; parameter $\alpha < 1$ determines s.d. $\sigma := \alpha p \approx n$



$$\Pr[e] = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(-\frac{e^2}{2\sigma^2}\right)$$



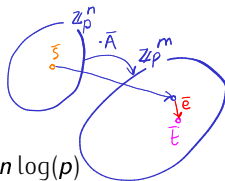
- \bar{t} “determines” \bar{s} , but efficient algorithm to recover \bar{s} not known

Let's Recall LWE from Lecture 10

- Solving “noisy” linear equations over $(\mathbb{Z}_p, +, \cdot)$ is hard

- Input (\bar{A}, \bar{t}) , where $\bar{A} \leftarrow \mathbb{Z}_p^{n \times m}$, $\bar{s} \leftarrow \mathbb{Z}_p^n$, $\bar{e} \leftarrow E^m$ and

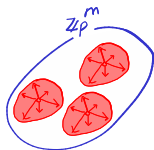
$$\bar{t}^\top := \bar{s}^\top \bar{A} + \bar{e}^\top \bmod p$$



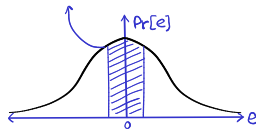
- Solution: \bar{s}
- Usual parameters:

- n =security parameter, $p = \text{poly}(n)$ and $m \approx n \log(p)$
 - Noise distribution $E = E_\alpha$, the *discrete Gaussian distribution* over \mathbb{Z}

- Centred at 0; parameter $\alpha < 1$ determines s.d. $\sigma := \alpha p \approx n$



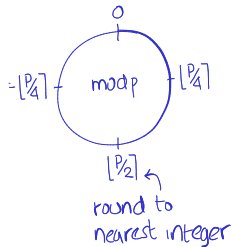
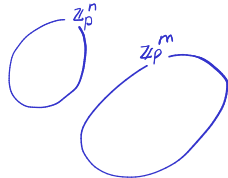
$$\Pr[e] = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(-\frac{e^2}{2\sigma^2}\right)$$



- \bar{t} “determines” \bar{s} , but efficient algorithm to recover \bar{s} not known
- Decision LWE (DLWE): $(\bar{A}, \bar{t}) \approx (\bar{A}, \bar{r})$, where $\bar{r} \leftarrow \mathbb{Z}_p^m$

Let's Recall Regev's PKE from Lecture 10...

- "Noisy/approximate" 1-bit key-exchange based on DLWE:



Let's Recall Regev's PKE from Lecture 10

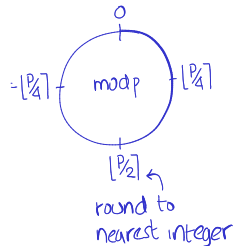
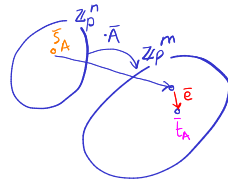
- "Noisy/approximate" 1-bit key-exchange based on DLWE:



← LWE instance

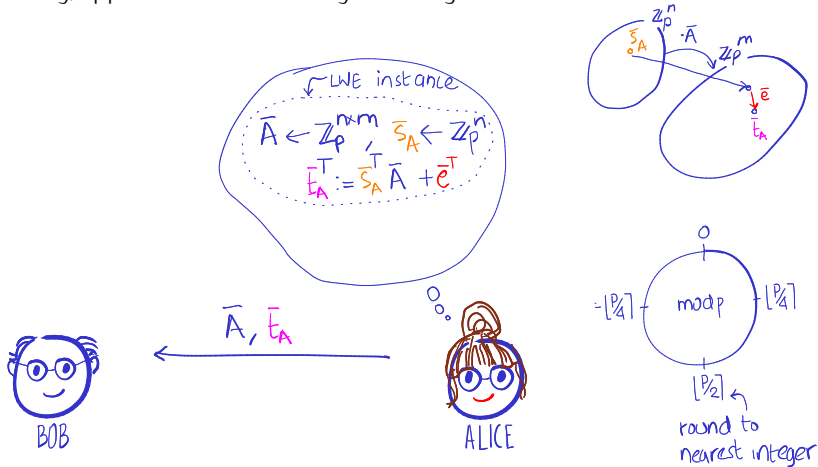
$$\bar{A} \leftarrow \mathbb{Z}_p^{nm}, \bar{s}_A \leftarrow \mathbb{Z}_p^n$$

$$\bar{t}_A^T := \bar{s}_A^T \bar{A} + \bar{e}^T$$



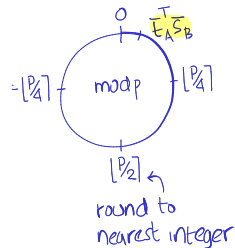
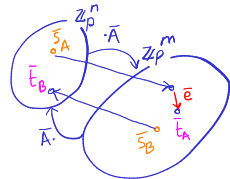
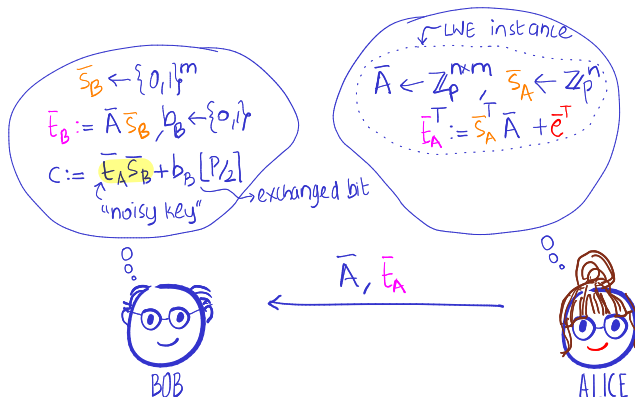
Let's Recall Regev's PKE from Lecture 10

- "Noisy/approximate" 1-bit key-exchange based on DLWE:



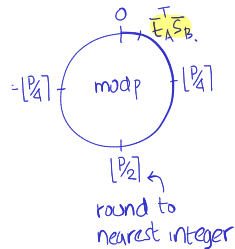
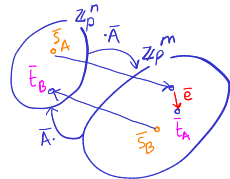
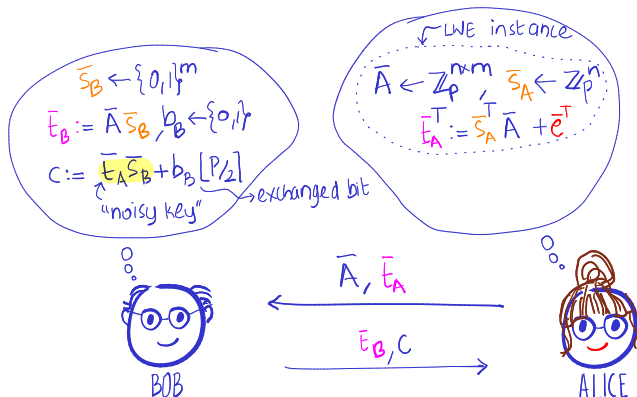
Let's Recall Regev's PKE from Lecture 10

- "Noisy/approximate" 1-bit key-exchange based on DLWE:



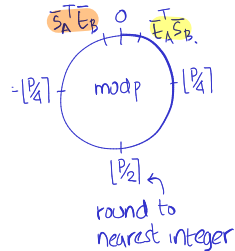
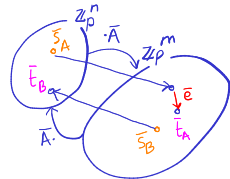
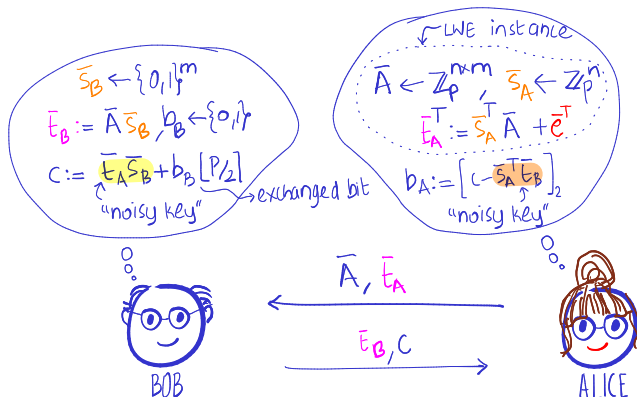
Let's Recall Regev's PKE from Lecture 10

- "Noisy/approximate" 1-bit key-exchange based on DLWE:



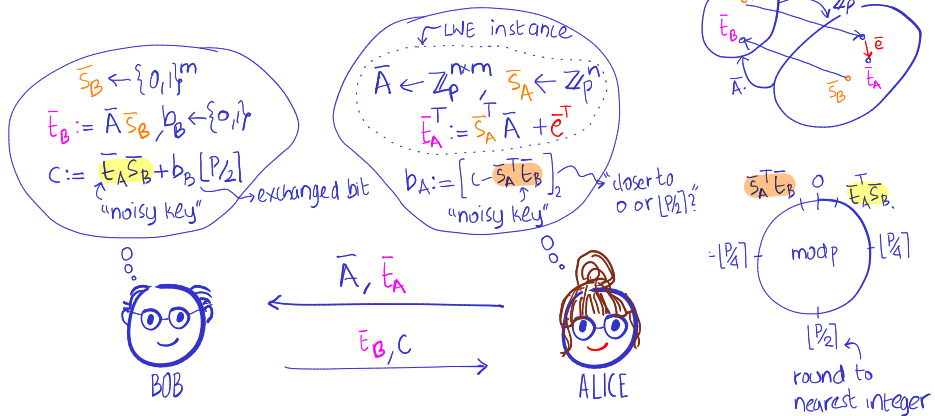
Let's Recall Regev's PKE from Lecture 10

- "Noisy/approximate" 1-bit key-exchange based on DLWE:



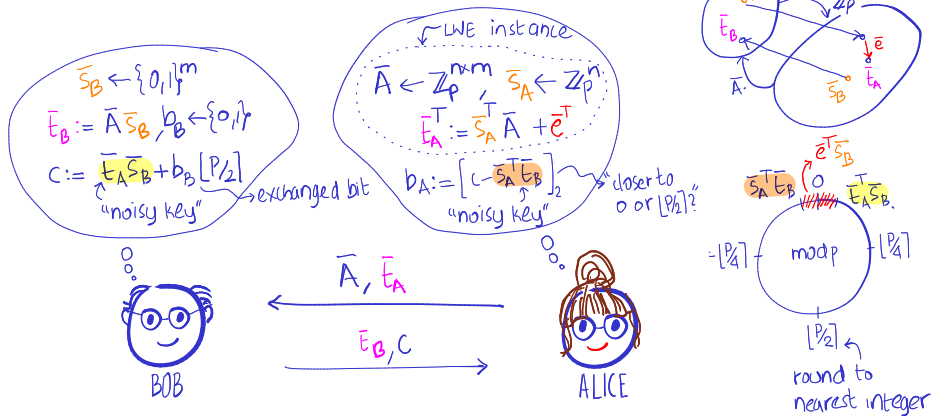
Let's Recall Regev's PKE from Lecture 10

- "Noisy/approximate" 1-bit key-exchange based on DLWE:



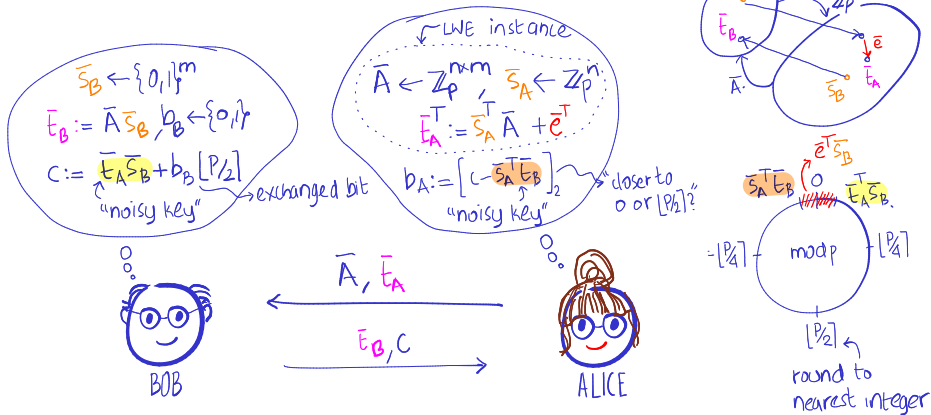
Let's Recall Regev's PKE from Lecture 10

- "Noisy/approximate" 1-bit key-exchange based on DLWE:



Let's Recall Regev's PKE from Lecture 10

- "Noisy/approximate" 1-bit key-exchange based on DLWE:



- Regev's PKE obtained by generic transformation

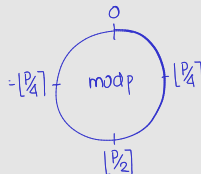
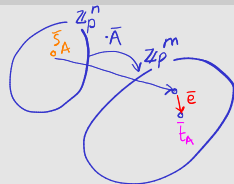
Let's Recall Regev's PKE from Lecture 10...

Construction 1 (Regev's PKE for parameters n, m, p and E_α)

■ Key generation $\text{Gen}(1^n; \bar{A}, \bar{s}_A, \bar{e})$:

$$pk := \left(\bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top \bmod p \right) \quad sk := \bar{s}_A$$

← LWE instance



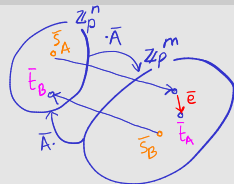
Let's Recall Regev's PKE from Lecture 10...

Construction 1 (Regev's PKE for parameters n, m, p and E_α)

- Key generation $\text{Gen}(1^n; \bar{A}, \bar{s}_A, \bar{e})$:

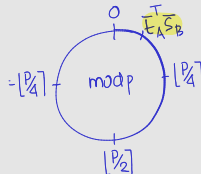
$$pk := \left(\bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top \bmod p \right) \quad sk := \bar{s}_A$$

← LWE instance



- Encryption $\text{Enc}(pk, b; \bar{s}_B)$:

$$\bar{c} := pk \bar{s}_B + \begin{pmatrix} 0^n \\ b \cdot \lfloor p/2 \rfloor \end{pmatrix} = \begin{pmatrix} \bar{t}_B := \bar{A} \bar{s}_B \\ \bar{t}_A^\top \bar{s}_B + b \cdot \lfloor p/2 \rfloor \end{pmatrix} \bmod p$$



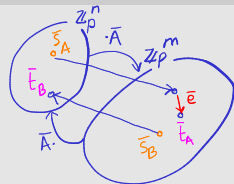
Let's Recall Regev's PKE from Lecture 10...

Construction 1 (Regev's PKE for parameters n, m, p and E_α)

- *Key generation* $\text{Gen}(1^n; \bar{A}, \bar{s}_A, \bar{e})$:

$$pk := \left(\bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top \bmod p \right) \quad sk := \bar{s}_A$$

← LWE instance

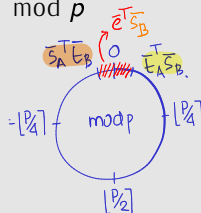


- *Encryption* $\text{Enc}(pk, b; \bar{s}_B)$:

$$\bar{c} := pk \bar{s}_B + \begin{pmatrix} 0^n \\ b \cdot \lfloor p/2 \rfloor \end{pmatrix} = \begin{pmatrix} \bar{t}_B := \bar{A} \bar{s}_B \\ \bar{t}_A^\top \bar{s}_B + b \cdot \lfloor p/2 \rfloor \end{pmatrix} \bmod p$$

- *Decryption* $\text{Dec}(sk, \bar{c})$:

$$b' := [(-\bar{s}_A^\top, 1) \bar{c}] = \bar{e}^\top \bar{s}_B + b \cdot \lfloor p/2 \rfloor \bmod p_2$$



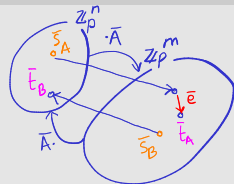
Let's Recall Regev's PKE from Lecture 10...

Construction 1 (Regev's PKE for parameters n, m, p and E_α)

- Key generation $\text{Gen}(1^n; \bar{A}, \bar{s}_A, \bar{e})$:

$$pk := \left(\bar{t}_A^\top := \bar{s}_A^\top \bar{A} + \bar{e}^\top \bmod p \right) \quad sk := \bar{s}_A$$

← WE instance

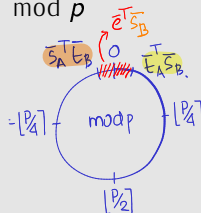


- Encryption $\text{Enc}(pk, b; \bar{s}_B)$:

$$\bar{c} := pk \bar{s}_B + \begin{pmatrix} 0^n \\ b \cdot \lfloor p/2 \rfloor \end{pmatrix} = \begin{pmatrix} \bar{t}_B := \bar{A} \bar{s}_B \\ \bar{t}_A^\top \bar{s}_B + b \cdot \lfloor p/2 \rfloor \end{pmatrix} \bmod p$$

- Decryption $\text{Dec}(sk, \bar{c})$:

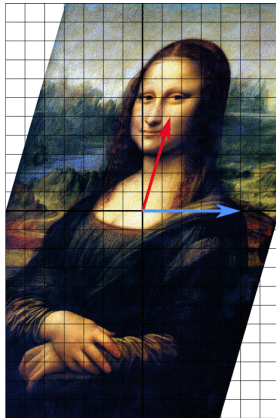
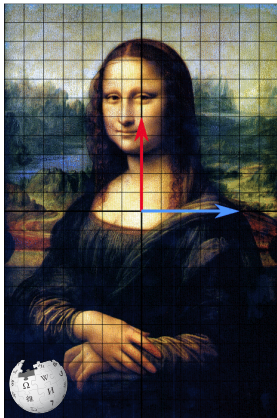
$$b' := [(-\bar{s}_A^\top, 1) \bar{c}] = \bar{e}^\top \bar{s}_B + b \cdot \lfloor p/2 \rfloor \bmod p_2$$



❓ What happens when you add two ciphertexts?

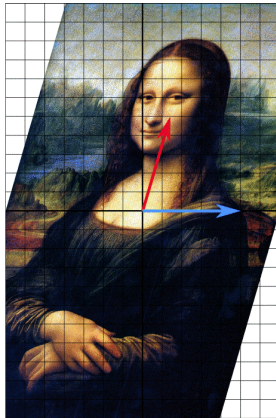
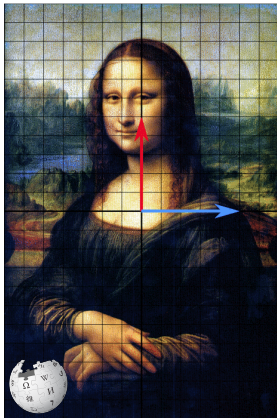
First Attempt: “Eigenvector” Encryption...

- Let's recall eigenvectors



First Attempt: “Eigenvector” Encryption...

- Let's recall eigenvectors



Definition 1 (Eigenvectors for matrices over \mathbb{F}_p)

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**.

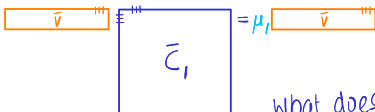
First Attempt: "Eigenvector" Encryption...

- Invariant: $n \times n$ "ciphertext" matrix \bar{C} encrypts bit μ under secret \bar{v} if $\bar{v} \bar{C} = \mu \bar{v}$

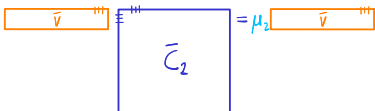
The diagram illustrates the invariant equation $\bar{v} \bar{C} = \mu \bar{v}$. It features three main components: an orange rectangle on the left containing the symbol \bar{v} , a blue square in the center containing the symbol \bar{C} , and another orange rectangle on the right containing the symbol \bar{v} . The first orange rectangle is connected to the blue square by a blue line with three small vertical tick marks. The blue square is connected to the second orange rectangle by a blue line with a single small vertical tick mark. An equals sign is placed between the blue square and the second orange rectangle, with a blue μ positioned just above the equals sign.

First Attempt: "Eigenvector" Encryption...

- Invariant: $n \times n$ "ciphertext" matrix \bar{C} encrypts bit μ under secret \bar{v} if $\bar{v} \bar{C} = \mu \bar{v}$


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

what does $\bar{C}_1 + \bar{C}_2$
correspond to?


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

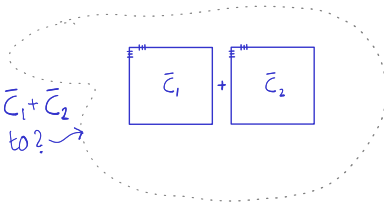
First Attempt: "Eigenvector" Encryption...

- Invariant: $n \times n$ "ciphertext" matrix \bar{C} encrypts bit μ under secret \bar{v} if $\bar{v} \bar{C} = \mu \bar{v}$

A diagram illustrating the encryption invariant for the first ciphertext matrix. On the left, an orange rectangle labeled \bar{v} is connected by a double line to a blue square labeled \bar{C}_1 . To the right of the square is an equals sign followed by a blue μ_1 and another orange rectangle labeled \bar{v} . The orange rectangles have three small vertical lines on their right side, and the blue square has three small horizontal lines on its top side.

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

what does $\bar{C}_1 + \bar{C}_2$
correspond to? →

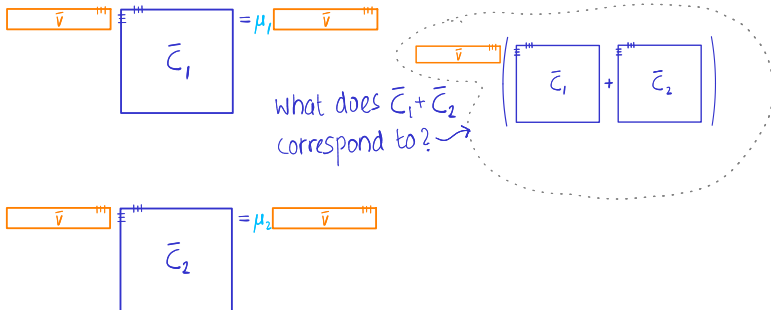


A diagram illustrating the encryption invariant for the second ciphertext matrix. On the left, an orange rectangle labeled \bar{v} is connected by a double line to a blue square labeled \bar{C}_2 . To the right of the square is an equals sign followed by a blue μ_2 and another orange rectangle labeled \bar{v} . The orange rectangles have three small vertical lines on their right side, and the blue square has three small horizontal lines on its top side.

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

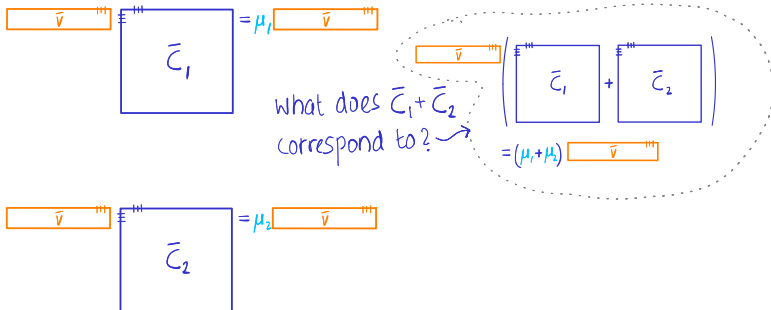
First Attempt: "Eigenvector" Encryption...

- Invariant: $n \times n$ "ciphertext" matrix \bar{C} encrypts bit μ under secret \bar{v} if $\bar{v} \bar{C} = \mu \bar{v}$



First Attempt: "Eigenvector" Encryption...

- Invariant: $n \times n$ "ciphertext" matrix \bar{C} encrypts bit μ under secret \bar{v} if $\bar{v} \bar{C} = \mu \bar{v}$



First Attempt: "Eigenvector" Encryption...

- Invariant: $n \times n$ "ciphertext" matrix \bar{C} encrypts bit μ under secret \bar{v} if $\bar{v} \bar{C} = \mu \bar{v}$

Diagram illustrating the encryption of bit μ_1 using matrix \bar{C}_1 and secret vector \bar{v} . The equation is $\bar{v} \bar{C}_1 = \mu_1 \bar{v}$. The vector \bar{v} is represented by an orange box with a double line on the left and a triple line on the right. The matrix \bar{C}_1 is a blue square. The result $\mu_1 \bar{v}$ is an orange box with a blue μ_1 to its left.

what does $\bar{C}_1 + \bar{C}_2$ correspond to? →

Diagram illustrating the addition of two ciphertext matrices \bar{C}_1 and \bar{C}_2 . The equation is $\bar{v} (\bar{C}_1 + \bar{C}_2) = (\mu_1 + \mu_2) \bar{v}$. The matrices \bar{C}_1 and \bar{C}_2 are blue squares. The result is an orange box with a blue $(\mu_1 + \mu_2)$ to its left. The entire expression is enclosed in a dotted oval.

Diagram illustrating the encryption of bit μ_2 using matrix \bar{C}_2 and secret vector \bar{v} . The equation is $\bar{v} \bar{C}_2 = \mu_2 \bar{v}$. The vector \bar{v} is represented by an orange box with a double line on the left and a triple line on the right. The matrix \bar{C}_2 is a blue square. The result $\mu_2 \bar{v}$ is an orange box with a blue μ_2 to its left.

what does $\bar{C}_1 \cdot \bar{C}_2$ correspond to?

Diagram illustrating the multiplication of two ciphertext matrices \bar{C}_1 and \bar{C}_2 . The matrices \bar{C}_1 and \bar{C}_2 are blue squares. The entire expression is enclosed in a dotted oval.

First Attempt: "Eigenvector" Encryption...

- Invariant: $n \times n$ "ciphertext" matrix \bar{C} encrypts bit μ under secret \bar{v} if $\bar{v} \bar{C} = \mu \bar{v}$

Diagram illustrating the encryption of bit μ_1 using matrix \bar{C}_1 and secret vector \bar{v} . The equation is $\bar{v} \bar{C}_1 = \mu_1 \bar{v}$. The vector \bar{v} is represented by an orange box with a double line on the left and a triple line on the right. The matrix \bar{C}_1 is a blue square. The result $\mu_1 \bar{v}$ is an orange box with a blue μ_1 to its left.

what does $\bar{C}_1 + \bar{C}_2$ correspond to?

Diagram illustrating the encryption of bit $\mu_1 + \mu_2$ using the sum of matrices \bar{C}_1 and \bar{C}_2 . The equation is $\bar{v} (\bar{C}_1 + \bar{C}_2) = (\mu_1 + \mu_2) \bar{v}$. The vector \bar{v} is an orange box. The sum of matrices $\bar{C}_1 + \bar{C}_2$ is shown in blue boxes. The result $(\mu_1 + \mu_2) \bar{v}$ is an orange box with a blue $(\mu_1 + \mu_2)$ to its left.

Diagram illustrating the encryption of bit μ_2 using matrix \bar{C}_2 and secret vector \bar{v} . The equation is $\bar{v} \bar{C}_2 = \mu_2 \bar{v}$. The vector \bar{v} is an orange box. The matrix \bar{C}_2 is a blue square. The result $\mu_2 \bar{v}$ is an orange box with a blue μ_2 to its left.

what does $\bar{C}_1 \cdot \bar{C}_2$ correspond to?

Diagram illustrating the encryption of bit $\mu_1 \cdot \mu_2$ using the product of matrices \bar{C}_1 and \bar{C}_2 . The equation is $\bar{v} (\bar{C}_1 \cdot \bar{C}_2) = \mu_1 \mu_2 \bar{v}$. The vector \bar{v} is an orange box. The product of matrices $\bar{C}_1 \cdot \bar{C}_2$ is shown in blue boxes. The result $\mu_1 \mu_2 \bar{v}$ is an orange box with blue $\mu_1 \mu_2$ to its left.

First Attempt: "Eigenvector" Encryption...

- Invariant: $n \times n$ "ciphertext" matrix \bar{C} encrypts bit μ under secret \bar{v} if $\bar{v} \bar{C} = \mu \bar{v}$

Diagram illustrating the encryption of bit μ_1 using matrix \bar{C}_1 and secret vector \bar{v} . The equation is $\bar{v} \bar{C}_1 = \mu_1 \bar{v}$. The vector \bar{v} is represented by an orange box with a double line on the left and a triple line on the right. The matrix \bar{C}_1 is a blue square. The result $\mu_1 \bar{v}$ is an orange box with a double line on the left and a triple line on the right.

what does $\bar{C}_1 + \bar{C}_2$ correspond to?

Diagram illustrating the encryption of bit $\mu_1 + \mu_2$ using the sum of matrices \bar{C}_1 and \bar{C}_2 . The equation is $\bar{v} (\bar{C}_1 + \bar{C}_2) = (\mu_1 + \mu_2) \bar{v}$. The vector \bar{v} is represented by an orange box with a double line on the left and a triple line on the right. The matrices \bar{C}_1 and \bar{C}_2 are blue squares. The result $(\mu_1 + \mu_2) \bar{v}$ is an orange box with a double line on the left and a triple line on the right.

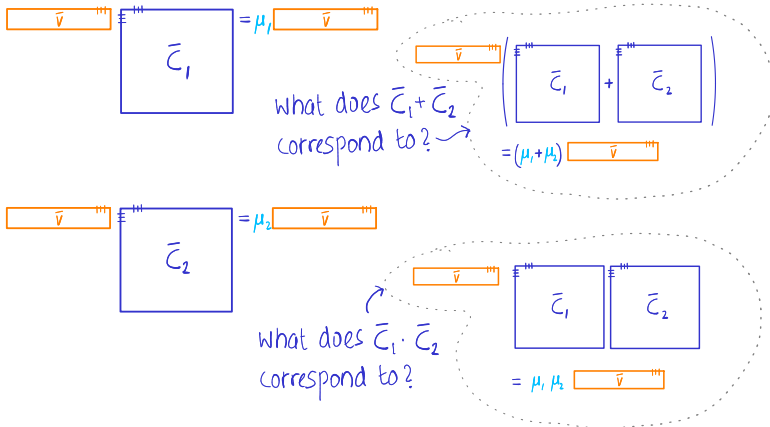
Diagram illustrating the encryption of bit μ_2 using matrix \bar{C}_2 and secret vector \bar{v} . The equation is $\bar{v} \bar{C}_2 = \mu_2 \bar{v}$. The vector \bar{v} is represented by an orange box with a double line on the left and a triple line on the right. The matrix \bar{C}_2 is a blue square. The result $\mu_2 \bar{v}$ is an orange box with a double line on the left and a triple line on the right.

what does $\bar{C}_1 \cdot \bar{C}_2$ correspond to?

Diagram illustrating the encryption of bit $\mu_1 \cdot \mu_2$ using the product of matrices \bar{C}_1 and \bar{C}_2 . The equation is $\bar{v} \bar{C}_1 \bar{C}_2 = \mu_1 \mu_2 \bar{v}$. The vector \bar{v} is represented by an orange box with a double line on the left and a triple line on the right. The matrices \bar{C}_1 and \bar{C}_2 are blue squares. The result $\mu_1 \mu_2 \bar{v}$ is an orange box with a double line on the left and a triple line on the right.

First Attempt: "Eigenvector" Encryption...

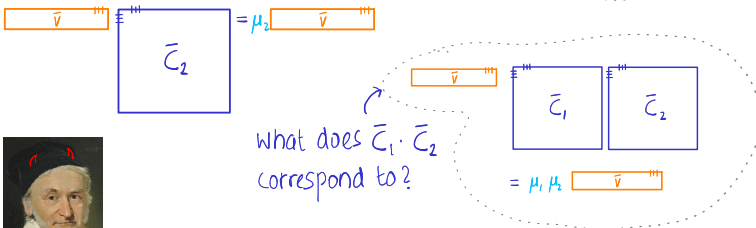
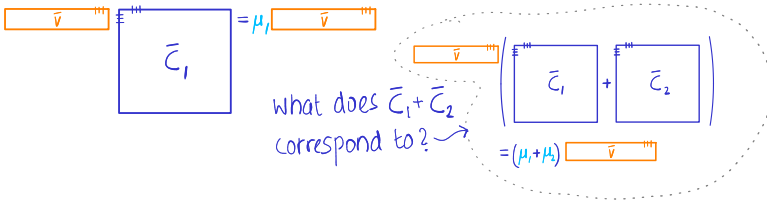
- Invariant: $n \times n$ "ciphertext" matrix \bar{C} encrypts bit μ under secret \bar{v} if $\bar{v} \bar{C} = \mu \bar{v}$



❓ Do we have an FHE?

First Attempt: "Eigenvector" Encryption...

- Invariant: $n \times n$ "ciphertext" matrix \bar{C} encrypts bit μ under secret \bar{v} if $\bar{v} \bar{C} = \mu \bar{v}$



❓ Do we have an FHE? No, can break by Gaussian elimination

Second Attempt: *Approximate* Eigenvector Encryption

- New invariant: \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for “short” \bar{e} (as in LWE)

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

Second Attempt: *Approximate* Eigenvector Encryption

- New invariant: \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for “short” \bar{e} (as in LWE)

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

Second Attempt: *Approximate* Eigenvector Encryption

- New invariant: \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for “short” \bar{e} (as in LWE)

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

what does $\bar{C}_1 + \bar{C}_2$
correspond to? ↘

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

Second Attempt: *Approximate* Eigenvector Encryption

- New invariant: \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" \bar{e} (as in LWE)

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

what does $\bar{C}_1 + \bar{C}_2$
correspond to? \searrow

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

Second Attempt: *Approximate* Eigenvector Encryption

- New invariant: \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for "short" \bar{e} (as in LWE)

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

what does $\bar{C}_1 + \bar{C}_2$
correspond to? ↘

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

Second Attempt: *Approximate* Eigenvector Encryption

- New invariant: \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for “short” \bar{e} (as in LWE)

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

what does $\bar{C}_1 + \bar{C}_2$
correspond to? ↘

$$\boxed{\bar{v}} \left(\boxed{\bar{C}_1} + \boxed{\bar{C}_2} \right) + \left(\begin{array}{c} \boxed{\bar{e}_1} \\ + \\ \boxed{\bar{e}_2} \end{array} \right) = (\mu_1 + \mu_2) \boxed{\bar{v}} + \boxed{\bar{e}_3}$$

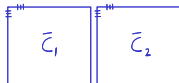
Second Attempt: *Approximate* Eigenvector Encryption

- New invariant: \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for “short” \bar{e} (as in LWE)

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

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

what does $\bar{C}_1 \cdot \bar{C}_2$
correspond to?



what does $\bar{C}_1 + \bar{C}_2$
correspond to?

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

Second Attempt: *Approximate* Eigenvector Encryption

- New invariant: \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for “short” \bar{e} (as in LWE)

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

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

what does $\bar{C}_1 \cdot \bar{C}_2$
correspond to?

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

what does $\bar{C}_1 + \bar{C}_2$
correspond to?

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

Second Attempt: *Approximate* Eigenvector Encryption

- New invariant: \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for “short” \bar{e} (as in LWE)

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

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

what does $\bar{C}_1 \cdot \bar{C}_2$ correspond to?

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

what does $\bar{C}_1 + \bar{C}_2$ correspond to?

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

- Do we have an FHE?

Second Attempt: *Approximate* Eigenvector Encryption

- New invariant: \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}$ for “short” \bar{e} (as in LWE)

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

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

what does $\bar{C}_1 \cdot \bar{C}_2$ correspond to?

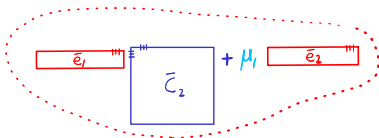
$$\boxed{\bar{v}} \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \end{matrix} \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \end{matrix} \boxed{\bar{C}_1} \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \end{matrix} \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \end{matrix}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

what does $\bar{C}_1 + \bar{C}_2$ correspond to?

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

- Do we have an FHE? No, overall error when multiplying depends \bar{C}_2 which can be large (even if \bar{e}_1 is small)

Gentry-Sahai-Waters FHE...

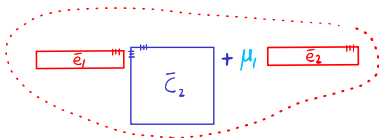


Solution: use $\bar{C}_1 \cdot G^{-1}(\bar{C}_2)$ for homomorphic multiplication

■ $G^{-1} : \mathbb{Z}_p^{n \times n} \rightarrow \mathbb{Z}_p^{n \times n \ell}$ is the **bit-decomposition** function
 $\downarrow \log(q)$

- 1 $G^{-1}(\bar{C}_2)$ has small entries (i.e., **low** infinity norm)
- 2 Linear algebra we carried out before still holds

Gentry-Sahai-Waters FHE...



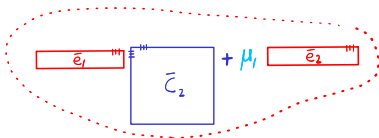
Solution: use $\bar{C}_1 \cdot G^{-1}(\bar{C}_2)$ for homomorphic multiplication

■ $G^{-1} : \mathbb{Z}_p^{n \times n} \rightarrow \mathbb{Z}_p^{n \times n \ell}$ is the **bit-decomposition** function

- 1 $G^{-1}(\bar{C}_2)$ has small entries (i.e., **low** infinity norm)
- 2 Linear algebra we carried out before still holds

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

Gentry-Sahai-Waters FHE...



Solution: use $\bar{C}_1 \cdot G^{-1}(\bar{C}_2)$ for homomorphic multiplication

■ $G^{-1} : \mathbb{Z}_p^{n \times n} \rightarrow \mathbb{Z}_p^{n \times n \ell}$ is the **bit-decomposition** function

- 1 $G^{-1}(\bar{C}_2)$ has small entries (i.e., **low** infinity norm)
- 2 Linear algebra we carried out before still holds

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

$$a_{i11} \dots a_{i1\ell}$$

Gentry-Sahai-Waters FHE...

$$\boxed{\bar{e}_1} + \mu \boxed{\bar{e}_2} \quad \text{and} \quad \boxed{\bar{c}_2}$$



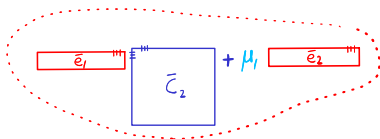
Solution: use $\bar{C}_1 \cdot G^{-1}(\bar{C}_2)$ for homomorphic multiplication

■ $G^{-1} : \mathbb{Z}_p^{n \times n} \rightarrow \mathbb{Z}_p^{n \times n \ell}$ is the **bit-decomposition** function

- 1 $G^{-1}(\bar{C}_2)$ has small entries (i.e., **low** infinity norm)
- 2 Linear algebra we carried out before still holds

$$\sum_{k \in [L]} a_{ik} 2^k = \begin{bmatrix} a_{i1} & \dots & a_{in} \\ & \ddots & \\ a_{n1} & \dots & a_{nm} \end{bmatrix} \xrightarrow{G^{-1}} \begin{bmatrix} a_{i11} \dots a_{i1\ell} & \dots & a_{i n1} \dots a_{i n \ell} \\ & \ddots & \\ & & \end{bmatrix}$$

Gentry-Sahai-Waters FHE...



Solution: use $\bar{C}_1 \cdot G^{-1}(\bar{C}_2)$ for homomorphic multiplication

- $G^{-1} : \mathbb{Z}_p^{n \times n} \rightarrow \mathbb{Z}_p^{n \times n\ell}$ is the **bit-decomposition** function

- 1 $G^{-1}(\bar{C}_2)$ has small entries (i.e., **low** infinity norm)
- 2 Linear algebra we carried out before still holds

$$\sum_{k \in [k]} a_{ik} 2^k = \begin{bmatrix} a_{i1} & \dots & a_{in} \\ & \ddots & \\ a_{n1} & \dots & a_{nm} \end{bmatrix} \xrightarrow{G^{-1}} \begin{bmatrix} \boxed{a_{i11} \dots a_{i1\ell}} & \dots & \boxed{a_{i1n} \dots a_{i1\ell}} \\ & \ddots & \\ \boxed{a_{n11} \dots a_{n1\ell}} & \dots & \boxed{a_{n1n} \dots a_{n1\ell}} \end{bmatrix}$$

- G^{-1} 's inverse computed using **gadget** matrix $\bar{G} : \mathbb{Z}_p^{n \times n\ell} \rightarrow \mathbb{Z}_p^{n \times n}$
 - $\forall \bar{v} : \bar{G} G^{-1}(\bar{v}) = \bar{v}$

Gentry-Sahai-Waters FHE...

- New invariant: $n \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} \stackrel{*}{=} \mu\bar{v}\bar{G}$ for "short" \bar{e}

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

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

Gentry-Sahai-Waters FHE...

- New invariant: $n \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v} \bar{C} + \bar{e} \stackrel{*}{=} \mu \bar{v} \bar{G}$ for "short" \bar{e}

What does $\bar{C}_1 + \bar{C}_2$
correspond to?

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

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

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

Gentry-Sahai-Waters FHE...

- New invariant: $n \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v} \bar{C} + \bar{e} \stackrel{*}{=} \mu \bar{v} \bar{G}$ for "short" \bar{e}

what does $\bar{C}_1 + \bar{C}_2$
correspond to?

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

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

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

what does $\bar{C}_1 \cdot \bar{G}(\bar{C}_2)$
correspond to?

Gentry-Sahai-Waters FHE...

- New invariant: $n \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v} \bar{C} + \bar{e} \stackrel{*}{=} \mu \bar{v} \bar{G}$ for "short" \bar{e}

what does $\bar{C}_1 + \bar{C}_2$ correspond to? $\nabla(\bar{C}_1 + \bar{C}_2) + (\bar{e}_1 + \bar{e}_2) = (\mu_1 + \mu_2) \bar{v} \bar{G}$

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

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

what does $\bar{C}_1 \cdot G^{-1}(\bar{C}_2)$ correspond to? $\nabla \cdot \bar{C}_1 \cdot G^{-1}(\bar{C}_2) \stackrel{*}{=} (\mu_1 \bar{v} \bar{G} - \bar{e}_1) \cdot G^{-1}(\bar{C}_2)$

Gentry-Sahai-Waters FHE...

- New invariant: $n \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v} \bar{C} + \bar{e} \stackrel{*}{=} \mu \bar{v} \bar{G}$ for "short" \bar{e}

what does $\bar{C}_1 + \bar{C}_2$
correspond to?

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

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

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

what does $\bar{C}_1 \cdot G^{-1}(\bar{C}_2)$
correspond to?

$$\begin{aligned} \nabla \cdot \bar{C}_1 \cdot G^{-1}(\bar{C}_2) &\stackrel{*}{=} (\mu_1 \bar{v} \bar{G} - \bar{e}_1) \cdot G^{-1}(\bar{C}_2) \\ &= \mu_1 \bar{v} \bar{G} \cdot G^{-1}(\bar{C}_2) - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \end{aligned}$$

Gentry-Sahai-Waters FHE...

- New invariant: $n \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v} \bar{C} + \bar{e} \stackrel{*}{=} \mu \bar{v} \bar{G}$ for "short" \bar{e}

what does $\bar{C}_1 + \bar{C}_2 \rightsquigarrow \nabla(\bar{C}_1 + \bar{C}_2) + (\bar{e}_1 + \bar{e}_2) = (\mu_1 + \mu_2) \bar{v} \bar{G}$ correspond to?

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

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

what does $\bar{C}_1 \cdot G^{-1}(\bar{C}_2) \rightsquigarrow \nabla \cdot \bar{C}_1 \cdot G^{-1}(\bar{C}_2) \stackrel{*}{=} (\mu_1 \bar{v} \bar{G} - \bar{e}_1) \cdot G^{-1}(\bar{C}_2)$ correspond to?

$$\begin{aligned} &= \mu_1 \bar{v} \bar{C}_1 \cdot G^{-1}(\bar{C}_2) - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \\ &= \mu_1 \bar{v} \bar{C}_2 - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \end{aligned}$$

Gentry-Sahai-Waters FHE...

- New invariant: $n \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v} \bar{C} + \bar{e} \stackrel{*}{=} \mu \bar{v} \bar{G}$ for "short" \bar{e}

what does $\bar{C}_1 + \bar{C}_2 \rightsquigarrow \nabla(\bar{C}_1 + \bar{C}_2) + (\bar{e}_1 + \bar{e}_2) = (\mu_1 + \mu_2) \bar{v} \bar{G}$ correspond to?

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

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

what does $\bar{C}_1 \cdot G^{-1}(\bar{C}_2) \rightsquigarrow \nabla \cdot \bar{C}_1 \cdot G^{-1}(\bar{C}_2) \stackrel{*}{=} (\mu_1 \bar{v} \bar{G} - \bar{e}_1) \cdot G^{-1}(\bar{C}_2)$ correspond to?

$$\begin{aligned} &= \mu_1 \bar{v} \bar{G} \cdot G^{-1}(\bar{C}_2) - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \\ &= \mu_1 \bar{v} \bar{C}_2 - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \\ &\stackrel{*}{=} \mu_1 (\mu_2 \bar{v} \bar{G} - \bar{e}_2) - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \end{aligned}$$

Gentry-Sahai-Waters FHE...

- New invariant: $n \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v} \bar{C} + \bar{e} \stackrel{*}{=} \mu \bar{v} \bar{G}$ for "short" \bar{e}

what does $\bar{C}_1 + \bar{C}_2 \rightsquigarrow \nabla(\bar{C}_1 + \bar{C}_2) + (\bar{e}_1 + \bar{e}_2) = (\mu_1 + \mu_2) \bar{v} \bar{G}$ correspond to?

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

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

what does $\bar{C}_1 \cdot G^{-1}(\bar{C}_2) \rightsquigarrow \nabla \cdot \bar{C}_1 \cdot G^{-1}(\bar{C}_2) \stackrel{*}{=} (\mu_1 \bar{v} \bar{G} - \bar{e}_1) \cdot G^{-1}(\bar{C}_2)$ correspond to?

$$\begin{aligned}
 &= \mu_1 \bar{v} \bar{G} \cdot G^{-1}(\bar{C}_2) - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \\
 &= \mu_1 \bar{v} \bar{C}_2 - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \\
 &\stackrel{*}{=} \mu_1 (\mu_2 \bar{v} \bar{G} - \bar{e}_2) - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \\
 &= \mu_1 \mu_2 \bar{v} \bar{G} - \mu_1 \bar{e}_2 - \bar{e}_1 \cdot G^{-1}(\bar{C}_2)
 \end{aligned}$$

Gentry-Sahai-Waters FHE...

- New invariant: $n \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v} \bar{C} + \bar{e} \stackrel{*}{=} \mu \bar{v} \bar{G}$ for "short" \bar{e}

what does $\bar{C}_1 + \bar{C}_2 \rightsquigarrow \nabla(\bar{C}_1 + \bar{C}_2) + (\bar{e}_1 + \bar{e}_2) = (\mu_1 + \mu_2) \bar{v} \bar{G}$ correspond to?

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

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

what does $\bar{C}_1 \cdot G^{-1}(\bar{C}_2) \rightsquigarrow \nabla \cdot \bar{C}_1 \cdot G^{-1}(\bar{C}_2) \stackrel{*}{=} (\mu_1 \bar{v} \bar{G} - \bar{e}_1) \cdot G^{-1}(\bar{C}_2)$ correspond to?

$$\begin{aligned}
 &= \mu_1 \bar{v} \bar{G} \cdot G^{-1}(\bar{C}_2) - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \\
 &= \mu_1 \bar{v} \bar{C}_2 - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \\
 &\stackrel{*}{=} \mu_1 (\mu_2 \bar{v} \bar{G} - \bar{e}_2) - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \\
 &= \mu_1 \mu_2 \bar{v} \bar{G} - \mu_1 \bar{e}_2 - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \leftarrow \text{short!}
 \end{aligned}$$

Gentry-Sahai-Waters FHE...

- New invariant: $n \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v} \bar{C} + \bar{e} \stackrel{*}{=} \mu \bar{v} \bar{G}$ for "short" \bar{e}

what does $\bar{C}_1 + \bar{C}_2 \rightsquigarrow \nabla(\bar{C}_1 + \bar{C}_2) + (\bar{e}_1 + \bar{e}_2) = (\mu_1 + \mu_2) \bar{v} \bar{G}$ correspond to?

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

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

what does $\bar{C}_1 \cdot G^{-1}(\bar{C}_2) \rightsquigarrow \nabla \cdot \bar{C}_1 \cdot G^{-1}(\bar{C}_2) \stackrel{*}{=} (\mu_1 \bar{v} \bar{G} - \bar{e}_1) \cdot G^{-1}(\bar{C}_2)$ correspond to?

$$\begin{aligned}
 &= \mu_1 \bar{v} \bar{G} \cdot G^{-1}(\bar{C}_2) - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \\
 &= \mu_1 \bar{v} \bar{C}_2 - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \\
 &\stackrel{*}{=} \mu_1 (\mu_2 \bar{v} \bar{G} - \bar{e}_2) - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \\
 &= \mu_1 \mu_2 \bar{v} \bar{G} - \mu_1 \bar{e}_2 - \bar{e}_1 \cdot G^{-1}(\bar{C}_2) \leftarrow \text{short!}
 \end{aligned}$$

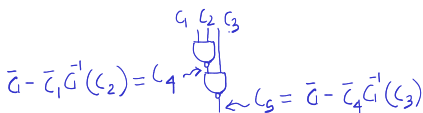
- How does the noise grow? Entries of μ_1 , \bar{e}_1 , \bar{e}_2 and \bar{C}_2 are at most $B \implies$ new noise at most $N \cdot B^2$

Putting it All Together

- Invariant: $n \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}\bar{G}$ for “short” \bar{e}
 - Secret key of the form $\bar{v} \in \mathbb{Z}_p^n$
 - Ciphertexts of the form $\bar{C} \in \mathbb{Z}_p^{n \times N}$

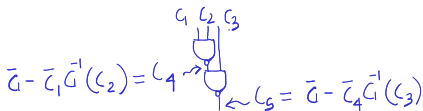
Putting it All Together

- Invariant: $n \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}\bar{G}$ for “short” \bar{e}
 - Secret key of the form $\bar{v} \in \mathbb{Z}_p^n$
 - Ciphertexts of the form $\bar{C} \in \mathbb{Z}_p^{n \times N}$
- To evaluate a NAND circuit $f : \{0, 1\}^\lambda \rightarrow \{0, 1\}$ on ciphertexts $(\bar{C}_1, \dots, \bar{C}_\lambda)$:
 - 1 Consider each gate G in f in topological order
 - 2 Let \bar{C}_i and \bar{C}_j denote ciphertexts corresponding to its inputs
 - 3 Output $\bar{C}_k := \bar{G} - \bar{C}_1 G^{-1}(\bar{C}_2)$ as its output ciphertext



Putting it All Together

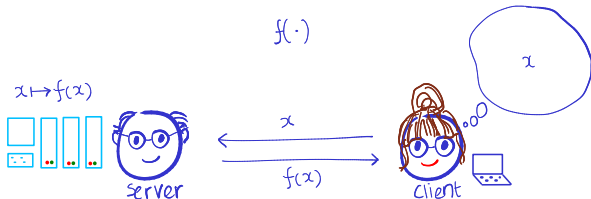
- Invariant: $n \times N$ matrix \bar{C} encrypts a bit μ under secret \bar{v} if $\bar{v}\bar{C} + \bar{e} = \mu\bar{v}\bar{G}$ for “short” \bar{e}
 - Secret key of the form $\bar{v} \in \mathbb{Z}_p^n$
 - Ciphertexts of the form $\bar{C} \in \mathbb{Z}_p^{n \times N}$
- To evaluate a NAND circuit $f : \{0, 1\}^\lambda \rightarrow \{0, 1\}$ on ciphertexts $(\bar{C}_1, \dots, \bar{C}_\lambda)$:
 - 1 Consider each gate G in f in topological order
 - 2 Let \bar{C}_i and \bar{C}_j denote ciphertexts corresponding to its inputs
 - 3 Output $\bar{C}_k := \bar{G} - \bar{C}_1 G^{-1}(\bar{C}_2)$ as its output ciphertext



- If the depth is d then the noise in ciphertext of output wire is $B(N + 1)^d$
 - \Rightarrow modulus $q \gg B(N + 1)^d$

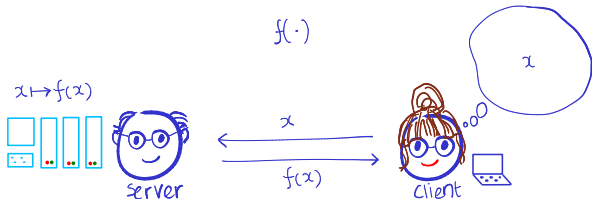
To Recap Today's Lecture

- Task 7: secure outsourcing in the client-server setting
 - Task 7.a: *private* outsourcing in the client-server setting



To Recap Today's Lecture

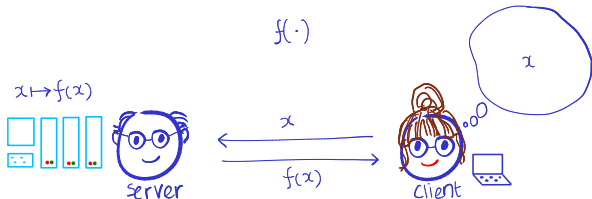
- Task 7: secure outsourcing in the client-server setting
 - Task 7.a: *private* outsourcing in the client-server setting



- Key tool: Fully homomorphic encryption (FHE)
 - FHE \rightarrow private outsourcing of computation
 - Possible: FHE \rightarrow 2PC of arbitrary functions!

To Recap Today's Lecture

- Task 7: secure outsourcing in the client-server setting
 - Task 7.a: *private* outsourcing in the client-server setting



- Key tool: Fully homomorphic encryption (FHE)
 - FHE \rightarrow private outsourcing of computation
 - Possible: FHE \rightarrow 2PC of arbitrary functions!
- GSW FHE from LWE assumption
 - Key idea: approximate eigenvectors
 - Similar idea used in approximate key exchange from LWE

Next Lecture

- Task 7: secure outsourcing in the client-server setting
 - Task 7.a: private outsourcing in the client-server setting
 - Task 7.b: *verifiable* outsourcing in the client-server setting

Next Lecture

- Task 7: secure outsourcing in the client-server setting
 - Task 7.a: private outsourcing in the client-server setting
 - Task 7.b: *verifiable* outsourcing in the client-server setting
- Key tool: succinct non-interactive argument (SNARG)

Next Lecture

- Task 7: secure outsourcing in the client-server setting
 - Task 7.a: private outsourcing in the client-server setting
 - Task 7.b: *verifiable* outsourcing in the client-server setting
- Key tool: succinct non-interactive argument (SNARG)
- SNARG for repeated squaring problem in RSA group
 - Pietrzak's interactive protocol
 - SNARG via Fiat-Shamir transform

References

- 1 Most of the lecture is based on Shai Halevi's survey [Hal17], which is a very nice resource on homomorphic encryption.
- 2 The partially homomorphic schemes we discussed are from [ElG84, GM82].
- 3 FHE was introduced in [RAD78], but the first candidate construction was given by Gentry only in [Gen09].
- 4 The GSW FHE was proposed in [GSW13]. The presentation here is taken from Halevi's survey [Hal17].



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.



Craig Gentry.

Fully homomorphic encryption using ideal lattices.

In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.



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.



Ronald L. Rivest, Len Adleman, and Michael L. Dertouzos.

On data banks and privacy homomorphisms.

In *Foundations of Secure Computation*, pages 165–179. 1978.